

Internetprogrammering I

På denna sida finns hela kursen utom utom OH-bilderna och exemplen.

Intro

Välkommen Pierre! På denna sida beskrivs kursen generellt.

Kurskoder

Kursen är på 5 poäng och går för följande universitet/högskolor:

- SU med kod *:94
- KTH med kod 2I1801
- KTH-FoV med kod 2I4114
- Nätuniversitetet med kod IB9010

Kursen ges även i en utökad fördjupande form, kallad IP1+, på 5 + 2 poäng för följande universitet/högskolor:

- SU med kod *:94+

Notera att man **inte senare kan komplettera** från IP1 till IP1+ utan att man måste välja IP1+ direkt från start.

Innehåll

Denna kurs lär ut grundläggande Internetprogrammering med fokusering på socket-programmering:

- Understödjande tekniker:
 - Multitrådning
 - Java arkivfiler (JAR)
 - Reguljära uttryck (REGEXP)
 - Javadoc
- Start från HTML:
 - Äpplen
 - Web Start
- Högnivåkopplingar till specifika servrar (HTTP, FTP, Gopher med mera) med URL och besläktade klasser:
 - Otolkad information
 - Tolkad information
- Lågnivåkopplingar godtyckliga servrar:
 - Stream sockets (TCP):
 - Klient-sidan
 - Server-sidan
 - Datagram sockets (UDP):
 - Unicast
 - Multicast
 - Raw sockets (ICMP med mera) via Java Native Interface (JNI)
- Databaskopplingar med Java Database Connectivity (JDBC)
- E-post med Javamail:
 - Sändning
 - Mottagning
- Mediaöverföring:

- Talsyntes med Java Speech
- Ljud med Java Sound över stream sockets
- Ljud med Java Sound över datagram sockets
- Video med med Java Media Framework (JMF) över RTP

Språk: Java, SQL.

Undervisningsform

Pedagogiken för denna kurs är:

- (1) Frihet, (2) tävling och (3) samarbete --> glädje och motivation --> ett bra resultat!

Frihet, tävling och samarbete gör att man går från passivt mottagande till aktivt sökande. Genom frihet tillåts man själv bli nyfiken och ställa frågor innan man får svar. Genom tävling motiveras man att göra sitt bästa. Detta leder fram till glädje och motivation och därmed till ett bra resultat.

Observera! Detta är en distanskurs och kräver förmåga till självständigt arbete. Kursen har dock, till skillnad mot vanliga distanskurser, kontinuerligt och direkt stöd via handledning på distans. Kursen är krävande och man bör inte läsa andra kurser samtidigt. Kursen fokuserar starkt på praktiskt arbete och examinationen bygger på detta arbete. Digitalslöjd!

Examination

Examinationen på kursen består av:

- Praktiska uppgifter, 4 poäng (R/G/VG) (R/3/4/5):
 - Flera bestämda/styrda
 - Ett fritt valt gesällprov
- Tentamen, 1 poäng (U/G) (U/3)

För fördjupningskursen gäller utöver ovanstående även:

- Extra praktiska uppgifter, 2 poäng (R/G) (R/3):
 - Flera bestämda/styrda

Totalbetyget på båda kurserna bestäms av de praktiska uppgifterna med tyngdpunkt på det fritt valda gesällprovet.

Notera att man måste lämna in lösningar på de praktiska uppgifterna som stämmer med **nuvarande läsårs praktiska uppgifter**. Kurserna utvecklas dock bara i små steg med gradvisa förfiningar så det innebär inte att man måste göra om allt om man inte har fått betyg på de praktiska uppgifterna vid den sista deadline i Augusti.

Litteratur

Litteraturen är:

- »[Java Network Programming \(andra utgåvan\)](#)« skriven av Harold och utgiven av »[O'Reilly](#)«

Kapitel 1-11 och 13-19 ingår i kursen. Man måste inte köpa boken utan kan använda »[SUN: The Java Tutorial](#)« som finns gratis på Internet men detta rekommenderas inte.

Billigaste boken hittas med hjälp av »[Bokfynd.nu](#)«. Vi rekommenderar »[Amazon](#)« och »[WHSmith](#)« i England som brukar ha bra priser och snabba leveranser. Köp inte böcker utanför EU eftersom det tar lång tid (2-3 veckor) att få böckerna och sen får man en ganska stor extra avgift från tullen.

4 steg för att klara denna kurs

Gör följande för att klara denna kurs:

1. Registrera dig hos »[DSV: Studievägledningen](#)« som har e-post svl@dsv.su.se
2. Köp och läs kurslitteraturen
3. Lös och redovisa uppgifterna
4. Anmäl dig till tentamen (minst en vecka innan den går) och klara denna

Lärare

Vi önskar alla en givande och rolig kurs!

- »[Pierre A. I. Wijkman](#)«, ansvarig
- Handledarna

Lycka till!

Schema

På denna sida finns alla tider och platser för kursen. Alla tider är preliminära och kan ändras.

Föreläsningar

Detta är en distanskurs och vi ger inte en hel föreläsningsserie. Den inledande föreläsningen fungerar enbart som en kick-off eller frösättning. Notera att innehåll markerat som reserv innebär att ingen föreläsning ges men att detta kan ändras vid behov.

SU

Tid	Rum	Föreläsare	Innehåll
2004-08-23 kl 13-17	Aulan, Forum, Kista	Pierre A. I. Wijkman	Intro, innehåll, Internet, trådning, äpplen, JAR-filer, URL-klassen, sockets
2004-08-24 kl 13-17	Aulan, Forum, Kista	-	Reserv

KTH

Tid	Rum	Föreläsare	Innehåll
2004-08-30 kl 15-19	Aulan, Forum, Kista	Pierre A. I. Wijkman	Intro, innehåll, Internet, trådning, äpplen, JAR-filer, URL-klassen, sockets
2004-08-31 kl 13-17	Aulan, Forum, Kista	-	Reserv

Handledning

All handledning sker via e-post och news (se sidan Hjälp).

Samtliga studenter har handledning:

- Kursens 1:a - 5:e vecka: samtliga vardagar kl 10-17

Studenter som läser +-kursen (7 poäng) har även handledning på +-delen:

- Kursens 6:e - 7:e vecka: samtliga vardagar kl 10-17

Studenter från Nätuniversitetet som läser på halvfart har även handledning:

- Kursens 6:a - 10:e vecka: samtliga vardagar kl 10-17

För de som arbetar på DSV finns info om prio-tider för labbsalarna under »[DSV: Scheman](#)«.

Examination

Notera följande angående tid för rättning:

- Det brukar ta upp till **3 veckor tills rättningen är klar**
- Ansvarig kan **inte rätta vissa personers uppgifter före andras**, exempelvis de som har bråttom på grund av CSN
- När man fått betyg på kursen så dröjer det **ytterligare tid** tills allt är infört i Daisy och Ladok

Var god **e-posta inte ansvarig om detta** för att försöka påskynda processen, det är väldigt många studenter på samtliga kurser.

Man får välja tid för tenta och deadline oberoende av om man går på SU eller KTH och så vidare. Typen nedan är enbart en rekommendation.

Deadlines för uppgifterna

Det finns flera deadlines för inlämning av uppgifterna och man väljer själv en som passar. Man redovisar samtliga uppgifter direkt eftersom det av praktiska skäl tyvärr inte går att delrätta och ge återkoppling efter hand som deluppgifterna blir klara. De som går en + kurs kan dock lämna in ordinarie uppgifter och + uppgifter separat vid olika deadlines. Varje deadline är normalt daterad till dagen innan tentamen.

Tid	Rum	Typ
2004-09-24 kl 24	Internet	Deadline 1 SU kurs IP1
2004-10-19 kl 24	Internet	Deadline 1 KTH och SU kurs IP2+
2004-12-09 kl 24	Internet	Deadline 2 SU
2005-03-16 kl 24	Internet	Deadline 2 KTH
2005-08-10 kl 24	Internet	Sommardeadline

Observera:

- Redovisade uppgifter **rättas endast i samband med dessa deadlines och uppskov ges aldrig**
- Får man **rest** så **rättas den igen först vid nästa deadline**
- Betyget påverkas inte beroende på vilken deadline man väljer
- Man måste inte välja en deadline som ligger i anslutningen till en tenta
- Handledningen slutar ofta innan deadline 1 (det finns tyvärr inga resurser för en längre handledningsperiod)

Tentamen

Inga hjälpmedel är tillåtna på tentamen. Man bör ha gjort klart de obligatoriska uppgifterna för 5 poängskursen (exklusive gesällprovet) innan tentamen eftersom tentamen kommer att fråga om dessa (detta är dock inget strikt krav för att få gå upp på tentamen). Vissa frågor kan vara mer

generella och vissa kan vara mer tekniska och fråga om hur viss kodning (med avseende på uppgifterna) görs.

Tid	Rum	Typ
2004-09-25 kl 09-13	Laduvikssalen, Frescati	Tentamen SU
2004-10-20 kl 15-19	Forum, Kista	Tentamen KTH
2004-12-10 kl 16-20	Forum, Kista	Omtentamen SU
2005-03-17 kl 13-17	Forum, Kista	Omtentamen KTH
2005-08-11 kl 13-17	Forum, Kista	Sommartentamen

Observera:

- Man måste anmäla sig i Daisy **senast en vecka innan tentamen går**, se »[DSV: Att tentera vid DSV](#)«
- Man måste inte välja en tenta som ligger i anslutningen till en deadline
- Sal meddelas i Daisy eller på telefonsvarare 08 674 70 04 senast en dag innan tentamen, se även »[DSV: Scheman](#)«

Var god **e-posta inte ansvarige** (han har inga befogenheter här) **om detta**, utan kontakta istället »[DSV: Studentexpeditionen](#)«.

Kursmaterial

På denna sida finns information om arbetssätt och vilka mjukvaror man behöver. Här finns även OH-bilder och tidigare tentor.

Arbetssätt och mjukvaror

På denna kurs kan man arbeta på följande sätt:

- På DSV:
 - [MySQL](#)
- På distans online kopplad via telnet eller SSH till sitt DSV/Unix-konto:
 - [Terminalprogram](#)
 - [MySQL](#)
- På distans offline med det egna systemet konfigurerat för allt som behövs:
 - [Editor](#)
 - [Java](#)
 - [MySQL](#)
 - [Filöverföringsprogram](#)

Det senare är det klart roligaste och mest lärorika sättet eftersom man då måste installera allt som behövs på den egna datorn. Notera att ingen handledning ges på detta eftersom det finns alltför många olika varianter av konfigurationer.

OH-bilder

OH-bilderna finns enbart här:

- [00_intro.pdf](#)
- [01_content.pdf](#)
- [02_internet.pdf](#)
- [03_java_net.pdf](#)
- [04_java_applets.pdf](#)
- [05_java_jars.pdf](#)

- [06_java_threads.pdf](#)

Tidigare tentor

Tidigare tentor finns enbart här:

- [Tidigare tentor](#)

Notera att de är baserade på de uppgifter som var obligatoriska vid tidpunkten för tentan och som möjligtvis inte är det nu.

Terminalprogram

På denna sida beskrivs hur man använder terminalprogram.

Terminalserver

Om man behöver logga in och köra kommandon på en avlägsen dator så använder man ett terminalprogram/klient. På DSV används följande dator som terminalserver:

- `triton.dsv.su.se`

Denna dator kör Linux och har både telnet- och SSH-server igång.

Om man inte släpps in kan det bero på att DSV's brandvägg inte släpper in just det IP-nummer som man använder. Då kan man kontakta [Christer Jansson](#) som kan registrera IP-nummret i brandväggen så att man kommer förbi den.

Telnet

I Win32 och Linux så kan man köra ett Terminal-program direkt i ett kommando-fönster:

- `telnet triton.dsv.su.se`

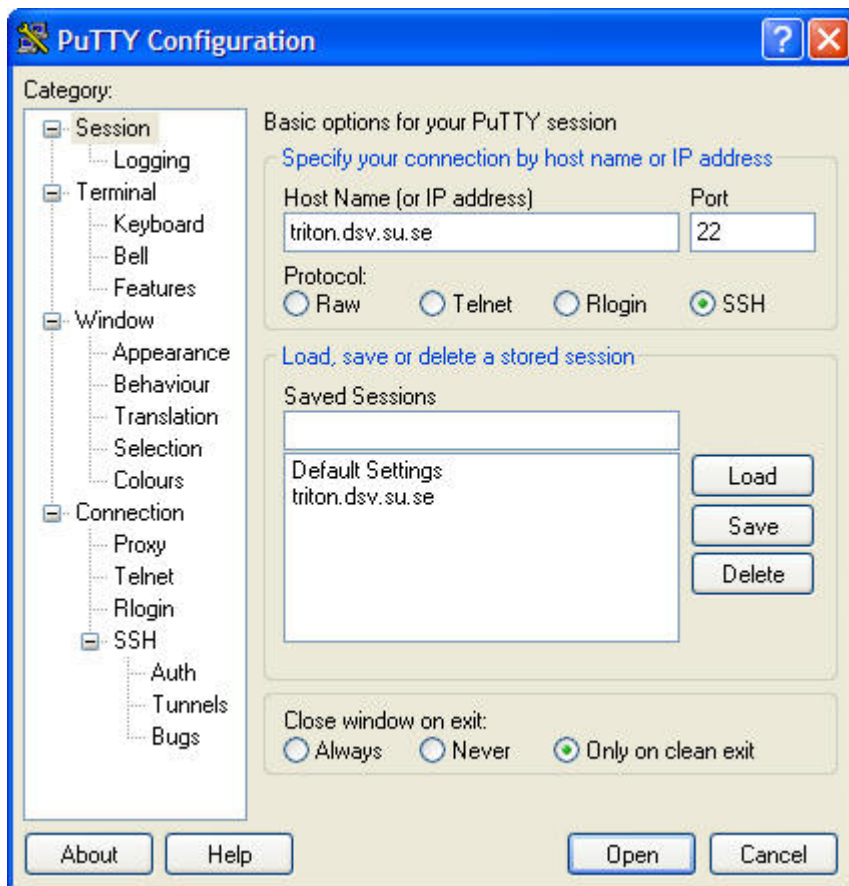
Man anger sedan användarnamn och lösenord.

SSH

När man använder telnet så sänds all information okrypterad över Internet och det kan vara farligt. Vill man undvika det kan man istället använda SSH och följande program rekommenderas:

- Win32: »[Putty](#)«
- Linux: `ssh`

Så här ser det ut när man startar Putty:



Editor

På denna sida beskrivs det verktyg man använder för att skapa kod med.

Intro

Man kan använda allt från en enkel editor till en avancerad grafisk utvecklingsmiljö. Valet är fritt! En avancerad grafisk utvecklingsmiljö kan vara något svårare att komma igång med men är överlägsen när man väl lärt sig den. En grafisk utvecklingsmiljö kan generera vissa delar av koden i ett program automatiskt, exempelvis den kod som tar hand om det grafiska användargränssnittet, vilket snabbar upp konstruktionsarbetet betydligt. En stor nackdel med detta är att koden tenderar att bli omfattande och främmande för en nybörjare och det blir mycket svårt att anknyta sin egen kod till den kod som verktyget har skapat. Det bästa är att börja med en enkel editor och sedan gå över till en grafisk utvecklingsmiljö.

Observera att koden ska vara läsbar direkt i en vanlig browser för enklare rättning. Man bör därför inte använda en editor som använder kontrolltecken och tabbar för att indentera koden.

Enkla

- Notepad (Win 32)
- »[Pico](#)« (Unix)
- »[Emacs](#)« (Win32 och Unix)

Använder man Notepad så måste man spara filerna genom att skiva citationstecken runt filnamnet "X.java" när man namnger och sparar filen, annars sparas filen som X.java.txt.

Använder man emacs för Java bör man inte använda JDE-tillägget eftersom det lätt blir fel med classpath med mera. Kompilera, kör, sätt classpath med mera i ett kommandofönster istället:

- Skriv koden i en editor och spara filen (här x.java)
- Starta ett kommandofönster och gå till katalogen där koden ligger sparad
- Kompilera med `javac x.java`
- Kör med `java x`

Tips Win32:

- Har man Windows95/98/ME så bör man börja med att skriva doskey i kommandofönstret så att man kan använda "uppåtpil" och "nedåtpil" istället för att skriva om allt man skrivit tidigare
- Man kan lägga en genväg till DOS-promten i varje katalog och konfigurera (höger-klicka och välj Egenskaper) så att den startar i samma katalog:
 - Windows 98: Välj Program/Arbetsmapp och skriv en .
 - Windows 2000/XP: Välj Shortcut/Start in och gör detta text-fält tomt
- När det är långa krångliga kommandon så kan man i DOS använda BAT-filer genom att skriva kommandot i en fil med namnet run.bat så kan man sedan köra kommandot med run

Vill man strukturera upp sin XHTML-kod så rekommenderas:

- »[W3C: Tidy](#)«
- »[SourceForge: Tidy](#)«
- »[Charlie's Tidy Add-Ons](#)«

Mer komplexa

- »[SciTe](#)«, se även »[SciTe Bundled](#)«-versionen
- »[JEdit](#)« (för Java-kod)
- »[HTML-kit](#)« (för XML- och XHTML-kod)

Konfiguration av SciTE:

- Ladda den fil som bestämmer SciTE's funktion: Options / Load Global Options File
- Sen kan man sätta lite olika saker, exempelvis:
 - Använd 3 mellanslag vid intendering: `indent.size=3`
 - Använd inte tabbar vid intendering: `use.tabs=0`
- Spara sen filen

Sessionshantering med SciTE:

- Om man har många filer öppna och vill fortsätta med dessa senare så sparar man sessionen med: File / Save Session ...
- När man vill öppna alla filer igen sen så går man till: File / Load Session ...

Kompleta utvecklingsmiljöer

- »[Eclipse](#)« (för Java-kod)

Java

På denna sida beskrivs installation av Java och Java API specifikationen samt konfiguration av Java.

Installation av Java

Java finns i tre huvudversioner:

- Java Micro Edition (J2ME) som är gjord för mobila/trådlösa enheter
- Java Standard Edition (J2SE) som har fokus på klientsidan
- Java Enterprise Edition (J2EE) som har fokus på serversidan

Vi ska i fortsättningen titta på J2SE som finns i sex huvudversioner: 1.0.x, 1.1.x, 1.2.x, 1.3.x, 1.4.x och 1.5.x (kallad 5.x). Det är stor skillnad mellan dessa versioner och på denna kurs används den senaste versionen. Den finns för nerladdning för olika operativsystem:

Version 1.4.2:

- DOS/Windows95/98/ME: »[J2SE 1.4.2](#)«
- Windows NT/2000/XP: »[J2SE 1.4.2](#)«
- Unix/Linux: »[J2SE 1.4.2](#)«
- MacOS: »[J2SE 1.4.2](#)«

Version 1.5.0:

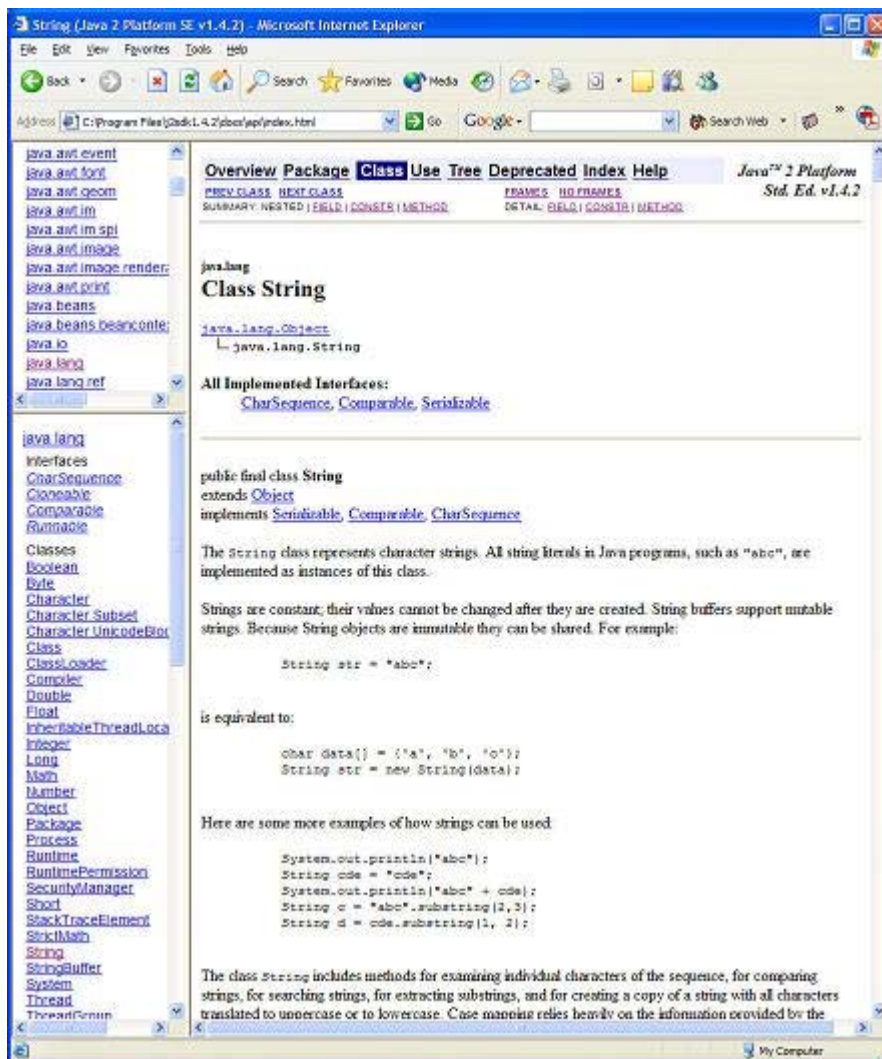
- DOS/Windows95/98/ME: »[J2SE 1.5.0](#)«
- Windows NT/2000/XP: »[J2SE 1.5.0](#)«
- Unix/Linux: »[J2SE 1.5.0](#)«
- MacOS: <finns ej än>

I texten nedan används version 1.4.2 i sökvägar med mera och man får byta denna text så att den passar den version som man själv använder. Efter nerladdning installerar man Java genom att starta installationsprogrammet.

Installation av Java API specifikationen

Java API specifikationen är en datorbaserad uppslagsbok som behövs eftersom Javas bibliotek av klasser är enormt stort och växer snabbt hela tiden för varje ny version av Java. Om man ska slå upp specifika saker som klass-namn, variabel-namn, metod-namn med mera så är Java API specifikationen till mycket stor hjälp. Notera att man bara får dåligt självförtroende om man börjar med Java API specifikationen utan att ha en grundläggande förståelse och utan att man har testat några programexempel tidigare.

Java API specifikationen kan läsas online men även laddas ner från samma ställe som Java hämtas från om man vill läsa den offline. Uppslagsboken körs i en webb-läsare och ser ut så här:



Notera att det finns en stor mängd tillbehör till Java som inte ingår i någon av huvudversionerna vars API-beskrivningar måste hämtas ner separat.

Konfiguration av Java

Enkel körning

För att kunna nå alla program som ingår i J2SE på ett enkelt sätt så sätter man omgivningsvariabeln path till den katalog där programmen ligger. Då kan man exempelvis köra java- och javac-programmen från alla kataloger utan att behöva skriva hela sökvägen till dessa program eftersom systemet nu hittar dessa program automatiskt.

DOS/Windows95/98/ME:

- Antag att vi har installerat java till c:\Program Files\j2sdk1.4.2_04
- Öppna filen c:\autoexec.bat från ett DOS-fönster genom att skriva edit autoexec.bat
- Skriv följande sist i denna fil:
 - Nya versioner av operativen: set path=%path%;C:\Program Files\j2sdk1.4.2\bin
 - Gamla versioner av operativen: set path=%path%;C:\Progra-1\j2sdk1.4.2\bin
- Spara, avsluta och starta om datorn

Windows NT/2000/XP:

- Antag att vi har installerat java till c:\Program Files\j2sdk1.4.2_04

- Klicka på Start/Settings/Control Panel/System/Advanced/Environment Variables...
- Under System variables, välj Variable: Path och klicka på Edit
- Lägg till följande sist i Variable Value: ;C:\Program Files\j2sdk1.4.2_04\bin
- Spara, avsluta och starta om datorn

Unix/Linux:

- Eftersom du är Unix/Linux-användare så ser du detta som en utmaning och vill inte fuska och ha någon instruktionen för detta :-).

MacOS:

- Eftersom du är MacOS-användare så fungerar allt automatiskt och du behöver inte någon instruktionen för detta :-).

Tillbehör

Det finns ett stort antal tillbehör till de tre huvudversionerna (J2ME, J2SE, J2EE). Dessa tillbehör kallas för standard extensioner eller extra paket och är utbyggnader som finns inom många olika områden, exempelvis säkerhet och media. För att använda ett tillbehör så går man till SUN's eller någon annan leverantörs sida på Internet och hämtar en eller flera filer.

Det finns två huvudtyper av tillbehör:

- Plattformsberoende gjorda enbart med Java
- Plattformsberoende gjorda med Java och native kod

Plattformsberoende tillbehör används för saker där effektiviteten är viktig eller om tillbehöret måste nå plattformsspecifik hårdvara som i exempelvis vid användande av 3D-grafik och/eller andra media-tekniker

Det finns tre huvudsätt att köra Java-program:

- Fristående program
- Äpplen
- Web Start

Varje sätt använder tillbehör på sitt eget speciella sätt och sammanfattningsvis kan man använda tillbehör på följande sätt:

- Fristående program:
 - Plattformsberoende tillbehör:
 - Tillbehöret läggs i speciella kataloger
 - Tillbehöret ligger på ett godtyckligt ställe och man sätter classpath dit
 - Tillbehöret ligger på samma ställe som huvudprogrammet
 - Plattformsberoende tillbehör:
 - Tillbehöret läggs i speciella kataloger
 - Tillbehöret ligger på samma ställe som huvudprogrammet
- Äpplen:
 - Plattformsberoende tillbehör:
 - Tillbehöret läggs i speciella kataloger
 - Tillbehöret ligger på samma ställe som huvudprogrammet
 - Plattformsberoende tillbehör:
 - Tillbehöret läggs i speciella kataloger
- Web Start:
 - Plattformsberoende tillbehör:
 - Tillbehöret läggs i speciella kataloger
 - Tillbehöret ligger på samma ställe som huvudprogrammet
 - Plattformsberoende tillbehör:

- Tillbehöret läggs i speciella kataloger

Vi ska inte gå in på alla detaljer om detta men titta lite generellt på de tre olika huvudsätten att använda tillbehör, se »[Java extensioner med nativelib-filer](#)« för mer detaljerad information.

Tillbehöret läggs i speciella kataloger

JAR- och eventuella DLL- (eller motsvarande) filer placeras enligt:

- JAR-filer läggs här:
 - För att kunna köra program (JRE): <sökväg>\Java\j2re1.4.2_04\lib\ext
 - För att kunna utveckla program (kompilera mm) (JDK): <sökväg>\j2sdk1.4.2_04\lib
- DLL-filer (eller motsvarande) läggs här:
 - För att kunna köra program (JRE): <sökväg>\Java\j2re1.4.2_04\bin

Notera att sökvägarna ovan kan vara olika på olika maskiner.

När JAR- och DLL- (eller motsvarande) filer ligger på dessa ställen behöver man inte sätta classpath till dessa filer för att kunna köra och utveckla program som är beroende av tillbehöret.

Tillbehöret ligger på ett godtyckligt ställe och man sätter classpath dit

Om filerna ligger på andra ställen än beskrivet ovan så kan man sätta omgivningsvariabeln classpath till antingen:

- en katalogstruktur med CLASS-filer, eller
- en eller flera JAR-filer

Genom att sätta classpath så berättar man för programmen javac, java med flera var detta tillbehör finns. Notera att man även kan sätta classpath direkt när man använder javac eller java.

DOS/Windows95/98/ME:

- I filen C:\autoexec.bat skriver man på sista raden set classpath=%classpath%;c\x\y

Windows NT/2000/XP:

- Gå till Kontrollpanelen/System/Avancerat/Omgivningsvariabler och sätt eller uppdatera classpath.

Unix/Linux:

- Man sätter classpath olika beroende på vilket skal man använder:
 - C-skal: setenv classpath <sökväg>
 - Bourne-skal: classpath=<sökväg> och sen: export classpath
 - Korn-skal: export classpath=<sökväg>
 - Bash-skal: export classpath=<sökväg>

Detta kan man lägga i den fil som kör automatiskt (som har olika namn i olika skal) när man loggar in.

MacOS:

- Se Unix/Linux

Tillbehöret ligger på samma ställe som huvudprogrammet

Man kan slutligen lägga tillbehöret på samma ställe som huvudprogrammet. Är tillbehöret packat i en JAR-fil så packar man upp det i denna katalog. Mer tydligt, antag att man har sina Java-program här:

- C:\x\y

Då ska man lägga tillbehörskatalogen z här:

- C:\x\y\z

En nackdel med detta sätt är att man måste ha flera kopior av tillbehöret om man ska använda det på flera ställen.

Filöverföringsprogram

På denna sida beskrivs hur man använder filöverföringsprogram.

Intro

För att kunna sända filer mellan olika datorer använder man filöverföringsprogram och vanliga standarder för det är FTP, SFTP och FTPS.

Ett vanligt fall är när man har HTML-filer som man vill sända till DSV för att synliggöra dem på Internet. Man använder då sitt DSV/Unix/Linux-konto och lägger allt under `www`-katalogen. När filerna är överförda måste man även sätta rättigheterna på filerna och katalogerna:

- Kataloger: `chmod 755`
- De flesta filtyperna: `chmod 744 *`
- CGI-filer: `chmod 711 *.cgi`

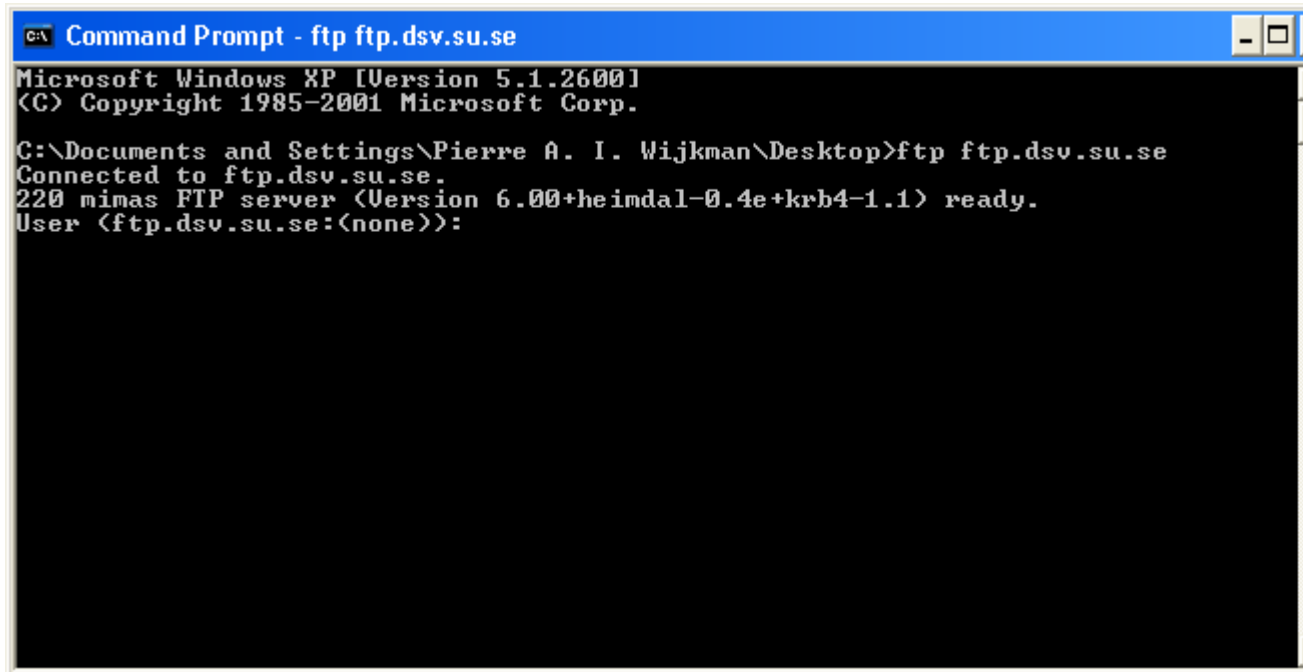
Notera att `www`-katalogen inte ska vara med i URL'ens sökväg när man sedan ska hämta filer i en browser:

- En fil `z.html` på DSV/Unix/Linux-kontot under katalogen `www/x/y` når man i en browser med `http://dsv.su.se/~<användarnamn>/x/y/z.html`

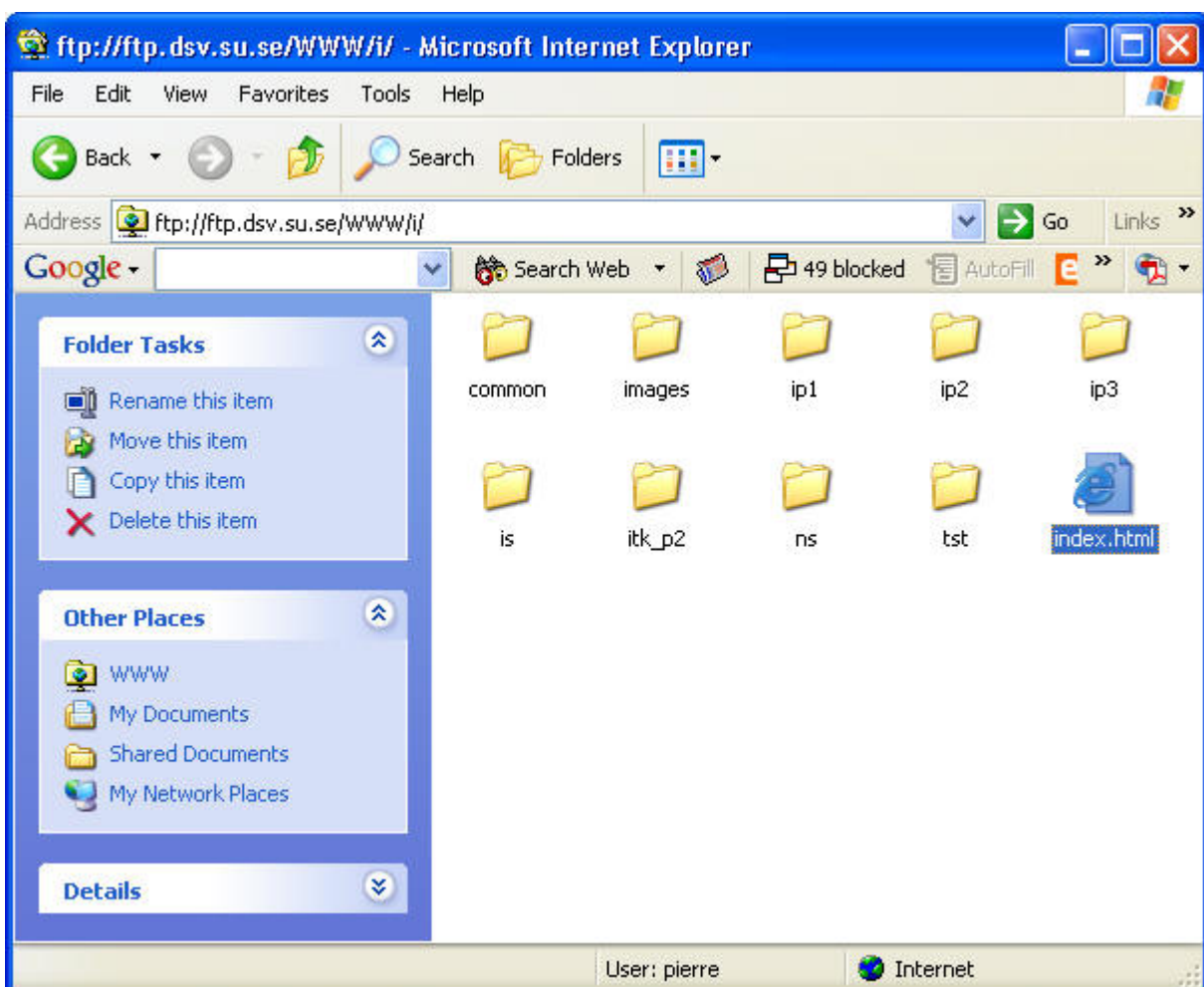
Man kan även nå sina filer på DSV/Win32 men det är endast DSV/Unix/Linux som har en tillförlitlig HTTP-server.

FTP

I Win32 och Linux så kan man köra en FTP-klient direkt i ett kommando-fönster:



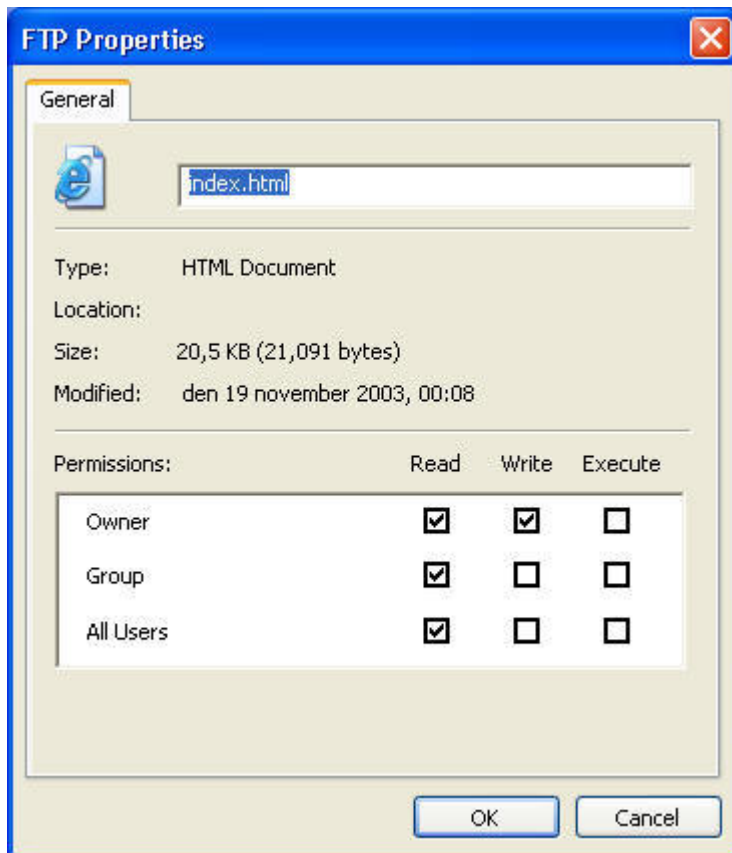
Det är dock ofta lättare att använda en GUI-baserad FTP-klient. Det finns många olika program och »Shareware.com« är ett ställe som man kan leta på. Man även använda en vanlig browser, exempelvis Internet Explorer:



Man använder sen drag-and-drop. Man kan även sätta fil- och katalogrättigheter från detta program:

- Högerklicka på fil eller katalog
- Välj Egenskaper
- Sätt rättigheter

Så här kan det se ut:



Notera dock att Internet Explorer inte stöder ASCII-mode.

Man ansluter sig till FTP-servern på följande dator:

- DSV/Unix/Linux: `ftp://ftp.dsv.su.se`

Använder man Internet Explorer så kan man skapa en genväg enligt:

- `ftp://<användarnamn><lösenord>@ftp.dsv.su.se`

Notera att man bör sända:

- Rena text-filer (som CGI-program med mera) i ASCII-mode
- Binära filer (som bilder, kompillerade program med mera) i BINARY-mode

Använder man en brandvägg så bör man se till att initieringen av dataöverföringskanalen sker på klient-sidan och det gör man genom att använda `passive mode` som kan ställas in i de flesta FTP-klienter.

SFTP och FTPS

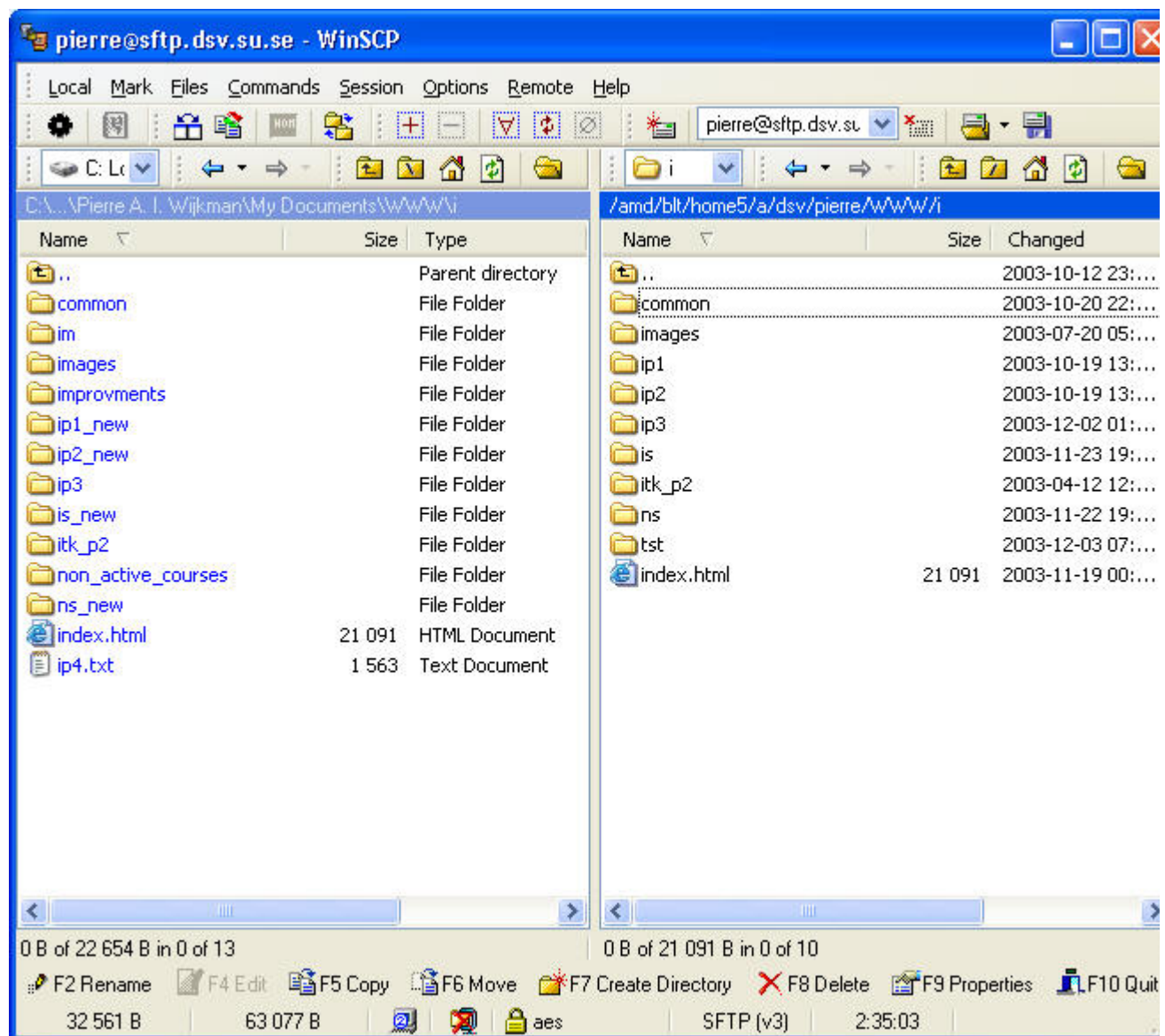
När man använder FTP så sänds all information, även lösenordet, okrypterat över Internet och det kan vara farligt. Man bör istället använda någon av följande tekniker:

- SFTP som använder sig av tunnling via SSH och först upprättar en SSH-anslutning som sedan används för att ansluta till en server via FTP
- FTPS som ansluter som vanlig FTP enligt RFC959 för att sedan begära en säker anslutning via SSL/TLS enligt RFC2228

DSV stöder enbart SFTP så man måste använda ett sådant program och bra program för det är:

- Unix/Linux: sftp som ingår i OpenSSH
- Win32: »[WinSCP](#)«

Exempelvis så ser WinSCP ut så här:



Man ansluter till SFTP-servern på följande datorer:

- DSV/Unix/Linux: sftp.dsv.su.se
- DSV/Win32: wftp.dsv.su.se

Detta säkrare sätt rekommenderas starkt.

MySQL

På denna sida finns hela MySQL beskrivningen.

Intro

På denna sida introduceras databasservern MySQL. Det finns två sätt att använda denna:

- Använda DSV's centrala MySQL
- Använda en egen MySQL

Både DSV's centrala MySQL och en egen MySQL kan användas men vi rekommenderar att man börjar med DSV's centrala MySQL.

Man kan använda ett GUI mot MySQL för att få en tydligare bild av vad som sker i databasen, exempelvis:

- [MySQLFront](#) (Win32), »[MyCC](#)«
- »[Navicat](#)« (Unix, Win32, OSX)
- »[YourSQL](#)« (OSX)
- »[PhpMyAdmin](#)« (DSV's via HTTP där man anger sitt MySQL-användarnamn och -lösenord)

@DSV

På denna sida beskrivs hur man använder DSVs databasserver.

Adress

På DSV körs en MySQL-databasserver här:

- Host: atlas.dsv.su.se
- Port: 3306

Kontouppgifter

Du (Pierre Wijkman) har en **egen databas** på denna MySQL-databasserver som har följande uppgifter:

- MySQL databasnamn: db_4926495
- MySQL användarnamn: usr_4926495
- MySQL lösenord: 926495

Dessa kontouppgifter är personliga och man ska **inte visa** dem för någon annan.

Privilegier

Man kan koppla sig till sin databas på två sätt:

- Via localhost och ett server-sides program som kör på atlas.dsv.su.se
- Direkt från en annan dator

Man har följande privilegier:

- ALTER

- CREATE
- DELETE
- DROP
- INDEX
- INSERT
- SELECT
- UPDATE

Notera att databasen från början inte innehåller några tabeller utan att man får skapa dessa själv, exempelvis direkt från sitt program eller från ett grafiskt användargränssnitt.

Kopiering av databas

Om man senare vill kopiera sina data i denna databas till en annan databas så använder man MySQL-programmet `mysqldump` med växel `--opt`. För att få mer information, se man `mysqldump` på en Linux-maskin. Själva överföringen av data till externt system kan göras med något av SFTP, SCP, FTP (helst kerberos), `wget`, bränna CD och så vidare.

@Hemma

På denna sida beskrivs hur man använder en egen databasserver.

Installation

Hämta och installera en egen databasserver »[MySQL](#)«.

Starta MySQL

Gå till katalogen `<sökväg>/mysql/bin` och kör:

- Unix/Linux: `nohup mysqld&`
- Win32: `mysqld`

Sätta root-lösenord

Gå till katalogen `<sökväg>/mysql/bin` och kör:

- Unix: `mysqladmin -u root password <password_root>`
- Win32: `mysqladmin -u root password <password_root>`

Skapa en databas och ett konto för databasen

Ett konto består av:

- Ett databasnamn, `<database_name>`
- Ett användarnamn, `<username>`
- Ett lösenord, `<password_account>`

Gå till katalogen `<sökväg>/mysql/bin` och kör:

- `mysql --user=root --password=<password_root> mysql`
- `CREATE DATABASE <database_name>;`

För att få access från localhost:

- GRANT ALL PRIVILEGES ON <database_name>.* TO '<username>'@'localhost' IDENTIFIED BY '<password_account>' WITH GRANT OPTION;

För att få access utifrån:

- GRANT ALL PRIVILEGES ON <database_name>.* TO '<username>'@'%' IDENTIFIED BY '<password_account>' WITH GRANT OPTION;

Avsluta med:

- QUIT

Stoppa MySQL

Gå till katalogen <sökväg>/mysql/bin och kör:

- Unix/Linux: mysqladmin -u root -p shutdown
- Win32: mysqladmin -u root -p shutdown

Man måste man stoppa MySQL innan man stänger av datorn.

Tidigare tentor

Tentamen

Internetprogrammering I

1. Beskriv hur man skapar ett användargränssnitt i Java
2. Beskriv hur händelsestyrd programmering fungerar
3. Beskriv hur man via strömmar och sockets kan skicka och hämta information via Internet
4. Beskriv hur trådning fungerar och visa med en skiss vilka komponenter som ett trådat program innehåller
5. Beskriv ditt gesällprov och visa med en skiss hur du gjort det
6. Beskriv möjliga förbättringar av ditt gesällprov

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 1997-06-04

Tentamen

Internetprogrammering I

1. Beskriv kortfattat samtliga paket i Java
2. Vilka metoder finns i Java för att hämta och skicka information på Internet? Beskriv dessa kortfattat
3. Beskriv hur trådning fungerar
4. Beskriv ditt gesällprov och visa med en skiss hur du gjort det
5. Beskriv möjliga förbättringar av ditt gesällprov

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 1997-08-30

Tentamen

Internetprogrammering I

1. Beskriv begreppen objekt, klass, abstrakt klass och interface
2. Beskriv och jämför de olika sätten att använda Internet i uppgifterna 1-3
3. Beskriv hur man bygger upp ett användargränssnitt och hur man kopplar händelser till komponenter i detta
4. Beskriv hur man genom (1) arv och (2) implementering av ett gränssnitt kan skapa en tråd

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 1998-12-16

Tentamen

Internetprogrammering I

Svaren skall vara koncisa och välstrukturerade, ca 1-2 sidor per fråga. För G krävs helt rätt på minst fyra frågor. Samtliga frågor gäller språket Java.

1. Vad är en klass-metod? Vad är en klass-variabel? Varför är main-metoden deklarerad statisk?
2. Vad är ett interface (ej användargränssnitt)?
3. Vad är en socket? Vilka typer av sockets finns det? Beskriv dessa
4. Beskriv hur man skapar en lyssnare till en knapp
5. Beskriv två sätt att få ett program trådat. Beskriv för- och nackdelar för dessa
6. Beskriv hur undantagshanteringen fungerar. Hur skapar man en metod som kan generera en exceptionell händelse?

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 1999-01-16

Tentamen

Internetprogrammering I

Beskriv detaljerat hur du gjorde gesällprovet. I beskrivningen ska finnas både en generell (vanligt språk) beskrivning och en specifik (Java) beskrivning. Den specifika beskrivningen behöver inte vara hela programmet utan bör endast visa hur olika, för uppgiften speciella, funktioner implementerats. Koden skall skrivas på ett mycket välstrukturerat sätt (exempelvis ej med rak vänster). Den specifika beskrivningen skall kommenteras med vanligt språk. Totalt bör hela beskrivningen omfatta minst 5 sidor. Alla dessa villkor måste vara uppfyllda för att uppgiften skall ge 1 p.

För G krävs 1 poäng. Inga hjälpmedel. Pierre A. I. Wijkman ▫ 1999-08-30

Tentamen

Internetprogrammering I

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

1. Beskriv med Java-kod två sätt att göra (multi)trådade program
2. Beskriv hur och varför man signerar kod
3. Beskriv med Java-kod användningen av klasserna Socket och ServerSocket
4. Beskriv kort hur Servlets används

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 1999-12-16

Tentamen

Internetprogrammering I

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

1. Beskriv med Java-kod hur man gör en enkel browser
2. Beskriv begreppet Multicast
3. Beskriv med Java-kod användningen av klassen ServerSocket
4. Beskriv hur omgivningsvariablerna kan användas av en Servlet

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2000-01-12

Tentamen

Internetprogrammering I

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Beskriv generellt och med Java-kod hur man gör en enkel chat-klient och chat-server.

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2000-08-30

Tentamen

Internetprogrammering I

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Beskriv först generellt hur multicast fungerar. Beskriv sedan med Java-kod hur man implementerar en enkel multicast-chat.

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2000-10-24

Tentamen

Internetprogrammering I

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Beskriv först generellt hur stream-sockets fungerar. Beskriv sedan med Java-kod hur man implementerar en enkel chat-klient med detta.

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2000-12-14

Tentamen

Internetprogrammering I

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Beskriv först generellt hur stream-sockets fungerar. Beskriv sedan med Java-kod hur man implementerar en enkel chat-server med detta.

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2001-01-12

Tentamen

Internetprogrammering I

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Beskriv först generellt Ditt gesällarbete (sista uppgiften). Beskriv sedan i större detalj hur Ditt gesällarbete är kodat, beskriv vilka klasser Du använt och hur dessa samarbetar (Du behöver givetvis inte beskriva all kod i dessa klasser).

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2001-08-27

Tentamen

Internetprogrammering I

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Beskriv först på en generell nivå skillnader och likheter mellan stream- och datagram-sockets. Varför finns det två typer exempelvis? Beskriv sedan generellt (utan kod) hur man implementerar en chat-klient och en chat-server. Vilka Java-klasser och metoder är essentiella i denna implementering?

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2001-09-22

Tentamen

Internetprogrammering I

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Beskriv först på en generell nivå skillnader och likheter mellan klient-server och peer-to-peer modellen. Beskriv sedan generellt (utan kod) hur man implementerar ett "vilka-är-online" system (Uppgift 4 på kursen). Vilka Java-klasser och metoder är essentiella i denna implementering?

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2001-10-24

Tentamen

Internetprogrammering I

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Beskriv med Java-kod hur man gör en enkel chat-klient och en enkel chat-server.

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2002-01-10

Tentamen

Internetprogrammering I

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Beskriv med Java-kod hur man gör ett enkelt program som använder multicast.

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2002-08-16

Tentamen

Internetprogrammering I

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Beskriv vilka Java-klasser och metoder som är essentiella när man gör uppgiften som handlar om multicast (beskriv inte GUI-kod). Visa med kod-exempel hur dessa kod-delar används.

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2002-09-23

Tentamen

Internetprogrammering I

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Beskriv vilka Java-klasser och metoder som är essentiella när man gör uppgift 3a och 3b som handlar om stream sockets (beskriv enbart det viktigaste, inte GUI-kod mm). Visa med kod-exempel hur dessa kod-delar används.

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2002-10-26

Tentamen

Internetprogrammering I

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Beskriv först generellt ditt gesällarbete (sista uppgiften). Beskriv sedan i större detalj hur de centrala delarna i ditt gesällarbete är kodat (man ska givetvis inte beskriva all kod).

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2002-12-18

Tentamen

Internetprogrammering I

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Beskriv först generellt ditt gesällarbete (sista uppgiften). Beskriv sedan i större detalj hur de centrala delarna i ditt gesällarbete är kodat (man ska givetvis inte beskriva all kod).

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2003-08-16

Tentamen

Internetprogrammering I

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Visa med Java-kod hur två program (utan grafiskt gränssnitt) kan kommunicera med hjälp av stream sockets.

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2003-09-19

Tentamen

Internetprogrammering I

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Beskriv med Java-kod hur man gör ett enkelt program som använder datagram-sockets. Programmet ska både kunna sända och ta emot information.

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2003-10-23

Tentamen

Internetprogrammering I

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Beskriv först generellt ditt gesällarbete (sista uppgiften). Beskriv sedan i större detalj hur de centrala delarna i ditt gesällarbete är kodat (man ska givetvis inte beskriva all kod).

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2003-12-19

Tentamen

Internetprogrammering I

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Visa med Java-kod hur man gör ett enkelt program använder högnivåkopplingar för att hämta information på Internet.

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2004-08-14

Tentamen

Internetprogrammering I

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Visa med Java-kod två sätt (arv och implementering av interface) att göra (multi)trådade program. Programmen ska vara fullständiga (med class, metod osv) men minimala och fungera.

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2004-09-25

Tentamen

Internetprogrammering I

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Visa med Java-kod hur man gör ett program utan grafiskt gränssnitt som med hjälp av URL-klassen hämtar en viss HTML-sida på Internet och skriver ut denna i terminalfönstret. Programmet ska vara fullständigt (med class, metod osv) men minimalt och fungera.

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2004-10-20

Tentamen

Internetprogrammering I

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Visa med Java-kod hur man gör ett program utan grafiskt gränssnitt som använder datagram sockets för både sändning och mottagning. Programmet ska vara fullständigt (med class, metod osv) men minimalt och fungera.

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2004-12-10

Tentamen

Internetprogrammering I

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 1 sida.

Beskriv i detalj varför man använder "synchronized".

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2005-03-17

Externa källor

På denna sida finns länkar till externa källor. Tipsa gärna om fler bra länkar.

Sockets

- » [SUN: Custom Networking](#)«
- » [SUN: New and Enhanced Networking Features in the Java 2 Platform, Standard Edition 1.4 \(J2SE\)](#)«
- » [SUN: New I/O APIs](#)«
- » [SUN: Network Programming with Java 2 Platform, Standard Edition 1.4 \(J2SE\)](#)«
- » [University of California, Irvine: Jpcap](#)«

Multicast

- » [SUNET: Personal systems - Mbone suite](#)«
- » [Ohio State University: IP Multicasting: Concepts, Algorithms, and Protocols](#)«
- » [Microsoft: Multicast Streaming - An Introduction](#)«
- » [Microsoft research: MBONE and IP Multicast](#)«
- » [Tekniska Högskolan i Luleå: Multicast och routing](#)«

Speech

- » [SUN: Java Speech API](#)«
- » [SUN: Java Speech Programmer's Guide](#)«
- » [SUN: Java Speech API 1.0](#)«
- » [Speex](#)«
- » [VoiceGenie](#)«

Ljud

- » [SUN: Java Sound API](#)«
- » [SUN: Java Sound Programmer's Guide](#)«
- » [MIDI Manufacturer's Association](#)«
- » [Roland](#)«

Video

- » [SUN: Java Media Framework API](#)«
- » [SUN: Java Media Framework API Guide](#)«
- » [SUN: Java Media Framework API 2.2.1 API Specification](#)«
- » [Apple: MPEG4: The Next Generation](#)«

Uppgifter

På denna sida finns uppgifterna till kursen. Läs först noga alla [krav](#) och sedan alla [tips](#). Siffror inom () är kapitelhänvisningar.

Start från HTML

- 1.1 [Äpplen](#) obligatorisk (1-3, 5, 9)
- 1.2 [Web Start](#) obligatorisk (8)

Högnivåkopplingar

- 2.1 [Otolkad information](#) obligatorisk (6, 7, 15)
- 2.2 [Tolkad information](#) obligatorisk

Lågnivåkopplingar

Stream sockets

- 3.1.1 [Klient-sidan](#) obligatorisk (4, 10)
- 3.1.2 [Server-sidan](#) obligatorisk (11)

Datagram sockets

- 3.2.1 [Unicast](#) obligatorisk (13, 14)
- 3.2.2 [Multicast](#) frivillig IP1, obligatorisk IP1+ (13, 14)

Raw sockets

- 3.3 [Raw sockets](#) frivillig

Databaskopplingar

- 4 [Databaskopplingar](#) obligatorisk

E-post

- 5.1 [Sändning](#) frivillig IP1, obligatorisk IP1+ (19)
- 5.2 [Mottagning](#) frivillig IP1, obligatorisk IP1+ (19)

Mediaöverföring

- 6.1 [Talsyntes](#) frivillig IP1, obligatorisk IP1+
- 6.2.1 [Ljud över stream sockets](#) frivillig IP1, obligatorisk IP1+
- 6.2.2 [Ljud över datagram sockets](#) frivillig
- 6.3 [Video med RTP](#) frivillig

Gesällprov

- 7 [Gesällprov](#) obligatorisk

Krav

På denna sida finns de krav som gäller för uppgifterna.

Samarbete och betyg

- Alla uppgifter utom gesällprovet ska göras i grupp om 1-2 personer, gesällprovet ska göras ensam
- Vissa av uppgifterna är frivilliga men desto fler av dessa som man gör desto större blir sannolikheten för att få betyget VG/5
- Man kan givetvis ändå få betyget VG/5 om man gör ett väldigt bra gesällprov
- Bästa gesällprov till deadline 1 vinner ett pris

Kodens utseende

En vanlig orsak till betyget R är att nedanstående krav inte är uppfyllda.

All kod

- All kod ska vara välstrukturerad
- Man får själv bestämma stil på koden men man måste vara konsekvent
- Använd inte en editor som använder kontrolltecken och tabbar för indentering, koden ska kunna läsas och se bra ut i en vanlig enkel editor som exempelvis Notepad och därmed även i en browser
- Designen på det (eventuella) grafiska användargränssnittet är fri men försök göra det så snyggt som möjligt (man bör alltså inte göra programmen identiska med de ofta lite tråkiga exempelprogram som ges för varje uppgift)

Java-kod

- All Java-kod ska vara välkommenterad med en kommentar per klass och metod, se vidare »[SUN: Code Conventions for the Java Programming Language](#)«
- Eventuella äpplen som ska redovisas kan vara körbara antingen via appletviewer eller via en plug-in i en vanlig browser
- Använd Swing-GUI-komponenter, inte de nu omoderna AWT-GUI-komponenterna (givetvis är det ok att använda layoutmanagers, händelseklasser med mera från AWT)

Tips

På denna sida finns tips för uppgifterna.

Ordning och organisation

- Använd en katalog för varje huvuduppgift
- Under hela konstruktionsprocessen:
 - Välstrukturerad kod
 - Bra namn på klasser, variabler, konstanter

Evolutionär utveckling

Många moderna programmeringsspråk består av:

- En liten språkkärna
- Ett mycket stort bibliotek av färdigkonstruerade klasser

Den självklara fördelen är att man slipper konstruera en stor mängd saker som annars skulle ta mycket lång tid att göra men nackdelen är att det kan vara svårt att veta och hitta vad som finns samt att man inte får samma grundläggande förståelse.

Matematik är ingenting man förstår, matematik är något som man vänjer sig vid

von Neumann

Till att börja med bör man skaffa sig en övergripande förståelse för det område man ska ge sig in på. Detta görs bäst med hjälp av böcker, tidningar och/eller de tutorials och artiklar som finns på Internet. När man har skaffat sig en denna övergripande förståelse så bör man så fort som möjligt testa några av de exempel man stött på. Det är ofta först när man praktiskt tillämpar ett område som man lär sig, eller vänjer sig vid, området och därmed får en förankring eller förståelse av beteendet hos objekten man laborerar med.

Man bör inte skriva av koden "rakt av" utan man bör istället:

- Konstruera en minimal körbar kärna (med sin egen stil på koden)

Dett gör man för varje nytt område man ger sig in på. Man kan sedan:

- Förändra denna minimala körbara kärna mot ett önskbart mål med **små förändringar** av koden

Detta är mycket viktigt! Man går mot ett önskat mål med:

- Små förändringar av koden
- Kompilering
- Testkörning

Man släpper aldrig kontakten med det fungerande programmet.

We are like sailors who on the open sea must reconstruct their ship but are never able to start afresh from the bottom. Where a beam is taken away a new one must at once be put there, and for this the rest of the ship is used as support. In this way, by using the old beams and driftwood the ship can be shaped entirely anew, but only by gradual reconstruction.

Otto Neurath

Detta är en mycket viktig tumregel. Försöker man förändra koden i stora steg så blir det fel och dessa fel är ofta svåra att lokalisera och rätta till. Ofta samverkar flera fel och detta gör det givetvis ännu svårare.

Ofta är stora program konstruerade av många lager/nivåer där varje nivå består av program konstruerade enligt ovanstående enkla principer. Man använder sedan program på en nivå som enheter för konstruktion av program på nästa nivå, och även denna process följer de ovan anvisade principerna.

Övrigt

- Detta är inte övningsuppgifter och man måste läsa kursboken och viss information som finns på Internet för att klara uppgifterna
- Alla program är testade i DSVs miljö för både Win32 och (det överlägsna :-) Unix/Linux
- Alla program går att utveckla lokalt ej uppkopplad (på Win32 och Unix/Linux) men man måste ha TCP/IP installerat på sin dator (vilket man har om man exempelvis surfar på Internet)
- Man kan använda IP-adressen 127.0.0.1 eller localhost som motsvarar den egna maskinen

Start från HTML: Äpplen

Allmänt

Uppgiften går i huvudsak ut på att visa:

- Hur man gör program som kan startas från ett HTML-dokument i en browser och vi tittar först på program som kallas för äpplen

Det är mycket viktigt att man kan trådning och JAR-filer bra så denna uppgift innehåller även dessa moment:

- Vi tittar på hur man gör när man konstruerar ett enkelt multitrådat program. Det finns två sätt att göra en klass trådad på och vi övar här på båda. Det är mycket viktigt att man förstår båda dessa sätt bra eftersom några av de efterföljande uppgifterna är trådade. Trådning är en mycket vanlig och kraftfull teknik.
- Vi tittar på hur man kan arkivera och komprimera en mängd CLASS-filer (och ibland även andra typer av filer som bild-filer med mera) med JAR-verktyget.

Uppgift

Gör ett äpple (med ett grafiskt användargränssnitt) som har två knappar och en text-area. Respektive knapp ska ha texterna Aktivera utskrift från tråd 1 och Aktivera utskrift från tråd 2. Använd tre klasser:

- En klass för äpplet (och det grafiska gränssnittet)
- En klass för tråd 1 som ska ärva från klassen Thread
- En klass för tråd 2 som inte ska ärva från klassen Thread utan implementera gränssnittet Runnable

Respektive tråd ska i en loop i run-metoden kunna skriva ut meddelandet Utskrift från tråd 1 och utskrift från tråd 2 i programmets text-area. Utskriften ska göras en gång per sekund.

Vid tryck på den ena knappen ska tråd 1 aktiveras och texten på knappen ska ändras till Deaktivera utskrift från tråd 1. Ett tryck till på samma knapp ska deaktivera tråd 1 (det vill säga stoppa tråd 1 helt) och texten på knappen ska återställas till Aktivera utskrift från tråd 1. Motsvarande gäller för den andra knappen men för tråd 2.

De färdiga CLASS-filerna ska packas i en JAR-fil och HTML-dokumentet som hör till äpplet ska använda denna JAR-fil.

Exempel

Ett enkelt exempel kan köras från kommandoprompten med:

```
appletviewer http://atlas.dsv.su.se/~pierre/i/05_ass/ip1/a_1_1/threader.html
```

Tips

Vid implementering av gränssnittet Runnable så måste man deklarera och initiera en variabel för tråden:

```
Thread tread = new Thread(this);
```

Man måste givetvis även starta tråden (på ett lämpligt ställe) med:

```
tread.start();
```

Läs i API-beskrivningen under `java.lang.Thread` om varför stop-metoden med flera inte fungerar. Där finns bra tips på hur man stoppar en tråd.

Man får automatisk scrollning av en JTextArea med hjälp av en JScrollPane och lite annan kod:

```
JScrollPane scrollPane = new JScrollPane(textArea);
scrollPane.setHorizontalScrollBarPolicy(ScrollPaneConstants.HORIZONTAL_SCROLLBAR_NEVER);
textArea.append("T1 running\n");
textArea.setCaretPosition(textArea.getText().length());
```

Man måste göra ett nytt trådat objekt vid varje omstart, det går alltså inte att enbart köra start-metoden igen.

För den intresserade finns ett mer avancerat multi-trådat program för en enkel grafisk tillämpning, en (för datorn) [svår snöstorm](#), brr :-), där varje snöflinga är en egen tråd. Programmet ställer dock vissa krav på hårdvaran :-).

Man kan givetvis även starta ett äpple från en vanlig browser, det är ju det som är idén bakom äpplen. Man kan i huvudsak göra detta på två sätt:

- Via en i browsern inbyggd Java-tolk som finns i vissa browsrar. Detta är ett dåligt sätt eftersom det ofta inte stöds och i de fall det stöds så är det oftast en gammal version.
- Via »[Java plug-in](#)«:
 - Via ett [HTML-dokument som stöder även äldre browsrar](#). Man gör lättast ett sådant HTML-dokument med ett hjälpverktyg kallat »[HTMLConverter](#)«. Notera att HTML-dokumentet inte validerar som XHTML 1.1 efter användning av HTMLConverter.
 - Via ett [HTML-dokument som enbart stöder senare browsrar](#). Notera att HTML-dokumentet i detta fall validerar som XHTML 1.1.

Start från HTML: Web Start

Allmänt

Vi går vidare och tittar på det andra sättet att göra program som kan startas från ett HTML-dokument i en browser, Web Start. Fördelen med Web Start framför äpplen är att:

- Man kan använda fristående program
- Det är betydligt enklare när man vill göra ett program som ska kunna göra saker som man (av säkerhetsskäl) default inte kan göra med ett äpple (men som man med vissa svårigheter även kan göra med ett äpple)

Mer om detta på kursen Internetsäkerhet.

Uppgift

Gör om uppgiften [Start från HTML: Äpplen](#) till ett fristående program som kan köras igång med Web Start.

Exempel

Ett [enkelt exempel](#).

Tips

Testa först lokalt på den egna maskinen och använd då en JNLP-fil med ett codebase-attribut som använder file-protokollet:

```
<jnlp spec="1.0+" codebase="file:///C:/Documents and Settings/Administrator/My Documents/www/i/05_ass/
```

Byt till egen sökväg.

Testa sen via HTTP-servern på atlas.dsv.su.se eftersom HTTP-servern på www.dsv.su.se inte fungerar för detta (den inte är konfigurerad för MIME-typen application/x-java-jnlp-file). Nu ska denna rad ut så här:

```
<jnlp spec="1.0+" codebase="http://atlas.dsv.su.se/~pierre/i/05_ass/ip1/a_1_2" href="threder_web_star
```

Byt till eget användarnamn och egen sökväg.

Använd inte:

```
<offline-allowed/>
```

eftersom allt då cachas och man kan få svårt att ladda om programmet.

Högnivåkopplingar: Otolkad information

Allmänt

Vi ska i denna uppgift koppla oss till en speciell typ av server, en HTTP-server, ofta även kallad en webb-server. Vi ska använda en högnivåkoppling som är en koppling som kan prata det protokoll som används av en viss tjänst.

Uppgift

Gör ett fristående program med ett grafiskt användargränssnitt som implementerar en enkel browser. Användaren ska kunna ange en valfri HTTP-adress i ett text-fält och programmet ska hämta informationen på denna adress och presentera resultatet i en text-area. Själva hämtningen kan initieras när användaren trycker på return i text-fältet, på en knapp eller liknande.

När programmet startar ska det hämta och presentera information från den adress som finns default i textfältet från början. Observera! Informationen ska inte tolkas utan bara presenteras som ren text.

De färdiga CLASS-filerna ska packas i en JAR-fil och denna ska gå att köra med:

- `java -jar Browser.jar`

Exempel

Ett enkelt exempel kan köras enligt följande:

1. Hämta filen: [Browser.jar](#)
2. Kör programmet med: `java -jar Browser.jar`

Tips

Använd klassen `URL` och någon lämplig metod för ett objekt från denna klass, exempelvis metoden `openStream` vilket returnerar i en in-ström som man kan läsa från.

Högnivåkopplingar: Tolkad information

Allmänt

Vi fortsätter med högnivåkopplingar och tittar nu vidare på äpplen och dess begränsningar. I den här uppgiften skall vi bygga en browser som ett äpple och som kan tolka den information vi hämtar.

Uppgift

Gör samma sak som i uppgiften [Högnivåkopplingar: Otolkad information](#) men med följande modifikationer:

- Programmet ska vara ett äpple
- Den mottagna informationen ska tolkas, det vill säga inte enbart presenteras som ren text
- Länkarna ska fungera

Notera att det nu, av säkerhetsskäl, endast går att göra uppkopplingar till samma domän som äpplet laddades ner från. Mer om detta på kursen Internetsäkerhet.

De färdiga CLASS-filerna ska packas i en JAR-fil och HTML-dokumentet som hör till äpplet ska använda denna JAR-fil.

Exempel

Ett enkelt exempel kan köras från kommandoprompten med:

```
appletviewer http://atlas.dsv.su.se/~pierre/i/05_ass/ip1/a_2/browser_cool.html
```

Tips

Använd klassen `javax.swing.JEditorPane` (se API-beskrivningen).

Man kan testa sitt äpple under utvecklingen på två sätt. Man kan surfa till:

- En fil lokalt på samma dator:

- Protokoll: File
- Säkerhet: Filen måste komma från samma katalog som applet
- Exempel: `file:///C:/Documents and Settings/Administrator/My Documents/en_fil.html`
- En fil via en http-server från en annan dator:
 - Protokoll: HTTP
 - Säkerhet: Filen måste komma från samma domän som applet
 - Exempel: `http://atlas.dsv.su.se/en_fil.html`

Försöker man ladda en fil utanför det område beskrivet ovan så genereras en `Security Exception`.

För den intresserade så finns ett komplett Java-paket från »[IceSoft](#)« som man kan använda om man någon gång behöver en mer avancerad browser i ett Java-program. Det fungerar utmärkt och är mycket lätt att använda.

Lågnivåkopplingar: Stream sockets: Klient-sidan

Allmänt

Vi ska nu titta på den första huvudtypen av lågnivåkopplingar med sockets, stream sockets, som använder säkra men något långsamma (kontinuerlig-uppkoppling-typ) TCP vid transporten. Vi ska titta på klient-server paradigmen (men det är inget måste för denna typ av sockets) och börjar på klient-sidan.

Uppgift

Gör ett fristående program med ett grafiskt användargränssnitt som implementerar en chat-klient som kan sända och ta emot text-meddelanden. Chat-klienten ska koppla upp sig till en chat-server via en stream-socket. När en chat-klient sänder ett text-meddelande till chat-servern så sänder chat-servern detta text-meddelande till alla anslutna chat-klienter. Chat-klienten ska alltså både kunna:

- Sända text-meddelanden, inskrivet i exempelvis ett text-fält, till chat-servern
- Ta emot text-meddelanden från chat-servern och visa dessa text-meddelanden i en lämplig swing-komponent, exempelvis en text-area

Chat-klienten ska gå att startas på tre sätt:

1. `java Client` (då ska host default bli `127.0.0.1` och port default bli `2000`)
2. `java Client <host>` (då ska port default bli `2000`)
3. `java Client <host> <port>`

Efter detta ska chat-klienten försöka koppla sig till chat-servern. Om kopplingen lyckas så ska det i chat-klientens titelrad stå vilken server och port man är uppkopplad till. Om kopplingen inte lyckas så ska programmet avslutas.

De färdiga CLASS-filerna ska packas i en JAR-fil och denna ska gå att köra med:

- `java -jar Client.jar`

Frivillig utökning är att även ta med så att chat-klienten klarar av om kopplingen bryts till chat-servern (exempelvis om chat-servern avslutas) genom att exempelvis meddela användaren och avsluta sig själv.

Exempel

Ett enkelt exempel kan köras enligt följande:

1. Hämta filen: [Client.jar](#)
2. Kör programmet med: `java -jar Client.jar`

Observera! Chat-servern måste vara igång innan man startar chat-klienten, se nedan.

Tips

För att testa klienten måste man koppla den till en chat-server. Man kan antingen:

- Hämta en chat-server och köra den på den egna datorn:
 - Hämta filen [Server.jar](#)
 - Starta programmet med `java -jar Server.jar` som startar chat-servern på port 2000 men man kan även ange ett annat portnummer som ett tredje argument, exempelvis `java -jar Server.jar 2001`
- Använda en chat-server på en annan dator:
 - Host: `atlas.dsv.su.se`
 - Port: 9494

Man kan testa att en chat-server fungerar genom att telnetta till den, exempelvis:

- `telnet 127.0.0.1 2000`
- `telnet atlas.dsv.su.se 9494`

Använd klassen `Socket` och metoderna `getInputStream` och `getOutputStream` för att få strömmar att läsa och skriva från. Exempel på lämpliga ström-kopplingar:

```
in = new BufferedReader(new InputStreamReader(socket.getInputStream()));
out = new PrintWriter(socket.getOutputStream(), true);
```

Klienten bör vara multi-trådad. Hämtningen av meddelanden från servern bör ligga i en evig loop i metoden `run`.

När man stänger klienten så kan programmet låsa sig, testa då med att stänga out-strömmen före in-strömmen.

Chat-servern implementerar ett minimalt protokoll som man kan leka med. Om en chat-klient skickar `wwhhoo` så returnerar chat-servern alla uppkopplade chat-klienters IP-adresser (den inleder svaret med `wwHH00:` för varje chat-klient).

Finns det ingen att chatta med? Testa en pratstund med Pierre's syster Amy:

- Host: `atlas.dsv.su.se`
- Port: 4456

Det är en robot som pratar Engelska :-).

Som överkurs man även titta på paketet »[java.nio](#)«, exempelvis på klassen »[SocketChannel](#)«.

Lågnivåkopplingar: Stream sockets: Server-sidan

Allmänt

Vi fortsätter med lågnivåkopplingar med stream sockets men nu på server-sidan.

Uppgift

Gör ett fristående program med ett grafiskt användargränssnitt som implementerar en chat-server som kan sända och ta emot text-meddelanden. Chat-servern ska använda stream-sockets.

Chat-servern ska kunna ta emot begäran om uppkoppling från nya chat-klienter och placera alla chat-klienter i en egen tråd för att kunna ta emot ytterligare chat-klienter. Varje chat-klient-tråd ligger och väntar på att en chat-klient ska skicka ett text-meddelande. När så sker så ska chat-servern "broadcasta" detta meddelande till samtliga chat-klienter. Chat-servern måste således hålla reda på alla anslutna chat-klienter i någon datasamling.

Alla text-meddelanden som sänds från chat-klienterna ska visas en en lämplig swing-komponent, exempelvis en text-area. För samtliga text-meddelanden så ska följande skrivas ut i denna swing-komponent:

- Chat-klientens IP-adress
- Text-meddelandet från chat-klienten

Chat-servern ska gå att starta på två sätt:

1. `java Server` (då ska port default bli 2000)
2. `java Server <port>`

Efter detta ska chat-servern göra sig redo för att ta emot förfrågningar om anslutning från chat-klienter. Om detta lyckas så ska det i chat-servers titelrad stå den host och port som chat-servern kör på samt antalet chat-klienter som är uppkopplade.

Observera! Chat-servern ska upptäcka när en klient inte längre är ansluten och då ta bort den ur datasamlingen och döda tråden (eventuellt kan tråden döda sig själv).

De färdiga CLASS-filerna ska packas i en JAR-fil och denna ska gå att köra med:

- `java -jar Server.jar`

Exempel

Se chat-servern som användes i uppgift [Lågnivåkopplingar: Stream sockets: Klient-sidan](#).

Tips

Använd klassen `ServerSocket` och metoden `accept`.

Man måste vara försiktig med krokar. Man har ju ofta flera trådar igång för hantering av anslutna chat-klienter och det kan exempelvis bli fel om man samtidigt (1) tar bort en chat-klient-tråd och (2) broadcastar ett meddelande. För att lösa detta bör man deklarera vissa metoder i chat-servern som `synchronized`.

Hur upptäcker chat-servern att en chat-klient har kopplat ner? För varje chat-klient som kopplar upp sig till chat-servern så skapar chat-servern en egen tråd för chat-klienten. Antag att klassen som hanterar chat-klienterna heter `ClientHandler`. `ClientHandler` är en tråd som hela tiden ligger och lyssnar på om klienten har skickat något meddelande. Om ett meddelande kommer från en chat-

klient så anropar `ClientHandler` en metod i chat-servern som broadcastar meddelandet till alla anslutna chat-klienter. Det är i denna lyssningsfas som `ClientHandler` kan upptäcka om en klient har kopplat ner:

1. Om chat-klienten beter sig korrekt och stänger sin utström returnerar läsning med (`BufferedReader`) `readLine`-metoden `null`. Klienten under uppgift [Lågnivåkopplingar: Stream sockets: Klient-sidan](#) gör detta om sänder `exit` som meddelande.
2. Om chat-klienten inte beter sig korrekt och inte stänger sin utström blir det `IOException` vid läsning med (`BufferedReader`) `readLine`-metoden. Klienten under uppgift [Lågnivåkopplingar: Stream sockets: Klient-sidan](#) gör detta om man stänger fönstret för klienten.

Man får då undersöka dessa båda fall och om något av dem uppstår så ska man gå ut ur den "eviga" lyssnar-loopen, exempelvis:

```
public void run() {
    String msg;
    try {
        while ((msg = reader.readLine()) != null) {
            //... gör något med msg ...
        }
        writer.close();
        reader.close();
        socket.close();
    } catch (IOException e) {
    }
    server.killThread(this);
}
```

Precis efter try-blocket ligger satserna som stänger alla strömmar, och eftersom de genererar `IOException` kan de ju lika gärna ligga i try-blocket. Om ett `IOException` har inträffat tidigare kan man ju antagligen inte stänga strömmarna i alla fall.

Exempel på lämpliga ström-kopplingar till klient-socketen:

```
in = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));
out = new PrintWriter(clientSocket.getOutputStream(), true);
```

Hur får man reda på server-socketens host-adress:

```
String host = serverSocket.getInetAddress().getLocalHost().getHostName();
```

Hur får man reda på klient-socketens host-adress:

```
String host = socket.getInetAddress().getHostName();
```

Som överkurs kan man titta på paketet »[java.nio](#)«, exempelvis på klassen »[ServerSocketChannel](#)«.

Lågnivåkopplingar: Datagram sockets: Unicast

Allmänt

Vi ska nu titta på den andra huvudtypen av lågnivåkopplingar med sockets, datagram sockets, som använder osäkra men snabba (skicka-post-typs) UDP vid transporten. Vi ska titta på peer-to-peer paradigmen (men det är inget måste för denna typ av sockets).

Uppgift

Gör ett fristående program med ett grafiskt användargränssnitt som implementerar ett enkelt distribuerat ritprogram (whiteboard). Allt som ritas ska sändas med UDP till ett annat program identiskt som ritar upp exakt samma bild. Det andra programmet ska fungera på samma sätt.

De färdiga CLASS-filerna ska packas i en JAR-fil och denna ska gå att köra med:

```
java -jar Draw.jar <my port> <remote host> <remote port>
```

Exempel

Ett enkelt exempel kan köras enligt följande:

1. Hämta filen: [Draw.jar](#)
2. Kör igång programmet med: `java -jar Draw.jar <my port> <remote host> <remote port>`

Om man vill testa på samma maskin kan man exempelvis köra igång två program:

- `java -jar Draw.jar 2000 localhost 2001`
- `java -jar Draw.jar 2001 localhost 2000`

Men det är roligare om man testar två program på två olika datorer.

Tips

Utgå gärna från detta [enkla ritprogram](#) (men det är inget krav). Bygg gärna ut programmet så det fungerar med olika färger, olika tjocklek på penseln, olika figurer och så vidare (men det är inget krav).

Som överkurs kan man titta på paketet »[java.nio](#)«, exempelvis på klassen »[DatagramChannel](#)«.

Lågnivåkopplingar: Datagram sockets: Multicast

Allmänt

Vi ska nu fortsätta titta på den andra huvudtypen av lågnivåkopplingar med sockets, datagram sockets, men på en sub klass kallad multicast socket som används vid kommunikation en-till-många. Vi ska nu även använda ett egendefinierat applikationsnivå-protokoll.

Man använder multicast-adresser när man vill sända saker till flera användare. Multicast är bra om vi exempelvis vill sända ut en film från en server till många tittare. Utan multicast så skulle bandbredden som krävs bli direkt proportionell mot antalet användare. Med multicast så blir bandbredden lika stor som för en enda film, oberoende av antalet tittare. En nackdel med multicast är att filmerna måste gå vissa tider, det vill säga, de levereras inte on-demand, men detta kan ju enkelt approximeras genom att sända filmen på olika multicast adresser med vissa tids-intervall.

Uppgift

Gör ett fristående program med ett grafiskt användargränssnitt som gör följande:

1. Sänder upprepade text-meddelanden på en multicast-IP-adress som enbart berättar att man är uppkopplad

2. Tar emot text-meddelanden från en multicast-IP-adress som visar vilka som är uppkopplade nu

(1) Text-meddelanden som sänds ska sändas 1 gång per sekund och vara formaterade enligt ett egetkonstruerat protokoll, vi kallar det OTTP, för Online Text Transfer Protocol :-):

```
send(" From: " + name + " Host: " + host + " Comment: " + comment);
```

Här är:

- send en egendefinierad metod som sänder meddelandet givet som argument till metoden
- name en variabel som anger namnet på användaren
- host en variabel som anger användarens host
- comment en variabel som anger en kommentar från användaren

Det är viktigt att man inte sänder meddelandena oftare än 1 gång per sekund då det i UDP inte finns någon inbyggd "trafikövervakning". Sänder man med UDP för ofta kan nätverket bli överbelastat och gå ner.

(2) Text-meddelanden som tas emot ska parsas och för varje namn/host kombination så ska informationen visas på följande format så länge denna användare är uppkopplad:

```
name --- host --- comment
```

Endast en rad för varje användare ska visas. Om programmet tar emot ett meddelanden som inte följer OTTP-protokollet så ska det skriva ut `COULD NOT PARSE MESSAGE` i text-arean.

De färdiga CLASS-filerna ska packas i en JAR-fil och denna ska gå att köra med:

```
java -jar OnLine.jar <name> <comment>
```

Efter detta ska programmet koppla sig till multicast-adressen 234.235.236.237 på port 2000. Om kopplingen lyckas så ska det i programmets titelrad stå vilken IP-adress (och port) uppkopplingen skett till. Om kopplingen inte lyckas så ska programmet avslutas.

Exempel

Ett enkelt exempel kan köras enligt följande:

1. Hämta filen: [OnLine.jar](#)
2. Kör igång programmet med: `java -jar OnLine.jar <name> <comment>`

Kör gärna igång flera program på flera olika datorer. Tänk dock på att multicast är en ny teknik som för nuvarande fungerar bäst i intranät. Vill man använda denna teknik på Internet kan man använda Mbone som är ett nätverk av routrar som implementerar olika multicast-protokoll.

Tips

Använd exempelvis fyra klasser:

- En klass `onLine` för det grafiska gränssnittet och initialiseringen av `MulticastSocket`
- En inre trådad klass `Sender` för kontinuerlig sändning av `DatagramPacket` meddelanden en gång per sekund
- En inre trådad klass `Receiver` för kontinuerlig mottagning av `DatagramPacket` meddelanden där mottagna meddelanden (det vill säga texten) läggs i en datasamling (exempelvis ett `HashSet`) om de inte finns i datasamlingen tidigare
- En inre trådad klass `InterfaceHandler` för kontinuerlig uppdatering av gränssnittet var 5:e

sekund:

- Töm text-arean
- Parsa de meddelanden som finns i datasamlingen och lägg in resultaten i text-arean
- Töm datasamlingen

Har man implementerat detta med flera trådar som möjligtvis delar på någon resurs så kan krockar uppstå och man bör således deklarerera vissa metoder som `synchronized` och/eller använda de synkroniserade datasamlingarna. I exemplet har dock inget av detta använts.

För att dela upp ett inkommet meddelande i enstaka ord så är klassen `java.util.StringTokenizer` utmärkt.

Notera att:

- `multicastSocket.getInetAddress()` ger värdet `null`,
- `multicastSocket.getPort()` ger värdet `-1`

och att det är som det ska, bara att gå vidare med programmeringen :-)

Man måste skapa ett nytt `DatagramPacket` för varje meddelande som ska tas emot.

Man måste ha stöd för multicast-routing i sin dator eller hos den ISP som man kopplar upp sig mot. Detta kan testas med:

```
ping all-routers.mcast.net
```

Får man svaret:

```
all-routers.mcast.net is alive
```

så har man stöd. Hur gör man annars? Förslag:

- Arbeta på DSV (= dålig lösning på en distanskurs).
- Arbeta ej ansluten till Internet och konfigurera för multicast:
 - Windows 2000/XP:
 - Med nätverks-kort: Använd registry-editorn (`C:\WINNT\REGEDIT:EXE`) och lägg till `DisabledDHCPMediaSense REG_DWORD 0x00000001 (1)` här `HKEY_LOCAL_MACHINE / SYSTEM / CurrentControlSet / Services / Tcpip / Parameters`
 - Utan nätverks-kort: För nuvarande så stöds inte multicast-adresser i loopback adaptorn men man kan använda adressen `127.0.0.1` och testa lokalt med bara ett program och sen när det fungerar så kan man testa med adressen `234.235.236.237` uppkopplad mot DSV med flera program
 - Unix/Linux: Se kapitlet [IP Multicast Host Configuration](#) (PDF-format)
- Arbeta ansluten till Internet mot DSV:
 - Kör telnet eller SSH till DSV och gör [uppgiften utan GUI](#)
 - Använd »[X-Win32](#)« (eller motsvarande) och sätt `DISPLAY` till sitt IP-nummer: `export DISPLAY=xxx.xxx.xxx.xxx:0`

Har man konfigurerat sin dator för multicast så betyder inte det att man kan använda multicast utanför datorn eftersom många ISP'er inte använder multicast, exempelvis så stöder inte Telia multicast. Då kan man således bara använda sina multicast-program lokalt, vilket givetvis är trist. En lösning till det är att använda en tunnel till en dator som använder multicast (och helst är ansluten till MBone). På `atlas.dsv.su.se` kör nu en tunnel-server för multicast som man kan ansluta till genom:

- Hämta filen: »[Client.jar](#)«
- Kör med: `java -jar Client.jar <multicast-adress> <multicast-port> <ttl>`

Kör man utan argument så används:

- Multicast-adress: 234.235.236.237
- Multicast-port: 2000
- TTL: 1

Tunnel-klienten och tunnel-servern är gjord helt i Java och bygger på:

- Tunnel-servern:
 - Från MBone till tunnel-klienten:
 - Ta emot multicast-meddelanden från MBone på en multicast-socket
 - Sänd alla mottagna meddelanden till tunnel-klienten via en datagram-socket
 - Från tunnel-klienten till MBone:
 - Ta emot datagram-meddelanden från tunnel-klienten på en datagram-socket
 - Sänd alla mottagna meddelanden till MBone på en multicast-socket
- Tunnel-klienten:
 - Från program hos klienten till tunnel-servern:
 - Ta emot multicast-meddelanden från andra program hos klienten på en multicast-socket
 - Sänd alla mottagna meddelanden till tunnel-servern via en datagram-socket
 - Från tunnel-servern till program hos klienten:
 - Ta emot datagram-meddelanden från tunnel-servern på en datagram-socket
 - Sänd alla mottagna meddelanden till program hos klienten på en multicast-socket

Klienten och servern använder även stream-sockets för sessionshanteringen.

Med en tunnel kan man alltså använda multicast även utanför sin egen dator. Om den dator som tunnel-servern kör på inte är ansluten till MBone (som är ett nätverk av routrar som stöder multicast) så kräver det dock att alla klienter som vill kommunicera via multicast använder en tunnel till samma tunnel-server och då är vi ju tillbaka till vanlig unicast-kommunikation. Det bästa är givetvis om den dator som tunnel-servern kör på är ansluten till MBone, då behöver inte alla klienter använda samma tunnel-server och det blir mer likt verklig multicast. Tyvärr så är inte atlas eller andra datorer på DSV anslutna till MBone (men det kanske kommer senare), något som kan testas med:

- Hämta filen: [MBone.class](#)
- Kör med: `java MBone <multicast-adress> <multicast-port>`

Kör man utan argument så används:

- Multicast-adress: all-systems.mcast.net
- Multicast-port: 4000
- TTL: 1

Händer det inget så är man inte ansluten till MBone.

Lågnivåkopplingar: Raw sockets

Allmänt

Vi ska nu titta på en typ av sockets, raw sockets, där man själv kan specifi- ka allt om hur paketen man sänder iväg ska se ut. Dessa typer av sockets finns av säkerhetsskäl inte med i Java utan man måste anropa (via JNI) program skrivna i C. Det här är mycket svårt eftersom man måste kunna allt på en låg nivå (och läsa RFC'er med mera) men man kan bygga många användbara, roliga, spännande och farliga :-) saker.

Uppgift

Gör ett program som implementerar ping, traceroute eller något annat nätverksprogram som använder ICMP eller något som man inte kan göra i standardversionen av Java. Man ska inte anropa motsvarande funktioner i operativsystemet via exempelvis:

```
Runtime.getRuntime().exec("myprog.exe")
```

utan man ska använda ett Java-paket som anropar C-kod som implementerar raw sockets.

Exempel

Ett enkelt exempel för Win32 kan köras enligt följande:

1. Hämta filen: [jicmp.dll](#)
2. Hämta filen: [jicmp.jar](#)
3. Hämta filen: [Ping.class](#)
4. Kör programmet med:
 - o Unix/Linux: Detta kräver lite »[konfiguration](#)«
 - o Win32: java -classpath jicmp.jar;. Ping <host>

Med J2SE 5.0 kan man göra **vissa** saker utan tillbehör:

1. Hämta filen [Ping.class](#)
2. Kör programmet med: java Ping <host>

Detta program är gjort med hjälp av klassen `InetAddress` och metoden `isReachable`.

Tips

Använd ett redan existerande API, exempelvis »[Jpcap](#)«.

Databaskopplingar

Allmänt

Denna uppgift belyser hur man kopplar sig till en databas-server via JDBC på Internet.

Uppgift

Gör ett äpple som implementerar en gästbok där användaren ska kunna fylla i följande fält:

- o Namn
- o E-post
- o Hemsida
- o Kommentar

Användarens inlägg ska sparas i databas-servern MySQL som kör på `atlas.dsv.su.se`. Alla inlägg ska visas i en text-area. All text ska kontrolleras så att den inte innehåller HTML-kod och om den gör det så ska texten bytas ut mot exempelvis ordet `censur`.

De färdiga CLASS-filerna ska packas i en JAR-fil och HTML-dokumentet som hör till äpplet ska använda denna JAR-fil.

Exempel

Ett enkelt exempel kan köras från kommandoprompten med:

```
appletviewer http://atlas.dsv.su.se/~pierre/i/05_ass/ip1/a_4/sql_guestbook.html
```

Tips

Man behöver här ett:

- JDBC-gränssnitt mot MySQL [mm.mysql-2.0.14-bin.jar](#)

Packa upp JAR-filen i katalogen för denna uppgift.

Anslutningen görs med:

```
Class.forName("org.gjt.mm.mysql.Driver").newInstance();  
Connection dbConnection = DriverManager.getConnection(url, username, password);
```

där:

- url = "jdbc:mysql://" + computer + "/" + db_name;
- computer = atlas.dsv.su.se
- db_name, username och password finns under [MySQL](#).

När man ska skapa en JAR-fil med alla CLASS-filer inkluderat de från filen mm.mysql-2.0.14-bin.jar så gör man enligt följande:

```
jar cf SQLGuestbook.jar *.class org
```

Paketet java.util.regex är användbart vid kontrollen av HTML-kod. Man kan exempelvis använda:

```
Pattern p = Pattern.compile("<.*>");
```

Notera att ett äpple bara får koppla sig tillbaka till samma plats som det kommer från. Använder man DSV's MySQL-databasserver så måste äpplet ligga på DSV Unix/Linux-kontot och laddas via atlas.dsv.su.se:

```
appletviewer http://atlas.dsv.su.se/~<användarnamn>/<sökväg>/<htmlfil>
```

E-post: Sändning

Allmänt

Denna uppgift belyser hur man implementerar sändning av e-post.

Uppgift

Gör ett fristående program med ett grafiskt användargränssnitt som med hjälp av JDK-extensionen Javamail kan sända e-post. Användaren ska kunna skriva in följande:

- Mail-server
- Från
- Till

- Ämne
- Meddelande

Exempel

Ett enkelt exempel kan köras enligt följande:

- Hämta filen: [mailSender.jar](#)
- Kör programmet med: `java -jar mailSender.jar`

Tips

Eftersom vi använder ett tillbehör till Java så behöver man följande extra JAR-filer:

- [activation.jar](#)
- [mail.jar](#)

Om man inte vill sätta classpath till dessa så kan de packas upp med:

```
jar xf mail.jar
jar xf activation.jar
```

E-post: Mottagning

Allmänt

Denna uppgift belyser hur man implementerar mottagning av e-post.

Uppgift

Gör ett fristående program med ett grafiskt användargränssnitt som med hjälp av JDK-extensionen Javamail kan mottaga e-post. Användaren ska kunna skriva in följande:

- Mail-server
- Användare
- Password

Exempel

Ett enkelt exempel kan köras enligt följande:

- Hämta filen: [mailReceiver.jar](#)
- Kör programmet med: `java -jar mailReceiver.jar`

Tips

Se tips för uppgiften [E-post: Sändning](#).

Mediaöverföring: Talsyntes

Allmänt

Denna uppgift belyser hur man använder Java Speech för att åstadkomma talsyntes i Internet-baserade program.

Uppgift

Bygg ut någon av de tidigare uppgifterna så att de fungerar med talsyntes, exempelvis:

- En browser som läser upp texten i HTML-sidor
- En chat klient som läser upp mottagna meddelanden
- En e-post-klient som läser upp e-brev och även meddelar användaren när man får nya brev (med avsändare och ärende)

Exempel

Ett exempel på chat-klient kan köras enligt följande:

Hämta filerna:

- [cmu_time_awb.jar](#)
- [cmu_us_kal.jar](#)
- [cmudict04.jar](#)
- [cmulex.jar](#)
- [cmutimelex.jar](#)
- [en_us.jar](#)
- [freetts.jar](#)
- [jsapi.jar](#)
- [Client.class](#)

Kör programmet med:

- Unix/Linux: `java -classpath cmu_time_awb.jar:cmu_us_kal.jar:cmudict04.jar:cmulex.jar:cmutimelex.jar:en_us.jar:freetts.jar;j:Client <host> <port>`
- Win32: `java -classpath cmu_time_awb.jar;cmu_us_kal.jar;cmudict04.jar;cmulex.jar;cmutimelex.jar;en_us.jar;freetts.jar;j:Client <host> <port>`

Om man inte anger host och/eller port så kopplar sig chat-klienten till:

- Host: atlas.dsv.su.se
- Port: 9494

Detta är en vanlig chat-klient som är utbyggd med hjälp av »[Java Speech API](#)« som enbart fungerar som textuppläsare eftersom talförståelsedelen av API't inte är färdig än. Man kan givetvis även köra programmet mot en annan server, exempelvis Amy:

- Host: atlas.dsv.su.se
- Port: 4456

Rösten är dock inte helt optimerad för henne i detta exempel :-).

Tips

Eftersom Java Speech API är en standard extension så måste man givetvis använda en mängd JAR- och CLASS-filer (med mera).

För att användaren ska slippa använda filen `speech.properties` i användarens root-katalog (som man måste göra i [HelloWorld1.java](#)) så kan man göra enligt [HelloWorld2.java](#).

Mediaöverföring: Ljud över stream sockets

Allmänt

Vi ska nu gå vidare med media och titta på »[Java Sound](#)« som är mycket användbart i många olika program. Java Sound består av två huvuddelar:

- `javax.sound.sampled`
- `javax.sound.midi`

Vi ska fokusera på `javax.sound.sampled` paketet och börja med att titta på hur man i Internet-baserade program:

1. Spelar in ett ljud
2. Sänder ett ljud med TCP
3. Tar emot ett ljud med TCP
4. Spelar upp ett ljud

Uppgift

Bygg ut någon av de tidigare uppgifterna så att de använder samplat ljud över stream sockets, exempelvis:

- En chat klient som sänder och tar emot ljud
- En e-post-klient för ljudmeddelanden
- En Internetbaserad telefonsvarare där besökare kan spela in ljud som sedan sparas i en SQL-databas där ägaren sedan via en annan (hemlig) sida kan lista alla ljudmeddelanden och lyssna av dem
- En SQL-gästbok för ljud

Exempel

Ett exempel på en icke kontinuerlig chat-klient kan köras enligt följande:

1. Hämta filen: [Client.jar](#)
2. Kör programmet med: `java -jar Client.jar`

Detta är en vanlig chat-klient som är utbyggd med hjälp av Java Sound API. Chat-klienten kopplar sig default till:

- Host: `atlas.dsv.su.se`
- Port: `4949`

På denna adress kör en vanlig chat-server som är något omgjord så att den hanterar objekt istället för strängar (via `ObjectInputStream` och `ObjectOutputStream`). Man kan testa systemet lokalt om man har en långsam uppkoppling:

- Hämta filen: [Server.jar](#) (utan GUI)
- Kör chat-server programmet: `java -jar Server.jar`
- Kör chat-klient programmet: `java -jar Client.jar localhost`

Notera att man måste ha en mikrofon och högtalare anslutna till datorn för att man ska kunna testa detta (det går att ansluta detta till datorerna på DSV).

Tips

Java Sound API ingår numera i JDK så man behöver inga extra filer. Vill man göra ett äpple körbart i en browser så måste man signera äpplet (mer om det på kursen Internetsäkerhet) eftersom äpplen default bara får spela upp ljud, inte spela in ljud.

Här finns några exempel som man kan titta på för att komma igång:

- Spela in ljud och spara till en fil (på AU-format):
 - Källkod: [FileRecorder.java](#)
 - Användning: `java FileRecorder <ljud.au>`
- Spela upp ljud från en fil (på AU-format):
 - Källkod: [FilePlayer.java](#)
 - Användning: `java FilePlayer <ljud.au>`
- Spela in och upp ljud och spara ljudet internt i ett serialiserbart objekt storage:
 - Källkod:
 - [RecordAndPlay.java](#)
 - [Storage.java](#)
 - Användning: `java RecordAndPlay <ljud.au>`

När man använder ljud över Internet så är det mycket viktigt att man sänder så lite information som möjligt, speciellt om det finns användare som är uppkopplade via modem. Hur gör man det? Man använder två olika sätt:

- Sampla ljudet med:
 - Färre samples per tidsenhet, 8KHz är tillräckligt för tal
 - Färre bitar för varje sample
 - Färre kanaler, använd mono
- Komprimera ljudet innan sändning och dekomprimera det efter mottagning:
 - Metoder specialiserade för ljud komprimerar ofta så att information försvinner och det finns många olika metoder för detta (många olika codecs):
 - En del specialiserade på talat ljud:
 - ALAW är en standard som används i telefonerna i Europa med mera, denna codec ingår i JDK och halverar mängden information men ljudet blir av ganska dålig kvalitet
 - ULAW är en standard som används i telefonerna i USA och Japan, denna codec ingår i JDK och halverar mängden information men ljudet blir av ganska dålig kvalitet
 - GSM är en standard som används av mobil-telefoner, denna codec ingår inte i JDK och informationen blir ca 10 gånger mindre och ljudet blir av ganska bra kvalitet
 - En del specialiserade på musik: MP3, MP4 med flera
 - Metoder som är generella, exempelvis ZIP, komprimerar ofta så att inte information försvinner

Notera att de flesta ljudkort inte har direkt stöd för olika codecs utan att man måste omkoda så att man alltid använder PCM längst fram (input och output). Vill man använda GSM, MP3 eller andra codecs som inte ingår i JDK så måste man använda tillbehörs-filer, exempelvis från »[Tritonus](#)«. Man kan själv testa följande exempel för att se hur mycket man kan komprimera ljudet:

- Hämta filen: [Client.jar](#)
- Kör programmet med: `java -jar Client.jar`

Default så startar detta program med GSM- och ZIP-koprimering/dekomprimering mot:

- Host: atlas.dsv.su.se
- Port: 4949

Man kan köra med följande varianter på komprimeringar:

- Kör med: java -jar Client.jar atlas.dsv.su.se 4949 alaw
- Kör med: java -jar Client.jar atlas.dsv.su.se 4949 ulaw
- Kör med: java -jar Client.jar atlas.dsv.su.se 4949 gsm
- Kör med: java -jar Client.jar atlas.dsv.su.se 4949 alaw no_zip
- Kör med: java -jar Client.jar atlas.dsv.su.se 4949 ulaw no_zip
- Kör med: java -jar Client.jar atlas.dsv.su.se 4949 gsm no_zip

I både detta exempel och i det tidigare (utan codec och ZIP) används:

```
AudioFormat format = new AudioFormat(AudioFormat.Encoding.PCM_SIGNED, 8000.0F, 16, 1, 2, 8000.0F, false);
```

Det vill säga 16 bitars ljud samplat med 8 KHz i mono. Om man använder codec och ZIP så får man följande siffror i ett exempel där man räknar sakta från ett till tio:

- Utan codec och utan ZIP: 176000 Byte.
- Med ALAW-codec och utan ZIP: 88024 Byte.
- Med ALAW-codec och med ZIP: 45462 Byte.
- Med GSM-codec och utan ZIP: 18150 Byte.
- Med GSM-codec och med ZIP: 15965 Byte.

Man komprimerar alltså som mest 11 ggr, vilket ju är enormt bra!

För GSM-codecen så använder programmet två JAR-filer:

- [tritonus_share.jar](#)
- [tritonus_gsm.jar](#)

Dessa finns redan inbakade i client.jar så man behöver dem inte för att testköra. Vill man själv baka in allt i en JAR-fil så gör enligt följande:

- Packa upp tritonus_share.jar: jar xf tritonus_share.jar
- Packa upp tritonus_gsm.jar: jar xf tritonus_gsm.jar
- Kopiera katalogen META-INF\services till arbetskatalogen
- Jara ihop allt: jar cmfv mainClassClient Client.jar *.class META-INF\services* org

Det är ganska svårt att använda olika codecs men i stora drag så gör man så här när man encodar:

```
byte[] audio = out.toByteArray();
InputStream input = new ByteArrayInputStream(audio);
AudioInputStream ais1 = new AudioInputStream(input, format1, audio.length / format.getFrameSize());

AudioInputStream ais2 = AudioSystem.getAudioInputStream(encoding2, ais1);
ByteArrayOutputStream out2 = new ByteArrayOutputStream();
int nWrittenFrames = AudioSystem.write(ais2, fileType, out2);
```

Här ligger alla ljuddata initialt i out (som är ett objekt av typ ByteArrayOutputStream) och exempelvis:

```
AudioFormat.Encoding encoding1 = AudioFormat.Encoding.PCM_SIGNED;
AudioFormat format = new AudioFormat(encoding1, 8000.0F, 16, 1, 2, 8000.0F, false);
AudioFormat.Encoding encoding2 = Encodings.getEncoding("GSM0610");
AudioFileFormat.Type fileType = AudioFileTypes.getType("GSM", ".gsm");
```

Nu finns alla encodade ljuddata i out2. När man ska decoda ljuddata så gör man:

```
byte[] audio = out.toByteArray();
InputStream input = new ByteArrayInputStream(audio);
AudioInputStream aisTmp = AudioSystem.getAudioInputStream(input);
AudioInputStream ais = AudioSystem.getAudioInputStream(format, aisTmp);
```

Om man vill använda den server som finns på:

- Host: atlas.dsv.su.se
- Port: 4949

så sänd ljudet i objekt av typen `Storage`. Använd exempelvis `ObjectInputStream` och `ObjectOutputStream` kopplade till socketen och sänd och ta emot sådana (serialiserade) `Storage`-objekt som innehåller ljudet.

Om man försöker använda ovanstående program för att kontinuerligt sända och ta emot "klumpar" av ljud utan att man behöver klicka på några knappar så blir ljudet mycket dåligt och uppdelat (även om man enbart pratar med sig själv :-). En lösning är då att man har en speciell server som mixar ihop alla inkommande ljudströmmar till en enda ljudström (med klassen `Mixer`) och sen sänder ut denna till alla klienter.

Man kan även experimentera med ljudfilter och lägga på roliga effekter som exempelvis eko. Ett mycket enkelt exempel:

- Hämta filen: [RecordAndPlay.jar](#)
- Kör med: `java -jar RecordAndPlay.jar`

Med det programmet kan man testa fyra stycken enkla ljudprocessorer och alla kombinationer av dessa. Notera att ljudet sparas på två byten när man använder 16-bitars ljud och att det är lättare att använda en `short`-array än en `byte`-array om man samplat med 16-bitar och ska transformera ljudet på så här låg nivå. Det finns två metoder `short2byte` och `byte2short` och ett par exempel på användning av dem i källkoden (se klassen `DSP.java`):

- [RecordAndPlay.java](#)
- [Storage.java](#)
- [DSP.java](#)

Vill man lära sig göra avancerade ljudfilter och ljudprocessorer så är boken "Digital Audio with Java" skriven av Lindley utgiven av Prentice Hall bra:

- Ljudfilter:
 - Low-pass filters
 - High-pass filters
 - Band-pass filters
- Ljudprocessorer:
 - Amplitude adjust processor
 - Cache processor
 - Chorus/Flanger processor
 - Compressor/Expander/Limiter/Noise gate processor
 - Delay processor
 - Distortion processor
 - Graphic equalizer processor
 - Panner processor
 - Parametric equalizer processor
 - Phaser processor
 - Pitch shifter processor
 - Reverb processor

Notera dock att den är något gammal (1999) och därmed inte utgår från Java Sound men man kan ändå använda det som beskrivs tillsammans med Java Sound. Alltså, fokus är på transformering av ljud, inte teknikerna i Java Sound.

Mediaöverföring: Ljud över datagram sockets

Allmänt

Denna uppgift belyser hur man använder »[Java Sound](#)« och UDP för att spela in och upp ljud i Internet-baserade program. Vi ska titta på hur man kan göra en IP-telefon.

Uppgift

Gör en chat-klient som sänder och tar emot ljud kontinuerligt, det vill säga en IP-telefon. Några möjliga utvidgningar:

- Telefonnummer: En användare känner sällan till sin egen IP-adress och man kan underlätta för användarna genom att de kan ringa till varandra enbart genom att ange motpartens e-postadress. Programmet kan då använda Javamail och sända den egna IP-adressen till motparten och när motparten svarar med sin IP-adress via e-post så kan kommunikationen börja.
- Telefonkatalog: Man kan ha en telefonkatalog med sina vänner och i den kan man se direkt vilka som är online och då bara klicka på deras telefonnummer för att ringa upp.
- Telefonen på: Om programmet alltid startas upp när användaren loggar in så kan någon ringa upp användaren "när som helst".
- Telefonsvarare: Programmet kan känna av om nya e-brev kommer in under samtalets gång och upplysa användaren att en viss person försöker komma fram. Användaren kan då växla mellan samtalen eller koppla in den nya personen i samtalet. Man kan också tänka sig att man i ett sådant läge möts av en telefonsvarare, där man kan ge olika meddelanden skraddarsydda till olika e-postadresser.

Exempel

Ett exempel på en kontinuerlig chat-klient, det vill säga en IP-telefon, kan köras enligt följande (på två olika datorer):

- På dator 1:
 - Hämta filen: [Phone.jar](#)
 - Kör programmet med: `java -jar Phone.jar`
- På dator 2:
 - Hämta filen: [Phone.jar](#)
 - Kör programmet med: `java -jar Phone.jar <IP-nummer för dator 1>`

Programmen kommunicerar direkt (P2P, det vill säga inte via en server).

Tips

Det kan vara svårt att hitta rätt storlek på datagram-paketen som ska sändas och tas emot för att få ett ljud som:

- Inte är fördröjt för mycket
- Av bra kvalitet utan avbrott

Här får man vara beredd på en del experimenterande och en lösning vore om användaren själv kan ställa in detta och själv hitta rätt avvägning.

Mediaöverföring: Video med RTP

Allmänt

Vi ska nu gå vidare med media och titta på »[Java Media Framework \(JMF\)](#)« som är en JDK-extension som möjliggör att man kan koppla programmen till datorns mikrofon, kamera, vidoekamera med mera och sända data från dessa saker över nätet (både i klump och streamat). Med detta API kan man göra mycket intressanta tillämpningar, exempelvis ett video-baserat chat-program.

Uppgift

Gör en video-chat som använder JMF. Använd gärna RTP-protokollet.

Exempel

Inget exempel ges.

Tips

Inget tips ges.

Gesällprov

Allmänt

Gesällprovet är en uppgift för den egna kreativiteten och är ett fritt valt (men obligatoriskt) arbete där du får chansen att visa vad du går för konstruktionsmässigt (istället för minnesmässigt som på en tenta) i "lugn och ro".

Uppgift

Följande krav måste uppfyllas:

- Ska baseras på de tekniker som kursen bygger på men får givetvis även innehålla tekniker från andra områden
- Ska vara nätverksbaserat
- Ska vara bra både till både form och funktion
- Ska inte vara en av de tidigare uppgifterna på kursen men gärna en utökning av en eller flera av dessa

Tips

Huvudsaken är att man gör något som är kul! Några exempel:

- HTTP:
 - Bättre browser
 - RSS-samlare
- Chat:

- IRC-chat
 - Använd Java Media Framework (JME) för att göra ett chat-system som fungerar med ljud
- Multicast:
 - En tunnel för de som har en ISP som inte stöder multicast
- E-post:
 - Ett komplett e-post system som kan sända/ta emot bifogade filer med mera
 - Mail2Phone: ett program på server-sidan som med jämna mellanrum (med hjälp av crontab) kopplar sig till en mail-server och sänder nya brev vidare till en mobiltelefon
- News:
 - NNTP-klient för att hämta news
 - NNTP-server för att serva news
- FTP:
 - FTP-klient för att sända och ta emot filer
 - FTP-server för att sända och ta emot filer
- Telnet:
 - Telnet-klient
 - Telnet-server
- Spel:
 - Ett roligt nätverksbaserat spel

Använd gärna chat-servern (och ett eget protokoll) på `atlas.dsv.su.se` port 9494.

Hjälp

På denna sida finns alla hjälpresurser på kursen.

Handledning

Kör man fast så får man hjälp via news och e-post. Notera dock att det inte är någon handledning på:

- Kunskap som ingår i förkunskapskraven (kompileringsfel med mera)
- Gesällprovet

News är tänkt som en plattform för diskussion mellan studenterna men handledarna kommer att besvara de frågor som fortfarande är obesvarade senare under de dagar som det är handledning. E-post går direkt till handledarna och de kommer att försöka svara snabbt men eftersom de ibland får väldigt många frågor kan svaren dröja. Notera att handledarna enbart arbetar inom angivna handledningstider vilket medför att man kanske inte hinner få svar samma dag på en sent ställd fråga.

News

News bör användas för frågor som kan intressera andra:

- Läs först igenom de tidigare frågorna och svaren
- Hittar man inte en lösning så:
 - Ange ett väl beskrivande ämne för brevet
 - Var mycket tydlig vid beskrivningen av problemet
 - Sänd inte in mycket kod eller fullständiga program

News-server:

- Adress: `atlas.dsv.su.se`
- Port: 1199 (notera att detta inte är default numret och därför måste sättas manuellt)

News-grupper:

- public.ip1.a.1 för frågor om uppgift 1
- public.ip1.a.2 för frågor om uppgift 2
- ... och så vidare ...
- public.ip1.misc för övriga frågor

Se även [news](#) för mer detaljerad information om hur man konfigurerar en news-klient.

E-post

E-post bör användas för specifika frågor utan intresse för andra:

- Sänd inte e-post direkt till lärarna utan använd länken nedan
- Om man ska sända med kod:
 - Sänd endast med de delar som är relevanta för frågan
 - Koden ska vara välkommenterad och välstrukturerad
- Använd "svara till alla" på eventuella följdfrågor

E-post-server:

- Ingående e-brev (POP3):
 - Adress: mail.dsv.su.se
 - Port: 110
- Utgående e-brev (SMTP):
 - Adress: mail.dsv.su.se
 - Port: 25

Notera att om man sitter uppkopplad utanför DSV's domän (exempelvis via Telia Bredband) så ska man använda leverantörers SMTP-server för utgående brev.

E-post länk:

- [Klicka här för handledning via e-post](#)

Se även [e-post](#) för mer detaljerad information om hur man konfigurerar en e-post-klient.

FAQ

Här följer de frågor och svar som brukar vara vanligast:

Core Java

Fråga/svar 1

Hur når knapparna i min JPanel x textfälten i JPanel y?

Följande exempelprogram visar hur man kan referera till olika saker. Notera konstruktorn hos knappen som tar emot en referens från "framen" den ligger i och att denna frame har knapparna definierade som instansvariabler (inte riktigt snyggt, variablerna borde vara modifierade private och man ska nå dem via metoder ...):

```
import java.awt.event.*;
import javax.swing.*;
import java.util.Random;

public class JF extends JFrame {
```

```

JB jb1, jb2;

public static void main(String[] args) {
    new JF();
}

public JF() {
    jb1 = new JB(this);
    jb2 = new JB(this);
    getContentPane().add("North", jb1);
    getContentPane().add("South", jb2);
    setSize(100, 100);
    show();
}
}

class JB extends JButton implements ActionListener {
    Random rnd = new Random();
    JF jf;

    public JB(JF jf) {
        this.jf = jf;
        addActionListener(this);
    }

    public void actionPerformed(ActionEvent ea) {
        if(ea.getSource() == jf.jb1)
            jf.jb2.setText("Slump: " + rnd.nextInt(5));
        else
            jf.jb1.setText("Slump: " + rnd.nextInt(5));
    }
}

```

Om vi har exempelvis en JButton Draw som vid tryck skall rita om en JPanel så måste denna knapp ha en referens x till JPanel'en för att kunna nå dennas repaint-metod: x.repaint();.

Fråga/svar 2

Jag får felet: Can't make a static reference to nonstatic variable x in class MyFrame. Vad gör jag för fel och vad betyder egentligen static?

Metoden main är deklarerad static vilket betyder att den ska kunna anropas utan att man först skapat en instans av klassen. Detta betyder att om man i main vill använda sig av variabler deklarerade utanför main så måste även dessa vara deklarerade static, det vill säga tillhöra klassen, eftersom man ska kunna använda main utan att något objekt skapats. Eftersom x här är en instansvariabel, det vill säga tillhör ett objekt, så går följande inte att kompilera:

```

public class MyClass {
    int x = 10;

    public static void main(String[] args) {
        System.out.println("x: " + x);
    }
}

```

Följande går fint eftersom x här endast gäller internt i main-metoden:

```

public class MyClass {
    public static void main(String[] args) {
        int x = 10;
        System.out.println("x: " + x);
    }
}

```

Även följande går bra eftersom x själv är deklarerad static, det vill säga x tillhör klassen:

```

public class MyClass {
    static int x = 10;

    public static void main(String[] args) {
        System.out.println("x: " + x);
    }
}

```

```
}  
}
```

Fråga/svar 3

Hur tar man reda på vilken typ av objekt som är lagrat på ett visst ställe i en vector?

I klassen `Object` finns metoden `getClass` som returnerar ett objekt av typen `Class`. I klassen `Class` finns metoden `getName` som returnerar en `String`. Ett enkelt exempel:

```
import java.util.*;  
  
public class T7 {  
    public static void main(String[] args) {  
        Vector v = new Vector();  
        v.add(new Double(1.234));  
        v.add(new Integer(1));  
  
        for(int i = 0; i < v.size(); i++)  
            System.out.println(v.elementAt(i).getClass().getName());  
    }  
}
```

Kom ihåg att använda: `string1.equals(string2)` om man ska jämföra två strängar (inte: `string1 == string2`).

Fråga/svar 4

Hur får man, i en applikation, in en bild i en `JPanel`?

Betrakta följande exempel:

```
import java.awt.*;  
import javax.swing.*;  
  
class ImagePanel extends JPanel {  
    Image i = Toolkit.getDefaultToolkit().getImage("smiley.gif");  
  
    public ImagePanel() {  
        MediaTracker tracker = new MediaTracker(this);  
        tracker.addImage(i, 0);  
        try {  
            tracker.waitForID(0);  
        } catch (InterruptedException e) {}  
    }  
  
    public void paintComponent(Graphics g) {  
        g.drawImage(i, 0, 0, getSize().width, getSize().height, this);  
    }  
}  
  
public class I extends JFrame {  
    public I() {  
        setSize(100, 100);  
        getContentPane().add(new ImagePanel());  
        show();  
    }  
  
    public static void main(String[] args) {  
        new I();  
    }  
}
```

Observera!

- Ordningen på metदानropen i klassen `I`'s konstruktor är viktig.
- Bildfilen i detta exempel heter `smiley.gif` och som det ser ut ovan så måste den ligga i samma katalog som programmet gör.

Fråga/svar 5

Hur får man, i ett äpple, in en bild i en JPanel?

Betrakta följande exempel:

```
import java.awt.*;
import javax.swing.*;

class ImagePanel extends JPanel {
    Image i;

    public ImagePanel(Image i) {
        this.i = i;
    }

    public void paintComponent(Graphics g) {
        g.drawImage(i, 0, 0, getSize().width, getSize().height, this);
    }
}

public class I extends JApplet {
    public void init() {
        setSize (100, 100);

        Image i = getImage(getCodeBase(), "smiley.gif");
        MediaTracker tracker = new MediaTracker(this);
        tracker.addImage(i, 0);
        try {
            tracker.waitForID(0);
        } catch (InterruptedException e) {}

        getContentPane().add(new ImagePanel(i));
    }
}
```

Observera! Det är, som synes, ganska stor skillnad mellan hur man gör för applikationer och äpplen.

Fråga/svar 6

Jag får inte mitt äpple att reagera på tangetbordet, vad gör jag för fel?

Betrakta följande exempel:

```
import java.awt.event.*;
import javax.swing.*;

public class MyApplet extends JApplet {
    public void init() {
        setSize(100, 100);
        addKeyListener(new L1());
        requestFocus();
    }

    public void start() {
    }

    class L1 extends KeyAdapter {
        public void keyPressed(KeyEvent ke) {
            System.out.println("A key was pressed!");
            getToolkit().beep();
        }
    }
}
```

Det fungerar i både appletviewer och i Microsofts browser. Det är viktigt att man gör requestFocus på den komponent som har key-lyssnarna, i det här fallet äpplet självt.

Fråga/svar 7

Hur gör man JAR-filer?

För applikationer:

```
jar cmf manifest X.jar *.class
```

där:

- `c` betyder att man ska skapa (create)
- `m` betyder att man ska använda en manifest-fil (kallad manifest här). Innehållet i denna fil visar vilken klass som är startklass, exempelvis om `Y` är startklassen: `Main-Class: Y` Observera att denna fil inte ska skapas med emacs eftersom emacs lägger till skräptecken samt att det ska vara ny rad efter `Y` ovan.
- `f` betyder att output ska gå till fil (`x.jar` här) istället för till `STDOUT`.

För äpplen:

```
jar cf X.jar *.class
```

Man anger då startklassen med `code`-taggen:

```
<applet code="X.class" archive="X.jar" width="640" height="480"></applet>
```

När man har en JAR-fil så behöver man inte ha kvar CLASS-filerna.

Om något har gått snett:

- Ta bort alla manifest-, CLASS- och JAR-filer.
- Kompilera om allt.
- Fortsätt enligt ovan.

Fråga/svar 8

Mitt program använder bilder och när jag JAR'ar ihop allt så kommer inte bilderna med?

Man behöver en mer utförlig vägbeskrivning till bilden:

```
private URL url = ClassLoader.getResource("x.jpg");  
private Icon i = new ImageIcon(url);
```

E-post

På denna sida beskrivs hur man konfigurerar e-post-klienten. Följande är ett exempel på hur man konfigurerar e-post i Outlook Express, förfarandet är ungefär detsamma i andra moderna e-post/news-klienter:

- Klicka på Verktyg/Konton
- Klicka på Add/Mail
- Skriv in namn och e-post-adress
- Skriv in adressen för inkommande och utgående e-post-servrar
- Skriv in användarnamn och lösenord
- Klicka på Avsluta

Vill man använda exempelvis hotmail istället så kan man göra följande för att få all e-post som kommer till DSV vidareänd till hotmail-kontot:

- telnet triton.dsv.su.se

- emacs .forward
- Skriv in hotmail-adressen, spara och avsluta emacs
- Logga ut

De som använder ett e-postkonto på DSV kan även använda »[DSV: Webmail](#)«.

News

På denna sida beskrivs hur man konfigurerar en news-klient. Följande är ett exempel på hur man konfigurerar news i Outlook Express, förfarandet är ungefär detsamma i andra moderna e-post/news-klienter:

- Klicka på Verktyg/Konton
- Klicka på Lägg till/News
- Skriv in namn och e-post-adress
- Ange adressen för news-servern
- Klicka på Avsluta
- Klicka på mappen Alla och markera aktuell news-server
- Klicka på Egenskaper
- Klicka på Avancerat
- Ange port nummer till news-servern
- Klicka på Ok och sen Stäng
- Svara Ja på frågan Vill du ladda ner nyhets-grupper från news-kontot
- Dubbelklicka på alla grupper som är av intresse
- Klicka på ok

Default så visar Outlook Express bara de senaste meddelandena. Vill man ändra på det så:

- Klicka på Verktyg/Alternativ
- Klicka på mappen Underhåll
- Ändra vid Ta bort news meddelanden efter 5 dagar (exempelvis genom att ta bort markeringen helt)

Anledningen till att vi använder news istället för FirstClass är att news är en öppen mycket vanlig standard på Internet och att det även ingår som några förslag på gesällprov.

Redovisning i detalj

Denna beskrivning utgår från att man använder DSV's datorer men man kan använda andra ställen än DSV för redovisningen. Notera dock att andra ställen kan använda en annan huvudkatalog för webpublicering och andra rättigheter på kataloger och filer.

Steg för steg

1. Starta ett [filöverföringsprogram](#) och koppla dig till ditt DSV/Unix/Linux-konto
2. Skapa katalogstrukturen nedan och ladda upp filerna i denna struktur
3. Sätt rättigheterna på kataloger och filer så att allt blir läsbart från en browser
4. Testa att allt kan nås från en browser och att all källkod ser bra ut från en browser
5. Ange på redovisningssidan:
 - URL'en till denna katalogstruktur: `http://www.dsv.su.se/~x/ip1/`, där x är användarnamnet
 - Att allt är klart för redovisning

Katalogstruktur

Man måste ha **exakt följande katalogstruktur** men använder man en annan plats än DSV så måste inte www katalogen ha det namnet.

```
www
  ip1
    ip1.zip (ip1 zippad, ej innehållande ip1.zip givetvis)
    1.1
      källkods- och jar-filer för uppgift 1.1
    1.2
      källkods- och jar-filer för uppgift 1.2
    2.1
      källkods- och jar-filer för uppgift 2.1
    2.2
      källkods- och jar-filer för uppgift 2.2
    ... osv ...
  7
    källkods- och jar-filer för gesällprovet
    description
      beskrivning i text och bild av gesällprovet
    docs
      javadoc-beskrivning
```

Observera:

- Katalogen www måste ha stora/versala bokstäver
- Skapa inte kataloger för frivilliga uppgifter som inte är gjorda
- Lägg ingen fil med namnet index.html eller index.htm direkt i uppgiftskatalogerna eftersom man då inte kan lista innehållet i dessa kataloger från Internet

Beskrivning av gesällprovet

Gesällprovet ska beskrivas generellt med både text och bild (på programmet under körning) på något av följande sätt:

- HTML-dokument med bilderna inkluderade
- PDF-dokument med bilderna inkluderade
- TXT-dokument med bilderna separata

Ett HTML-dokument kan göras antingen för hand i en vanlig text-editor eller med ett högnivåverktyg.

För att skapa PDF-filer i Win32 kan man exempelvis:

1. Lägg till en skrivare, exempelvis HP Color LaserJet PS
2. Högerklicka på skrivaren och välj Egenskaper, klicka på Portar och kryssa i FILE
3. Skriv ut Word-dokumentet till denna skrivare varefter en fil <namn>.prn skapas
4. Gå till »PS2PDF.com« och konvertera filen till PDF

En bild (använd inte BMP-format) av ett program under körning fås enligt:

- Win32: Tryck på <CTRL><ALT><PRINT SCR> samtidigt, vilket medför att bilden på programmet läggs i urklippsbufferten som sedan lätt kan klistras in i ett HTML-dokument i en HTML-editor
- Linux/Unix: Skriv i kommandofönstret `import dump.jpeg` och klicka sen på det fönster som man vill ha en bild på (ändrar man till en annan extension så får man bilden som en annan bildtyp)

All eventuell java-kod i gesällprovet ska beskrivas med verktyget javadoc.

Efter erhållet betyg

När man fått G/VG eller 3/4/5 på ett arbete så bör man ändra rättigheterna på källkodsfilerna så att ingen annan kan läsa dem.

Nyheter

På denna sida visas alla uppdateringar.

Tid	Rum	Uppdatering
2004-08-22 16:40	Allt	Nya systemet klart
2004-08-25 03:10	Utskrift	Implementerad
2004-09-01 00:10	Redovisning i detalj	Ny katalogstruktur
2004-09-07 23:30	Schema	Vissa nya tider för deadline och tenta
2004-10-10 01:10	Uppgifter	Uppgift 3.3, Ping med J2SE 5.0
2005-03-10 01:35	Tävlingen	Vinnarna online

Internetprogrammering II

På denna sida finns hela kursen utom utom OH-bilderna och exemplen.

Intro

Välkommen Pierre! På denna sida beskrivs kursen generellt.

Kurskoder

Kursen är på 5 poäng och går för följande universitet/högskolor:

- SU med kod *:9
- KTH med kod 2I1802
- KTH-FoV med kod 2I4113
- Nätuniversitetet med kod IB9030

Kursen ges även i en utökad fördjupande form, kallad IP2+, på 5 + 2 poäng för följande universitet/högskolor:

- SU med kod *:9+

Notera att man **inte senare kan komplettera** från IP2 till IP2+ utan att man måste välja IP2+ direkt från start.

Innehåll

Denna kurs lär ut Internetprogrammering fördjupat mot HTTP-server-sidan och man kan välja teknik och språk fritt:

- Servlets med Java
- CGI-programmering med exempelvis Perl
- ASP.NET med exempelvis C#
- JSP med Java
- PHP

- Med flera

Notera dock följande:

- Det finns flera **krav på kursen** avseende **kodseparation**, se sidan Uppgifter -> Krav
- **Handledning och stöd** ges endast inom **Servlets** och **CGI med Perl**

Oberoende av teknik och språk så tas följande upp:

- Understödjande tekniker:
 - HTTP-servrar (Apache, Tomcat)
 - HTTP-protokollet (GET, POST med mera)
- Output:
 - Text:
 - Säker filhantering
 - Omgivningsvariabler
 - Bild med dynamisk grafikgenerering
 - Omladdningar:
 - Klientstyrd
 - Serverstyrd: Tidsstyrd
 - Serverstyrd: Tidsstyrd: Olika mime-typer
 - Serverstyrd: Händelsestyrd
- Input:
 - URL och länkar
 - GET
 - Formulär:
 - GET
 - POST
 - Uppladdning av fil
- Kodseparation:
 - Statisk mängd data
 - Variabel mängd data
- Sessionshantering:
 - Omskrivning:
 - Länkar
 - Formulär
 - Lagra på klienten:
 - Kakor
 - Säkra kakor
- E-post:
 - Sändning:
 - Utan bifogade filer
 - Med bifogade filer
 - Mottagning:
 - Utan bifogade filer
 - Med bifogade filer
- Databaser:
 - Lättvikts med Database Management (DBM)
 - Relations/SQL:
 - I/O (three-tier, SQL-injection, MySQL)
 - Transaktioner
 - Effektivitet (connection pooling)
- Innehållshantering:
 - Manuellt innehåll:
 - Publiceringssystem (Content Management System, CMS), exempelvis WIKI
 - Syndikering
 - Automatiskt innehåll:
 - Kanaler
 - Fritextsökning (sökmotor)

Teknik och språk: Fritt vald teknik och språk, SQL.

Undervisningsform

Pedagogiken för denna kurs är:

- (1) Frihet, (2) tävling och (3) samarbete --> glädje och motivation --> ett bra resultat!

Frihet, tävling och samarbete gör att man går från passivt mottagande till aktivt sökande. Genom frihet tillåts man själv bli nyfiken och ställa frågor innan man får svar. Genom tävling motiveras man att göra sitt bästa. Detta leder fram till glädje och motivation och därmed till ett bra resultat.

Observera! Detta är en distanskurs och kräver förmåga till självständigt arbete. Kursen har dock, till skillnad mot vanliga distanskurser, kontinuerligt och direkt stöd via handledning på distans. Kursen är krävande och man bör inte läsa andra kurser samtidigt. Kursen fokuserar starkt på praktiskt arbete och examinationen bygger på detta arbete. Digitalslöjd!

Examination

Examinationen på kursen består av:

- Praktiska uppgifter, 4 poäng (R/G/VG) (R/3/4/5):
 - Flera bestämda/styrda
 - Ett fritt valt gesällprov
- Tentamen, 1 poäng (U/G) (U/3)

För fördjupningskursen gäller utöver ovanstående även:

- Extra praktiska uppgifter, 2 poäng (R/G) (R/3):
 - Flera bestämda/styrda

Totalbetyget på båda kurserna bestäms av de praktiska uppgifterna med tyngdpunkt på det fritt valda gesällprovet.

Notera att man måste lämna in lösningar på de praktiska uppgifterna som stämmer med **nuvarande läsårs praktiska uppgifter**. Kurserna utvecklas dock bara i små steg med gradvisa förfiningar så det innebär inte att man måste göra om allt om man inte har fått betyg på de praktiska uppgifterna vid den sista deadline i Augusti.

Litteratur

Litteraturen är:

- För de som väljer Servlets:
 - »[Java Servlet Programming \(andra utgåvan\)](#)« skriven av Hunter och utgiven av »[O'Reilly](#)«
- För de som väljer CGI tipsar vi om följande böcker men de är ej förbeställda på Akademibokhandeln i Kista:
 - »[Programming Perl \(tredje utgåvan\)](#)« skriven av Wall, Christiansen, Orwant och utgiven av »[O'Reilly](#)«
 - »[Official Guide to Programming with CGI.pm](#)« skriven av Stein och utgiven av »[Wiley](#)«

Billigaste boken hittas med hjälp av »[Bokfynd.nu](#)«. Vi rekommenderar »[Amazon](#)« och »[WHSmith](#)« i England som brukar ha bra priser och snabba leveranser. Köp inte böcker utanför EU eftersom det tar lång tid (2-3 veckor) att få böckerna och sen får man en ganska stor extra avgift från tullen.

4 steg för att klara denna kurs

Gör följande för att klara denna kurs:

1. Registrera dig hos »[DSV: Studievägledningen](#)« som har e-post svl@dsv.su.se
2. Köp och läs kurslitteraturen
3. Lös och redovisa uppgifterna
4. Anmäl dig till tentamen (minst en vecka innan den går) och klara denna

Lärare

Vi önskar alla en givande och rolig kurs!

- »[Pierre A. I. Wijkman](#)«, ansvarig
- Handledarna

Lycka till!

Schema

På denna sida finns alla tider och platser för kursen. Alla tider är preliminära och kan ändras.

Föreläsningar

Detta är en distanskurs och vi ger inte en hel föreläsningsserie. Den inledande föreläsningen fungerar enbart som en kick-off eller frösättning. Notera att innehåll markerat som reserv innebär att ingen föreläsning ges men att detta kan ändras vid behov.

Tid	Rum	Föreläsare	Innehåll
2004-10-25 kl 13-17	Sal A, Forum, Kista	Pierre A. I. Wijkman	Intro, HTTP, HTML, CGI, Servlets
2004-10-26 kl 14-18	Sal A, Forum, Kista	-	Reserv

Handledning

All handledning sker via e-post och news (se sidan Hjälp).

Samtliga studenter har handledning:

- Kursens 1:a - 5:e vecka: samtliga vardagar kl 10-17

Studenter som läser +-kursen (7 poäng) har även handledning på +-delen:

- Kursens 6:e - 7:e vecka: samtliga vardagar kl 10-17

Studenter från Nätuniversitetet som läser på halvfart har även handledning:

- Kursens 6:a - 10:e vecka: samtliga vardagar kl 10-17

För de som arbetar på DSV finns info om prio-tider för labbsalarna under »[DSV: Scheman](#)«.

Examination

Notera följande angående tid för rättning:

- Det brukar ta upp till **3 veckor tills rättningen är klar**
- Ansvarig kan **inte rätta vissa personers uppgifter före andras**, exempelvis de som har bråttom på grund av CSN
- När man fått betyg på kursen så dröjer det **ytterligare tid** tills allt är infört i Daisy och Ladok

Var god **e-posta inte ansvarig om detta** för att försöka påskynda processen, det är väldigt många studenter på samtliga kurser.

Man får välja tid för tenta och deadline oberoende av om man går på SU eller KTH och så vidare. Typen nedan är enbart en rekommendation.

Deadlines för uppgifterna

Det finns flera deadlines för inlämning av uppgifterna och man väljer själv en som passar. Man redovisar samtliga uppgifter direkt eftersom det av praktiska skäl tyvärr inte går att delrätta och ge återkoppling efter hand som deluppgifterna blir klara. De som går en + kurs kan dock lämna in ordinarie uppgifter och + uppgifter separat vid olika deadlines. Varje deadline är normalt daterad till dagen innan tentamen.

Tid	Rum	Typ
2004-11-26 kl 24	Internet	Deadline 1 SU kurs IP1
2004-12-17 kl 24	Internet	Deadline 1 KTH och SU kurs IP2+
2005-03-29 kl 24	Internet	Deadline 2
2005-08-21 kl 24	Internet	Sommardeadline

Observera:

- Redovisade uppgifter **rättas endast i samband med dessa deadlines och uppskov ges aldrig**
- Får man **rest** så **rättas den igen först vid nästa deadline**
- Betyget påverkas inte beroende på vilken deadline man väljer
- Man måste inte välja en deadline som ligger i anslutningen till en tenta
- Handledningen slutar ofta innan deadline 1 (det finns tyvärr inga resurser för en längre handledningsperiod)

Tentamen

Inga hjälpmedel är tillåtna på tentamen. Man bör ha gjort klart de obligatoriska uppgifterna för 5 poängskursen (exklusive gesällprovet) innan tentamen eftersom tentamen kommer att fråga om dessa (detta är dock inget strikt krav för att få gå upp på tentamen). Vissa frågor kan vara mer generella och vissa kan vara mer tekniska och fråga om hur viss kodning (med avseende på uppgifterna) görs.

Tid	Rum	Typ
2004-11-27 kl 10-14	Forum, Kista	Tentamen SU
2004-12-18 kl 10-14	Forum, Kista	Tentamen KTH
2005-03-30 kl 15-19	Forum, Kista	Omtentamen
2005-08-22 kl 14-18	Forum, Kista	Sommartentamen

Observera:

- Man måste anmäla sig i Daisy **senast en vecka innan tentamen går**, se »[DSV: Att tentera vid DSV](#)«
- Man måste inte välja en tenta som ligger i anslutningen till en deadline
- Sal meddelas i Daisy eller på telefonsvarare 08 674 70 04 senast en dag innan tentamen, se även »[DSV: Scheman](#)«

Var god **e-posta inte ansvarige** (han har inga befogenheter här) **om detta**, utan kontakta istället »[DSV: Studentexpeditionen](#)«.

Kursmaterial

På denna sida finns information om arbetssätt och vilka mjukvaror man behöver. Här finns även OH-bilder och tidigare tentor.

Arbetssätt och mjukvaror

På denna kurs kan man arbeta på följande sätt:

- På DSV:
 - Servlets: [Tomcat](#)
 - CGI/Perl: [Apache](#)
- På distans online kopplad via telnet eller SSH till sitt DSV/Unix-konto:
 - [Terminalprogram](#)
 - Servlets: [Tomcat](#)
 - CGI/Perl: [Apache](#)
- På distans offline med det egna systemet konfigurerat för allt som behövs:
 - [Editor](#)
 - Servlets:
 - [Java](#)
 - [Tomcat](#)
 - CGI/Perl:
 - [Perl](#)
 - [Apache](#)
 - Sendmail: Ingår i Unix/Linux, för Win32 använd »[NTsendmail](#)« eller »[IndigoMail](#)«
 - [MySQL](#)
 - [Filöverföringsprogram](#)

Det senare är det klart roligaste och mest lärorika sättet eftersom man då måste installera allt som behövs på den egna datorn. Notera att ingen handledning ges på detta eftersom det finns alltför många olika varianter av konfigurationer.

Man kan även använda en [proxyserver](#) om man vill se i detalj vad som sänds mellan en HTTP-klient och en HTTP-server.

OH-bilder

OH-bilderna finns enbart här:

- [00_intro.pdf](#)
- [01_content.pdf](#)
- [02_http_server_side.pdf](#)
- [03_html_server_side.pdf](#)
- [04_cgi_perl.pdf](#)
- [05_servlets.pdf](#)

Tidigare tentor

Tidigare tentor finns enbart här:

- [Tidigare tentor](#)

Notera att de är baserade på de uppgifter som var obligatoriska vid tidpunkten för tentan och som möjligtvis inte är det nu.

Terminalprogram

På denna sida beskrivs hur man använder terminalprogram.

Terminalserver

Om man behöver logga in och köra kommandon på en avlägsen dator så använder man ett terminalprogram/klient. På DSV används följande dator som terminalserver:

- `triton.dsv.su.se`

Denna dator kör Linux och har både telnet- och SSH-server igång.

Om man inte släpps in kan det bero på att DSV's brandvägg inte släpper in just det IP-nummer som man använder. Då kan man kontakta [Christer Jansson](#) som kan registrera IP-nummret i brandväggen så att man kommer förbi den.

Telnet

I Win32 och Linux så kan man köra ett Terminal-program direkt i ett kommando-fönster:

- `telnet triton.dsv.su.se`

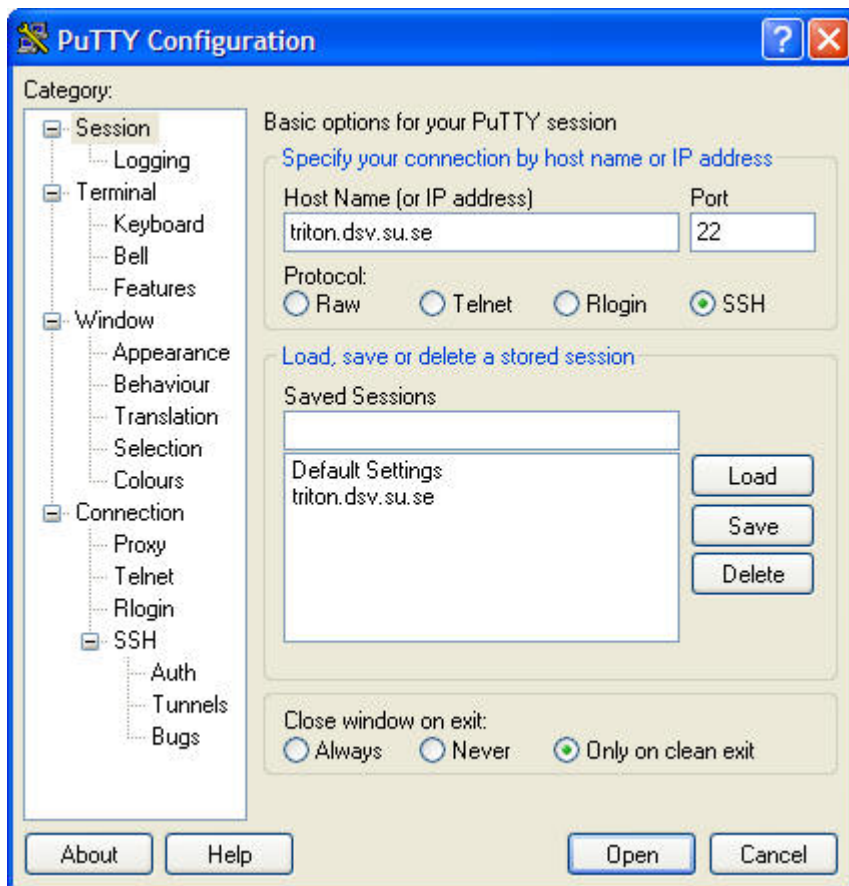
Man anger sedan användarnamn och lösenord.

SSH

När man använder telnet så sänds all information okrypterad över Internet och det kan vara farligt. Vill man undvika det kan man istället använda SSH och följande program rekommenderas:

- Win32: »[Putty](#)«
- Linux: `ssh`

Så här ser det ut när man startar Putty:



Editor

På denna sida beskrivs det verktyg man använder för att skapa kod med.

Intro

Man kan använda allt från en enkel editor till en avancerad grafisk utvecklingsmiljö. Valet är fritt! En avancerad grafisk utvecklingsmiljö kan vara något svårare att komma igång med men är överlägsen när man väl lärt sig den. En grafisk utvecklingsmiljö kan generera vissa delar av koden i ett program automatiskt, exempelvis den kod som tar han om det grafiska användargränssnittet, vilket snabbar upp konstruktionsarbetet betydligt. En stor nackdel med detta är att koden tenderar att bli omfattande och främmande för en nybörjare och det blir mycket svårt att anknyta sin egen kod till den kod som verktyget har skapat. Det bästa är att börja med en enkel editor och sedan gå över till en grafisk utvecklingsmiljö.

Observera att koden ska vara läsbar direkt i en vanlig browser för enklare rättning. Man bör därför inte använda en editor som använder kontrolltecken och tabbar för att indentera koden.

Enkla

- Notepad (Win 32)
- »[Pico](#)« (Unix)
- »[Emacs](#)« (Win32 och Unix)

Använder man Notepad så måste man spara filerna genom att skiva citationstecken runt filnamnet "X.java" när man namnger och sparar filen, annars sparas filen som X.java.txt.

Använder man emacs för Java bör man inte använda JDE-tillägget eftersom det lätt blir fel med classpath med mera. Kompilera, kör, sätt classpath med mera i ett kommandofönster istället:

- Skriv koden i en editor och spara filen (här x.java)
- Starta ett kommandofönster och gå till katalogen där koden ligger sparad
- Kompilera med `javac x.java`
- Kör med `java x`

Tips Win32:

- Har man Windows95/98/ME så bör man börja med att skriva doskey i kommandofönstret så att man kan använda "uppåtpil" och "nedåtpil" istället för att skriva om allt man skrivit tidigare
- Man kan lägga en genväg till DOS-promten i varje katalog och konfigurera (höger-klicka och välj Egenskaper) så att den startar i samma katalog:
 - Windows 98: Välj Program/Arbetsmapp och skriv en .
 - Windows 2000/XP: Välj Shortcut/Start in och gör detta text-fält tomt
- När det är långa krångliga kommandon så kan man i DOS använda BAT-filer genom att skriva kommandot i en fil med namnet run.bat så kan man sedan köra kommandot med run

Vill man strukturera upp sin XHTML-kod så rekommenderas:

- »[W3C: Tidy](#)«
- »[SourceForge: Tidy](#)«
- »[Charlie's Tidy Add-Ons](#)«

Mer komplexa

- »[SciTe](#)«, se även »[SciTe Bundled](#)«-versionen
- »[JEdit](#)« (för Java-kod)
- »[HTML-kit](#)« (för XML- och XHTML-kod)

Konfiguration av SciTE:

- Ladda den fil som bestämmer SciTE's funktion: Options / Load Global Options File
- Sen kan man sätta lite olika saker, exempelvis:
 - Använd 3 mellanslag vid intendering: `indent.size=3`
 - Använd inte tabbar vid intendering: `use.tabs=0`
- Spara sen filen

Sessionshantering med SciTE:

- Om man har många filer öppna och vill fortsätta med dessa senare så sparar man sessionen med: File / Save Session ...
- När man vill öppna alla filer igen sen så går man till: File / Load Session ...

Kompleta utvecklingsmiljöer

- »[Eclipse](#)« (för Java-kod)

Java

På denna sida beskrivs installation av Java och Java API specifikationen samt konfiguration av Java.

Installation av Java

Java finns i tre huvudversioner:

- Java Micro Edition (J2ME) som är gjord för mobila/trådlösa enheter
- Java Standard Edition (J2SE) som har fokus på klientsidan
- Java Enterprise Edition (J2EE) som har fokus på serversidan

Vi ska i fortsättningen titta på J2SE som finns i sex huvudversioner: 1.0.x, 1.1.x, 1.2.x, 1.3.x, 1.4.x och 1.5.x (kallad 5.x). Det är stor skillnad mellan dessa versioner och på denna kurs används den senaste versionen. Den finns för nerladdning för olika operativsystem:

Version 1.4.2:

- DOS/Windows95/98/ME: »[J2SE 1.4.2](#)«
- Windows NT/2000/XP: »[J2SE 1.4.2](#)«
- Unix/Linux: »[J2SE 1.4.2](#)«
- MacOS: »[J2SE 1.4.2](#)«

Version 1.5.0:

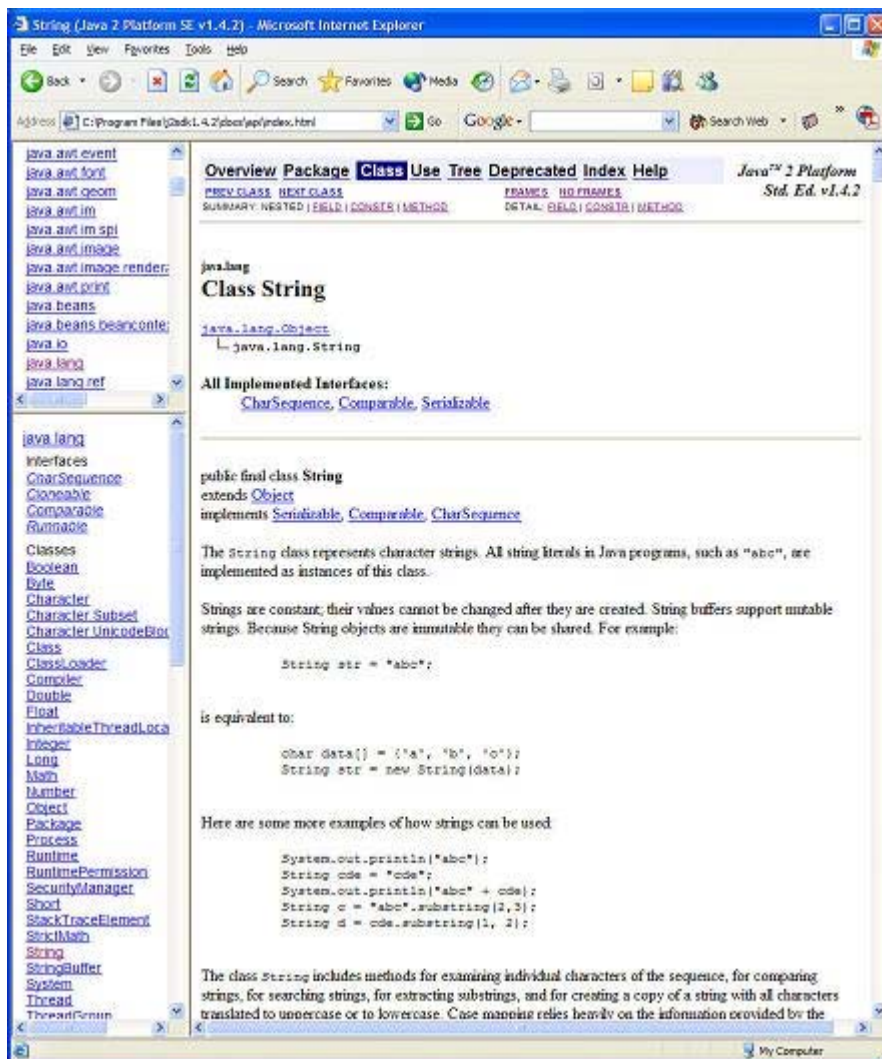
- DOS/Windows95/98/ME: »[J2SE 1.5.0](#)«
- Windows NT/2000/XP: »[J2SE 1.5.0](#)«
- Unix/Linux: »[J2SE 1.5.0](#)«
- MacOS: <finns ej än>

I texten nedan används version 1.4.2 i sökvägar med mera och man får byta denna text så att den passar den version som man själv använder. Efter nerladdning installerar man Java genom att starta installationsprogrammet.

Installation av Java API specifikationen

Java API specifikationen är en datorbaserad uppslagsbok som behövs eftersom Javas bibliotek av klasser är enormt stort och växer snabbt hela tiden för varje ny version av Java. Om man ska slå upp specifika saker som klass-namn, variabel-namn, metod-namn med mera så är Java API specifikationen till mycket stor hjälp. Notera att man bara får dåligt självförtroende om man börjar med Java API specifikationen utan att ha en grundläggande förståelse och utan att man har testat några programexempel tidigare.

Java API specifikationen kan läsas online men även laddas ner från samma ställe som Java hämtas från om man vill läsa den offline. Uppslagsboken körs i en webb-läsare och ser ut så här:



Notera att det finns en stor mängd tillbehör till Java som inte ingår i någon av huvudversionerna vars API-beskrivningar måste hämtas ner separat.

Konfiguration av Java

Enkel körning

För att kunna nå alla program som ingår i J2SE på ett enkelt sätt så sätter man omgivningsvariabeln path till den katalog där programmen ligger. Då kan man exempelvis köra java- och javac-programmen från alla kataloger utan att behöva skriva hela sökvägen till dessa program eftersom systemet nu hittar dessa program automatiskt.

DOS/Windows95/98/ME:

- Antag att vi har installerat java till c:\Program Files\j2sdk1.4.2_04
- Öppna filen c:\autoexec.bat från ett DOS-fönster genom att skriva edit autoexec.bat
- Skriv följande sist i denna fil:
 - Nya versioner av operativen: set path=%path%;C:\Program Files\j2sdk1.4.2\bin
 - Gamla versioner av operativen: set path=%path%;C:\Progra-1\j2sdk1.4.2\bin
- Spara, avsluta och starta om datorn

Windows NT/2000/XP:

- Antag att vi har installerat java till c:\Program Files\j2sdk1.4.2_04

- Klicka på Start/Settings/Control Panel/System/Advanced/Environment Variables...
- Under System variables, välj Variable: Path och klicka på Edit
- Lägg till följande sist i Variable Value: ;C:\Program Files\j2sdk1.4.2_04\bin
- Spara, avsluta och starta om datorn

Unix/Linux:

- Eftersom du är Unix/Linux-användare så ser du detta som en utmaning och vill inte fuska och ha någon instruktionen för detta :-).

MacOS:

- Eftersom du är MacOS-användare så fungerar allt automatiskt och du behöver inte någon instruktionen för detta :-).

Tillbehör

Det finns ett stort antal tillbehör till de tre huvudversionerna (J2ME, J2SE, J2EE). Dessa tillbehör kallas för standard extensioner eller extra paket och är utbyggnader som finns inom många olika områden, exempelvis säkerhet och media. För att använda ett tillbehör så går man till SUN's eller någon annan leverantörs sida på Internet och hämtar en eller flera filer.

Det finns två huvudtyper av tillbehör:

- Plattformsberoende gjorda enbart med Java
- Plattformsberoende gjorda med Java och native kod

Plattformsberoende tillbehör används för saker där effektiviteten är viktig eller om tillbehöret måste nå plattformsspecifik hårdvara som i exempelvis vid användande av 3D-grafik och/eller andra media-tekniker

Det finns tre huvudsätt att köra Java-program:

- Fristående program
- Äpplen
- Web Start

Varje sätt använder tillbehör på sitt eget speciella sätt och sammanfattningsvis kan man använda tillbehör på följande sätt:

- Fristående program:
 - Plattformsberoende tillbehör:
 - Tillbehöret läggs i speciella kataloger
 - Tillbehöret ligger på ett godtyckligt ställe och man sätter classpath dit
 - Tillbehöret ligger på samma ställe som huvudprogrammet
 - Plattformsberoende tillbehör:
 - Tillbehöret läggs i speciella kataloger
 - Tillbehöret ligger på samma ställe som huvudprogrammet
- Äpplen:
 - Plattformsberoende tillbehör:
 - Tillbehöret läggs i speciella kataloger
 - Tillbehöret ligger på samma ställe som huvudprogrammet
 - Plattformsberoende tillbehör:
 - Tillbehöret läggs i speciella kataloger
- Web Start:
 - Plattformsberoende tillbehör:
 - Tillbehöret läggs i speciella kataloger
 - Tillbehöret ligger på samma ställe som huvudprogrammet
 - Plattformsberoende tillbehör:

- Tillbehöret läggs i speciella kataloger

Vi ska inte gå in på alla detaljer om detta men titta lite generellt på de tre olika huvudsätten att använda tillbehör, se »[Java extensioner med nativelib-filer](#)« för mer detaljerad information.

Tillbehöret läggs i speciella kataloger

JAR- och eventuella DLL- (eller motsvarande) filer placeras enligt:

- JAR-filer läggs här:
 - För att kunna köra program (JRE): <sökväg>\Java\j2re1.4.2_04\lib\ext
 - För att kunna utveckla program (kompilera mm) (JDK): <sökväg>\j2sdk1.4.2_04\lib
- DLL-filer (eller motsvarande) läggs här:
 - För att kunna köra program (JRE): <sökväg>\Java\j2re1.4.2_04\bin

Notera att sökvägarna ovan kan vara olika på olika maskiner.

När JAR- och DLL- (eller motsvarande) filer ligger på dessa ställen behöver man inte sätta classpath till dessa filer för att kunna köra och utveckla program som är beroende av tillbehöret.

Tillbehöret ligger på ett godtyckligt ställe och man sätter classpath dit

Om filerna ligger på andra ställen än beskrivet ovan så kan man sätta omgivningsvariabeln classpath till antingen:

- en katalogstruktur med CLASS-filer, eller
- en eller flera JAR-filer

Genom att sätta classpath så berättar man för programmen javac, java med flera var detta tillbehör finns. Notera att man även kan sätta classpath direkt när man använder javac eller java.

DOS/Windows95/98/ME:

- I filen C:\autoexec.bat skriver man på sista raden set classpath=%classpath%;c\x\y

Windows NT/2000/XP:

- Gå till Kontrollpanelen/System/Avancerat/Omgivningsvariabler och sätt eller uppdatera classpath.

Unix/Linux:

- Man sätter classpath olika beroende på vilket skal man använder:
 - C-skäl: setenv classpath <sökväg>
 - Bourne-skäl: classpath=<sökväg> och sen: export classpath
 - Korn-skäl: export classpath=<sökväg>
 - Bash-skäl: export classpath=<sökväg>

Detta kan man lägga i den fil som kör automatiskt (som har olika namn i olika skal) när man loggar in.

MacOS:

- Se Unix/Linux

Tillbehöret ligger på samma ställe som huvudprogrammet

Man kan slutligen lägga tillbehöret på samma ställe som huvudprogrammet. Är tillbehöret packat i en JAR-fil så packar man upp det i denna katalog. Mer tydligt, antag att man har sina Java-program här:

- C:\x\y

Då ska man lägga tillbehörskatalogen z här:

- C:\x\y\z

En nackdel med detta sätt är att man måste ha flera kopior av tillbehöret om man ska använda det på flera ställen.

Filöverföringsprogram

På denna sida beskrivs hur man använder filöverföringsprogram.

Intro

För att kunna sända filer mellan olika datorer använder man filöverföringsprogram och vanliga standarder för det är FTP, SFTP och FTPS.

Ett vanligt fall är när man har HTML-filer som man vill sända till DSV för att synliggöra dem på Internet. Man använder då sitt DSV/Unix/Linux-konto och lägger allt under `www`-katalogen. När filerna är överförda måste man även sätta rättigheterna på filerna och katalogerna:

- Kataloger: `chmod 755`
- De flesta filtyperna: `chmod 744 *`
- CGI-filer: `chmod 711 *.cgi`

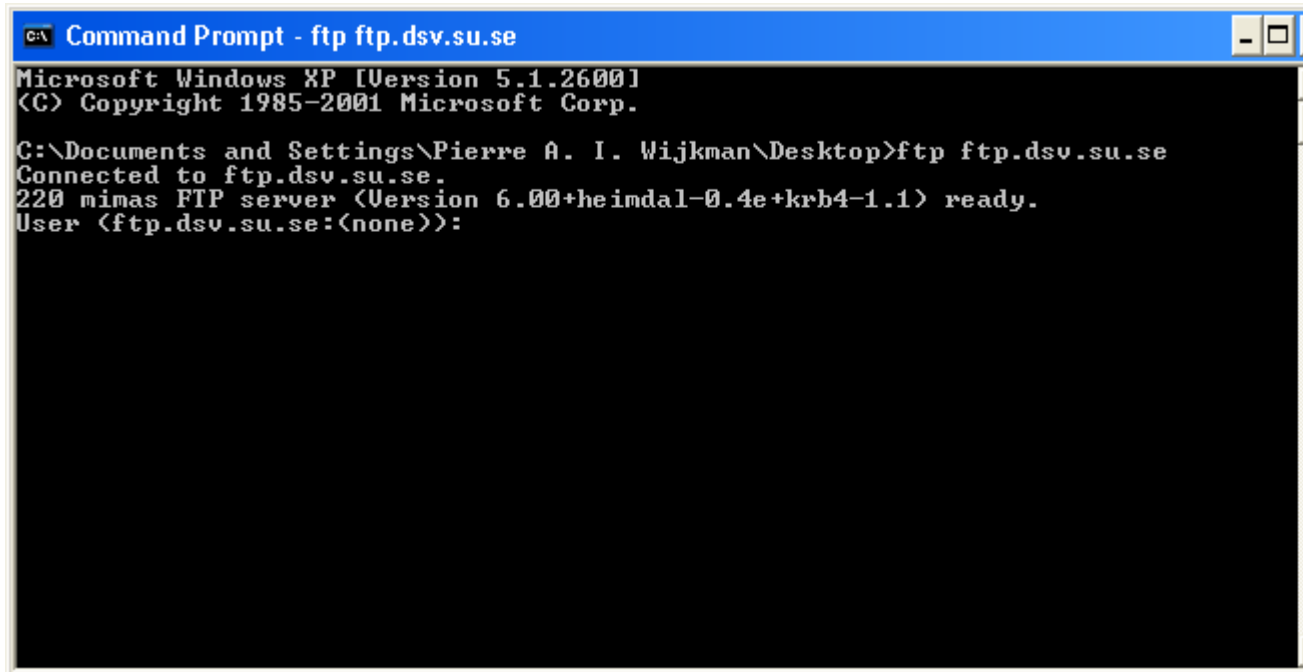
Notera att `www`-katalogen inte ska vara med i URL'ens sökväg när man sedan ska hämta filer i en browser:

- En fil `z.html` på DSV/Unix/Linux-kontot under katalogen `www/x/y` når man i en browser med `http://dsv.su.se/~<användarnamn>/x/y/z.html`

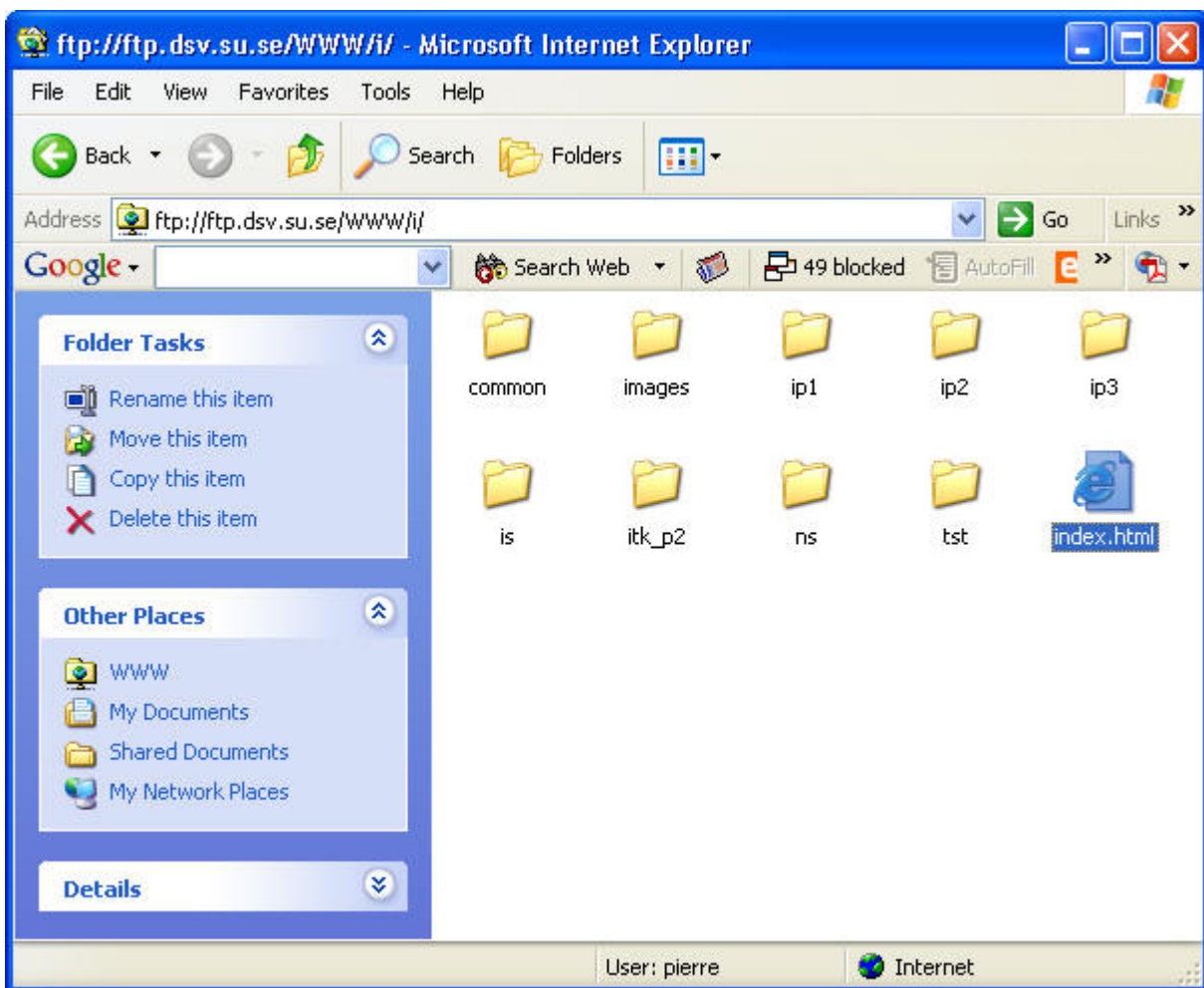
Man kan även nå sina filer på DSV/Win32 men det är endast DSV/Unix/Linux som har en tillförlitlig HTTP-server.

FTP

I Win32 och Linux så kan man köra en FTP-klient direkt i ett kommando-fönster:



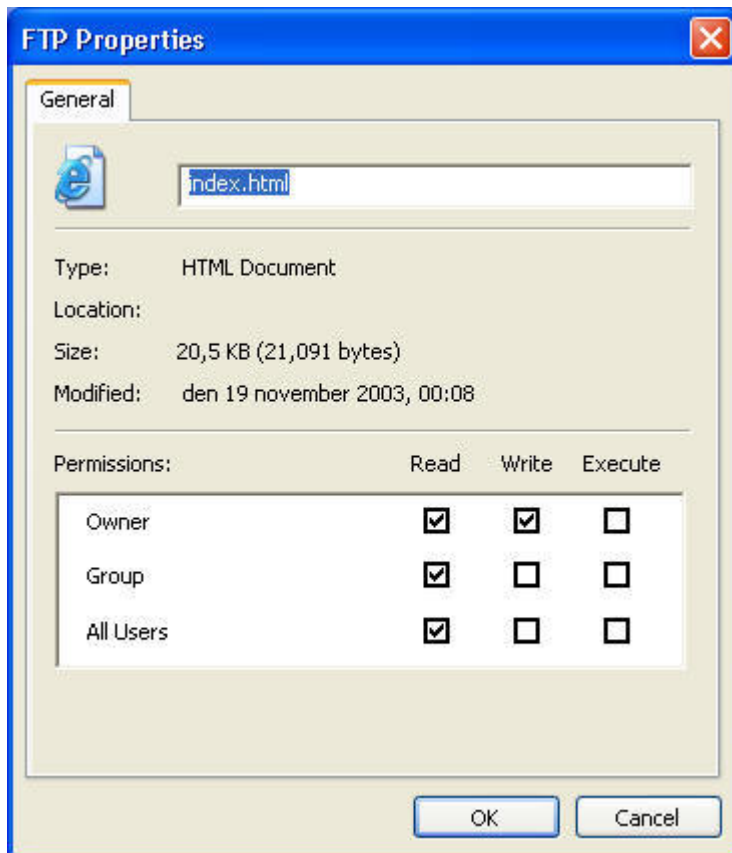
Det är dock ofta lättare att använda en GUI-baserad FTP-klient. Det finns många olika program och »Shareware.com« är ett ställe som man kan leta på. Man även använda en vanlig browser, exempelvis Internet Explorer:



Man använder sen drag-and-drop. Man kan även sätta fil- och katalogrättigheter från detta program:

- Högerklicka på fil eller katalog
- Välj Egenskaper
- Sätt rättigheter

Så här kan det se ut:



Notera dock att Internet Explorer inte stöder ASCII-mode.

Man ansluter sig till FTP-servern på följande dator:

- DSV/Unix/Linux: `ftp://ftp.dsv.su.se`

Använder man Internet Explorer så kan man skapa en genväg enligt:

- `ftp://<användarnamn><lösenord>@ftp.dsv.su.se`

Notera att man bör sända:

- Rena text-filer (som CGI-program med mera) i ASCII-mode
- Binära filer (som bilder, kompillerade program med mera) i BINARY-mode

Använder man en brandvägg så bör man se till att initieringen av dataöverföringskanalen sker på klient-sidan och det gör man genom att använda `passive mode` som kan ställas in i de flesta FTP-klienter.

SFTP och FTPS

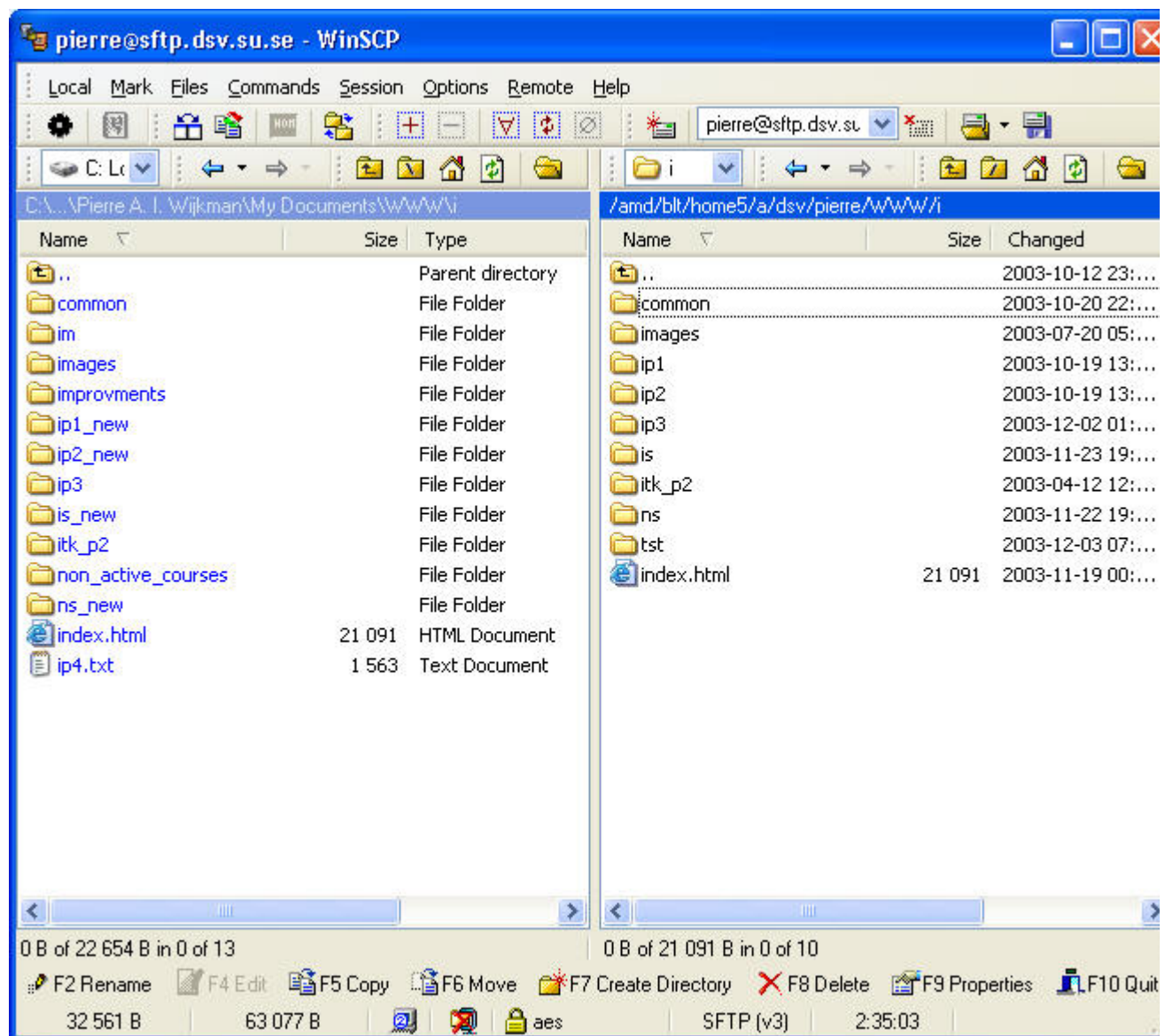
När man använder FTP så sänds all information, även lösenordet, okrypterat över Internet och det kan vara farligt. Man bör istället använda någon av följande tekniker:

- SFTP som använder sig av tunnling via SSH och först upprättar en SSH-anslutning som sedan används för att ansluta till en server via FTP
- FTPS som ansluter som vanlig FTP enligt RFC959 för att sedan begära en säker anslutning via SSL/TLS enligt RFC2228

DSV stöder enbart SFTP så man måste använda ett sådant program och bra program för det är:

- Unix/Linux: sftp som ingår i OpenSSH
- Win32: »[WinSCP](#)«

Exempelvis så ser WinSCP ut så här:



Man ansluter till SFTP-servern på följande datorer:

- DSV/Unix/Linux: sftp.dsv.su.se
- DSV/Win32: wftp.dsv.su.se

Detta säkrare sätt rekommenderas starkt.

Proxy-server

Vill man se exakt vad som sänds mellan en browser och en HTTP-server så kan man använda en proxy-server:

- Källkod: [Proxy.java](#)
- Hämta filen: [Proxy.jar](#)
- Kör med: `java -jar Proxy.jar`

Man måste även ändra inställningarna i browsern, i exempelvis Internet Explorer gör man enligt följande:

- Verktyg / Internet möjligheter / Anslutningar / LAN inställningar
- Under Proxy-server: kryssa i rutan och klicka på Avancerat
- Under HTTP:
 - Adress: localhost
 - Port: 9999

Nu sker alla förfrågningar till proxy-servern som i sin tur kopplar vidare till HTTP-servern. Proxy-servern ligger mellan dessa båda och visar all kommunikation. Notera att proxy-servern är konstruerad på 3 timmar och alltså är i version 0.01 :-):

Tomcat

På denna sida finns hela Tomcat beskrivningen.

Intro

På denna sida introduceras HTTP-servern Tomcat som används för att köra Servlets. Det finns två sätt att göra detta:

- Använda DSV's centrala Tomcat
- Använda en egen Tomcat

DSV's centrala Tomcat ska enbart användas för att köra ett fåtal färdigutvecklade och stabila Servlets. För att utveckla servlets ska man använda en egen Tomcat lokalt, antingen på en dator på DSV eller hemma.

@DSV

På denna sida beskrivs hur man använder DSV's HTTP-server för Servlets.

Adress

På DSV körs en Tomcat här:

- Host: `atlas.dsv.su.se`
- Port:
 - HTTP: 8080
 - HTTPS: 8443

DSV's Tomcat stöder:

- Java Servlets 2.3
- Java Server Pages 1.2

Till detta finns även stöd för en rad andra tekniker:

- I katalogen jakarta-tomcat-4.1.24/shared/lib: cos.jar, mixer.jar, mm.mysql-2.0.14-bin.jar
- I katalogen jakarta-tomcat-4.1.24/common/endorsed: xercesImpl.jar, xmlParserAPIs.jar
- I katalogen jakarta-tomcat-4.1.24/common/lib: activation.jar, ant.jar, commons-collections.jar, commons-dbcp.jar, commons-logging-api.jar, commons-pool.jar, jasper-compiler.jar, jasper-runtime.jar, jdbc2_0-stdext.jar, jndi.jar, jta.jar, mail.jar, naming-common.jar, naming-factory.jar, naming-resources.jar, servlet.jar

Notera att man **inte ska ta med någon av dessa filer** i lib-katalogen på den egna webapplikationen.

Krav för användning

Tomcat är mycket mer känslig än Apache med tekniken CGI/Perl eftersom alla Servlets kontinuerligt ligger laddade i minnet tillsammans med Tomcat. Om någon Servlet gör ett fel är risken stor att även Tomcat går ner. Följande krav gäller därför för användning:

- DSV's Tomcat får **inte användas för utveckling**, enbart väl testade Servlets får användas
- Använd **inte många Servlets** samtidigt
- Använd **aldrig system.exit(x)**
- **Städa undan databaskopplingar** som inte behövs

Man måste tyvärr räkna med att DSV's Tomcat går ner ibland och att DSV's Tomcat bara startas om automatiskt en gång per dygn men tidigare gjorda installationer installeras om automatiskt i detta för kurserna specialgjorda system.

Förfarande

Från detta formulär kan man styra om den egna webapplikationen ska köra på DSV's Tomcat:

URL till katalog eller WAR-fil:

Kommando:

Om installationen lyckades så kan webapplikationen köras från:

»http://atlas.dsv.su.se:8080/tc_4950015/«.

URL till katalog

Man har först utvecklat webapplikationen klart lokalt på en egen Tomcat. Webapplikationen finns då i en katalog under jakarta-tomcat-4.1.24/webapps, exempelvis root. Denna katalog ska man kopiera (med FTP eller SFTP) till exempelvis root-katalogen på DSV/Unix/Linus-kontot, chmodda med `chmod -R 755 <katalog>` och URL'en man ska ange i formuläret ovan blir i detta fall:

- `file://<absolut_sökväg>/<sökväg_till_katalog>`

Här är <absolut_sökväg> den absoluta sökvägen till root-katalogen på DSV/Unix/Linus-kontot och den fås genom att skriva `echo $HOME` i ett kommandofönster på DSV/Unix/Linux-kontot. Om du (Pierre Wijkman) använde en DSV e-postadress för inloggningen på detta system så kan systemet ge dig URL'en till exemplet med katalogen ROOT direkt: `file:///home/a/dsv/pierre/ROOT`.

URL till WAR-fil

Man kan även använda så kallade WAR-filer som är en komprimerad fil av katalogen för en webapplikation, exempelvis katalogen jakarta-tomcat-4.1.24/webapps/ROOT. En WAR-fil görs exempelvis enligt:

- I katalogen jakarta-tomcat-4.1.24/webapps/ROOT: `jar cf ../root.war *`

URL'en man ska ange i formuläret ovan blir i detta fall antingen:

- `jar:file://<absolut_sökväg>/<warfil.war>!/`
- `jar:http://<host>:<port>/<absolut_sökväg>/<warfil.war>!/`

Med en WAR-fil kan man således installera programmet till HTTP-servern Tomcat även utanför DSV's domän, men denna fil måste då kunna nås via en annan HTTP-server.

FAQ

Notera att det **inte ges någon handledning på hur man använder DSV's Tomcat** men här följer de frågor och svar som brukar vara vanligast:

Fråga/svar 1

Jag får inte igång min Servlet på DSV's Tomcat, vad är det för fel och vad ska jag göra?

Det vanligaste felen är:

- Filen `web.xml` är felaktig:
 - Det får inte finnas tomma rader överst
 - Det får inte finnas felaktiga radbrytningar
 - Innehållet måste stämma
 - Den måste vara välutformad
- Filen `web.xml` är på fel ställe eller saknas:
 - Den ska ligga direkt under katalogen `WEB-INF`
- Chmoddningen är fel:
 - Ska vara 755 på kataloger och filer
- URL för registrering är felaktig:
 - Ta bort applikationen och registrera med rätt URL

När man ändrat i filen `web.xml` så räcker det inte med att ladda om, man måste välja Ta bort och installera.

Fråga/svar 2

Jag har en Servlet som använder MySQL på DSV's Tomcat men den fungerar bara några timmar sen blir det flera SQL exceptions. Vad är det för fel och vad ska jag göra?

Det kan vara följande:

- MySQL servern stänger kopplingen från Servleten efter en viss tids inaktivitet
- Servleten lever fortfarande och tror att dess instans av `Connection` gäller och man får en exception när man försöker anropa databasen med en SQL-fråga

Åtgärd: Lägg till `?autoReconnect=true` i slutet av URL'en när man kopplar upp:

```
String db_url = "jdbc:mysql://" + db_comp + "/" + db_name + "?autoReconnect=true";
```

Om man har så och en Servlet försöker anropa databasen med en död koppling så kopplar MySQL-

drivrutinen upp igen automatiskt.

@Hemma

På denna sida beskrivs hur man använder en egen HTTP-server för Servlets.

Installation

Ladda ner och packa upp filen [jakarta-tomcat-4.1.24.zip](#) som innehåller en Tomcat som är speciellt konfigurerad för kurserna. Denna server går att köra på både Unix/Linux och Win32. I Unix/Linux så ska man chmodda alla sh-skript i katalogen jakarta-tomcat-4.1.24/bin körbara: `chmod 700 *.sh`.

Starta Tomcat

Innan man startar Tomcat måste man ange miljövariabeln `JAVA_HOME` som ska innehålla sökvägen till JDK:

- Unix/Linux:
 - C-skal: `setenv JAVA_HOME <sökväg till JDK>`
 - Bourne-skal:
 - `JAVA_HOME=<sökväg till JDK>`
 - `export JAVA_HOME`
 - Korn-skal: `export JAVA_HOME=<sökväg till JDK>`
 - Bash-skal: `export JAVA_HOME=<sökväg till JDK>`
- Win32:
 - 2000/XP eller senare: ingen ändring behövs
 - Övriga: Editera filen jakarta-tomcat-4.1.24\bin\startup.bat (högerklicka på filen startup.bat och välj Edit) så att översta raden stämmer med sökvägen till din JDK (nu står här: `set JAVA_HOME=C:\Program Files\j2sdk1.4.2`)

Starta sen Tomcat:

- Unix/Linux: `sh jakarta-tomcat-4.1.24/bin/startup.sh`
- Win32:
 - 2000/XP eller senare, något av följande sätt:
 - Dubbelklicka på filen: jakarta-tomcat-4.1.24\bin\startup_2.bat
 - Från ett DOS-fönster i katalogen jakarta-tomcat-4.1.24\webapps\ROOT\WEB-INF\classes: `st`
 - Övriga: Dubbelklicka på filen jakarta-tomcat-4.1.24\bin\startup.bat

Testa att allt fungerar: »<http://localhost:8080/>«.

Testa en Servlet

- Unix/Linux: `http://localhost:8080/Test01`
- Win32: Dubbelklicka på filen jakarta-tomcat-4.1.24\links\test01.html

Gör en egen Servlet

- Ändra koden i filen: jakarta-tomcat-4.1.24\webapps\ROOT\WEB-INF\classes\Test01.java
- Öppna ett kommando-fönster och gå till katalogen: jakarta-tomcat-4.1.24\webapps\ROOT\WEB-INF\classes
- Kompilera:

- Unix/Linux: `javac -classpath ../lib/tmp/activation.jar:../lib/cos.jar:../lib/tmp/mail.jar:../lib/mixer.jar:../lib/mysql-2.0.14-bin.jar:../lib/tmp/servlet.jar:.. *.java`
- Win32:
 - `sc`
 - `javac Test01.java`
- Vänta 5 sekunder på att Tomcat laddar om Servleten automatiskt
- Klicka på ladda-om i browsern

Notera att Tomcat är konfigurerad för att automatiskt ladda om en Servlet utan att man startar om Tomcat men att man ibland måste starta om servern, exempelvis om man har ändrat i någon HTML- eller CSS-fil.

Stoppa Tomcat

- Unix/Linux: `sh <sökväg>/jakarta-tomcat-4.1.24/bin/shutdown.sh`
- Win32: Stäng det DOS-fönster som skapas vid start av Tomcat

Översikt av katalogstrukturen

När man har zippat upp ZIP-filen så får man en katalogstruktur och här är en beskrivning av de saker man använder först:

- `jakarta-tomcat-4.1.24/bin`
 - Här ligger Tomcats program-filer
- `jakarta-tomcat-4.1.24/conf`
 - Här ligger viktiga konfigurations-filer, Tomcat är redan konfigurerad för kurserna så inget behöver ändras
- `jakarta-tomcat-4.1.24/webapps/ROOT`
 - Här lägger man HTML-, CSS-, bild-, ljud- och film-filer (även JSP-filer om man använder den tekniken) och man kan även anordna dessa filer i en mängd underliggande katalogstrukturer. Dessa filer nås enligt:
 - Från en browser med URL: `http://localhost:8080/test02.html`
 - Från en Servlet med Java-koden: `File f = new File(getServletContext().getRealPath("test02.html"));`
- `jakarta-tomcat-4.1.24/webapps/ROOT/WEB-INF`
 - Här ligger filen `web.xml` som beskriver programmen (Web Application Deployment Descriptor) och i den anger man exempelvis:
 - Initialiseringsparametrar
 - Säkerhetsbegränsningar
 - Associationen mellan en CLASS-fil och det namn som man vill att CLASS-filen ska nås med (från en browser)
 - Filen `web.xml` är redan konfigurerad för de filer som ligger i katalogen `classes`. Vill man lägga till fler Servlets så måste man lägga till extra `servlet` och `servlet-mapping` element i filen `web.xml`. Kontrollera noga så att filen `web.xml` är välutformad.
- `jakarta-tomcat-4.1.24/webapps/ROOT/WEB-INF/classes`
 - Här ligger alla Java- och CLASS-filer och det är här man skriver sin Java-kod för sina Servlets. Om klasserna är organiserade i paket så måste sökvägarna för dessa paket stämma, exempelvis så måste en Java-klass som heter `com.mycompany.mypackage.MyServlet` lagras i en fil med namnet `classes/com/mycompany/mypackage/MyServlet.class`
 - Använd de redan existerande filerna `UppgiftX` för uppgifterna. Vill man använda andra namn måste man ändra i filen `web.xml` (rekommenderas ej i början)
- `jakarta-tomcat-4.1.24/webapps/ROOT/WEB-INF/lib`
 - Här ligger några tillbehör som ofta behövs på kurserna:
 - `servlet.jar` för alla Servlets
 - `mixer.jar` för Mixer
 - `cos.jar` för hjälp-paketet `com.oreilly.servlet`
 - `mm.mysql-2.0.14-bin.jar` för JDBC-drivrutiner till MySQL

- activation.jar och mail.jar används för JavaMail

Använda HTTPS/SSL

Stoppa Tomcat och gör följande steg:

- Byt namn på filen jakarta-tomcat-4.1.24/conf/server.xml till jakarta-tomcat-4.1.24/conf/server_no_ssl.xml
- Hämta filen [server.xml](#) och spara den under katalogen jakarta-tomcat-4.1.24/conf
- Gör keystore: keytool -genkey -alias tomcat -keyalg RSA
 - Lösenord för keystore: changeit
 - Lösenord för certifikat: changeit
 - För och efternamn = localhost

Nu kan man surfa till exempelvis »<https://localhost:8443/Test01>«

Möjliga problem

De som använder Windows 98 kan behöva utöka miljövariabelutrymmet genom att högerklicka på startup.bat och sedan välja Properties / Memory / Initial Environment och sätta detta till minst 2816.

Om man använder en version av Java som är tidigare än J2SE 5.0 så måste man kompilera om alla Servlets i katalogen jakarta-tomcat-4.1.24/webapps/ROOT/WEB-INF/classes.

Apache

På denna sida finns hela Apache beskrivningen.

Intro

På denna sida introduceras HTTP-servern Apache som används för att köra CGI-program. Det finns två sätt att göra detta:

- Använda DSV's centrala Apache
- Använda en egen Apache

Både DSV's centrala Apache och en egen Apache kan användas för utveckling och körning men vi rekommenderar att man börjar med DSV's centrala Apache.

@DSV

På denna sida beskrivs hur man använder DSV's HTTP-server för CGI.

Adress

På DSV körs en Apache här:

- Host: atlas.dsv.su.se
- Port:
 - HTTP: 80
 - HTTPS: 443

DSV's Apache stöder CGI-program.

Krav för användning

Varje CGI/Perl-program måste börja med raderna:

```
#!/usr/local/bin/perl
alarm(30);
$POST_MAX = 5000;
use CGI::Carp qw(fatalsToBrowser);
```

Den första raden med `#!/usr/local/bin/perl` berättar för Unix/Linux-operativet vilken tolk som ska användas för körning av programmet. Den andra raden med `alarm(30)` gör att processen körs i max 30 sekunder. När man kör sina CGI-program via en HTTP-server, det vill säga när man startar dem via en browser, så startar denna HTTP-server CGI-programmet lokalt på samma maskin som HTTP-servern kör på. Eftersom inga studenterna kan logga in på denna maskin så betyder det att man inte kan påverka den process som man har startat, man kan exempelvis inte stoppa processen om den har fastnat i en evig loop. Om maskinen belastas med för många sådana processer går den till slut ner och hela maskinen måste då startas om. Den tredje raden med `$POST_MAX = 5000` gör så att max 5000 byte får uploadas av en användare till CGI-programmet. Den fjärde raden med `use CGI::Carp qw(fatalsToBrowser)` gör så att alla felmeddelanden sänds till browsern, vilket är mycket användbart vid testning.

Förfarande

Gör följande för att köra ett CGI-program på DSV's Apache:

- Skapa WWW-katalogen på ditt Unix/Linux/DSV-konto om den inte finns: `mkdir www`
- Sätt rättigheter på WWW-katalogen: `chmod 755 www`
- Lägg eller skapa CGI-programmet i denna katalog
- Sätt rättigheter på CGI-programmet: `chmod 700 <filnamn>.cgi`
- Kör CGI-programmet från en browser `http://atlas.dsv.su.se/<användarnamn>/<filnamn>.cgi`

Vi antar ovan att man arbetar med ett kommandofönster (via Telnet eller SSH) och ett filöverföringsprogram (via FTP eller SFTP). Ett exempel på CGI-program man kan testa med är:

```
#!/usr/local/bin/perl
alarm(30);
$POST_MAX = 5000;
use CGI::Carp qw(fatalsToBrowser);
print "Content-Type: text/plain\n\n";
print "I love Perl\n";
```

DSV's Apache använder suEXEC. Det betyder att CGI-skript körs som om det vore användaren som äger skriptet som körde dom och skyddskoden på CGI-skripten kan då alltså vara `rwX-----`:

- Kataloger `chmod`as 755
- Körbara CGI-program `chmod`as 700
- Skrivbara filer `chmod`as 700

FAQ

Här följer de frågor och svar som brukar vara vanligast:

Fråga/svar 1

Jag får inte igång mitt CGI-program på DSV's Apache, vad är det för fel och vad ska jag göra?

Det vanligaste felen är:

- Forbidden! You don't have permission to access x.cgi on this server beror oftast på att man har chmoddat fel
- Internal Server Error beror oftast på att Content-Type (header) saknas i Perl-koden
- Software error beror oftast på att man själv skrivit fel i Perl-koden

@Hemma

På denna sida beskrivs hur man använder en egen HTTP-server för CGI.

Installation

Använder man suveräna Linux så ingår ofta HTTP-servern Apache. I Win32 gör man enligt följande:

- Hämta installera »[Apache HTTP-server](#)« för Win32
- Apache startas automatiskt som en service
- Stoppas genom att högerklicka på ikonerna i aktivitetsfältet och välja Open Apache Monitor och sen klicka på Exit
- CGI-programmen läggs i katalogen: C:\Program Files\Apache Group\Apache2\cgi-bin
- CGI-programmen ska börja med sökvägen till Perl, exempelvis: #!C:\Perl\bin\perl.exe
- CGI-programmen nås från en browser med: `http://127.0.0.1/cgi-bin/<filnamn>.cgi`

Ett exempel på CGI-program man kan testa med är:

```
#!C:\Perl\bin\perl.exe
alarm(30);
$POST_MAX = 5000;
use CGI::Carp qw(fatalsToBrowser);
print "Content-Type: text/plain\n\n";
print "I love Perl\n";
```

Ett alternativ till Apache på Win32 är den medföljande HTTP-servern IIS eller »[Boa](#)« som är lätt att installera, liten och klarar CGI-program.

MySQL

På denna sida finns hela MySQL beskrivningen.

Intro

På denna sida introduceras databasservern MySQL. Det finns två sätt att använda denna:

- Använda DSV's centrala MySQL
- Använda en egen MySQL

Både DSV's centrala MySQL och en egen MySQL kan användas men vi rekommenderar att man börjar med DSV's centrala MySQL.

Man kan använda ett GUI mot MySQL för att få en tydligare bild av vad som sker i databasen, exempelvis:

- [MySQLFront](#) (Win32), »[MyCC](#)«
- »[Navicat](#)« (Unix, Win32, OSX)

- »[YourSQL](#)« (OSX)
- »[PhpMyAdmin](#)« (DSV's via HTTP där man anger sitt MySQL-användarnamn och -lösenord)

@DSV

På denna sida beskrivs hur man använder DSVs databasserver.

Adress

På DSV körs en MySQL-databasserver här:

- Host: atlas.dsv.su.se
- Port: 3306

Kontouppgifter

Du (Pierre Wijkman) har en **egen databas** på denna MySQL-databasserver som har följande uppgifter:

- MySQL databasnamn: db_4926495
- MySQL användarnamn: usr_4926495
- MySQL lösenord: 926495

Dessa kontouppgifter är personliga och man ska **inte visa** dem för någon annan.

Privilegier

Man kan koppla sig till sin databas på två sätt:

- Via localhost och ett server-sides program som kör på atlas.dsv.su.se
- Direkt från en annan dator

Man har följande privilegier:

- ALTER
- CREATE
- DELETE
- DROP
- INDEX
- INSERT
- SELECT
- UPDATE

Notera att databasen från början inte innehåller några tabeller utan att man får skapa dessa själv, exempelvis direkt från sitt program eller från ett grafiskt användargränssnitt.

Kopiering av databas

Om man senare vill kopiera sina data i denna databas till en annan databas så använder man MySQL-programmet mysqldump med växel --opt. För att få mer information, se man mysqldump på en Linux-maskin. Själva överföringen av data till externt system kan göras med något av SFTP, SCP, FTP (helst kerberos), wget, bränna CD och så vidare.

@Hemma

På denna sida beskrivs hur man använder en egen databasserver.

Installation

Hämta och installera en egen databasserver »[MySQL](#)«.

Starta MySQL

Gå till katalogen <sökväg>/mysql/bin och kör:

- Unix/Linux: `nohup mysqld&`
- Win32: `mysqld`

Sätta root-lösenord

Gå till katalogen <sökväg>/mysql/bin och kör:

- Unix: `mysqladmin -u root password <password_root>`
- Win32: `mysqladmin -u root password <password_root>`

Skapa en databas och ett konto för databasen

Ett konto består av:

- Ett databasnamn, <database_name>
- Ett användarnamn, <username>
- Ett lösenord, <password_account>

Gå till katalogen <sökväg>/mysql/bin och kör:

- `mysql --user=root --password=<password_root> mysql`
- `CREATE DATABASE <database_name>;`

För att få access från localhost:

- `GRANT ALL PRIVILEGES ON <database_name>.* TO '<username>'@'localhost' IDENTIFIED BY '<password_account>' WITH GRANT OPTION;`

För att få access utifrån:

- `GRANT ALL PRIVILEGES ON <database_name>.* TO '<username>'@'%' IDENTIFIED BY '<password_account>' WITH GRANT OPTION;`

Avsluta med:

- `QUIT`

Stoppa MySQL

Gå till katalogen <sökväg>/mysql/bin och kör:

- Unix/Linux: `mysqladmin -u root -p shutdown`
- Win32: `mysqladmin -u root -p shutdown`

Man måste man stoppa MySQL innan man stänger av datorn.

Tidigare tentor

Tentamen

Internetprogrammering II

1. Beskriv GET och POST
2. Hur läser ett CGI-program data från ett formulär på en webb-sida?
3. Hur sätter ett CGI-program en kaka?
4. Beskriv vad följande tre Perl-koder gör:
 - `print "H"; sleep 10;`
 - `print "H\n"; sleep 10;`
 - `$| = 1; print "H"; sleep 10;`
5. Hur fungerar kommandot `fork`?
6. Hur skapar man en server socket?
7. Beskriv ditt gesällprov

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 1998-08-28

Tentamen

Internetprogrammering II

1. I vilket sammanhang använder man POP-protokollet? (1 poäng)
2. En HTTP-server skickar ut två typer av information till en klient som begär en webb-sida, beskriv denna information (2 poäng)
3. Beskriv GET och POST (2 poäng)
4. Hur sätter och läser ett CGI-program en kaka? (2 poäng)
5. Nämn tre lagringsplatser som kan användas för att lagra information från en användare (3 poäng)
6. Hur skapar man en server socket? Hur fungerar en trådat chatserver? (5 poäng)

För G krävs > 10 poäng. Inga hjälpmedel. Pierre A. I. Wijkman ▫ 1999-03-12

Tentamen

Internetprogrammering II

1. Ange (minst) fem vanliga omgivningsvariabler som skickas med till ett CGI-program (1 poäng)
2. Beskriv HTTP-protokollet (2 poäng)
3. Beskriv (minst) tre vanliga dokumenthuvuden som kan skickas med i ett HTTP-request (2 poäng)
4. Beskriv hur klientstyrd och serverstyrd omladdning fungerar samt hur detta implementeras (5 poäng)
5. Beskriv hur en förbindelse upprättas mellan klient och server (socket-nivå) (5 poäng)

För G krävs > 10 poäng. Inga hjälpmedel. Pierre A. I. Wijkman ▫ 1999-04-07

Tentamen

Internetprogrammering II

1. Beskriv detaljerat hur du löste uppgift 4, kontinuerlig uppkoppling, chatserver (1 poäng)
2. Beskriv detaljerat hur du löste uppgift 5, gesällprovet (1 poäng)

För G krävs 2 poäng. Inga hjälpmedel. Pierre A. I. Wijkman ▫ 1999-05-07

Tentamen

Internetprogrammering II

1. En HTTP-server (webb-server) skickar ut två typer av information till en klient som begär en webb-sida, beskriv denna information
2. Varför använder man `nph-*.cgi` program vid serverstyrd omladdning?
3. Hur fungerar kommandot `flock`?
4. Hur läser ett CGI-program en kaka?
5. Vad gör följande Perl-kod: `select H; $| = 1;`
6. Beskriv detaljerat hur din chatserver i uppgift 4 är implementerad (inklusive hur man skapar en server socket mm)

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 1999-08-31

Tentamen

Internetprogrammering II

Beskriv detaljerat hur du löste Gesällprovet. I beskrivningen skall finnas både en generell (vanligt språk) beskrivning och en specifik (Perl) beskrivning. Den specifika beskrivningen behöver inte vara hela programmet utan bör endast visa hur olika, för uppgiften speciella, funktioner implementerats. Koden skall skrivas på ett mycket välstrukturerat sätt (exempelvis ej med rak vänster). Den specifika beskrivningen skall kommenteras med vanligt språk. Totalt bör hela beskrivningen omfatta minst 5 sidor. Alla dessa villkor måste vara uppfyllda för att uppgiften skall ge 1 p.

För G krävs 1 poäng. Inga hjälpmedel. Pierre A. I. Wijkman ▫ 1999-08-31

Tentamen

Internetprogrammering II

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

1. Beskriv GET och POST
2. Beskriv vad följande tre Perl-koder gör:
 - `print "H"; sleep 10;`
 - `print "H\n"; sleep 10;`
 - `$| = 1; print "H"; sleep 10;`
3. Hur fungerar kommandot `flock`? Visa ett exempel
4. Vad används DBD/DBI-modulerna till?

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2000-03-06

Tentamen

Internetprogrammering II

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

1. Ange fyra omgivningsvariabler som skickas med till ett CGI-program
2. Ange och beskriv två kommandon i HTTP-protokollet
3. Beskriv hur databaskopplingar i CGI fungerar
4. Beskriv hur klient- och serverstyrd omladdning fungerar

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2000-04-27

Tentamen

Internetprogrammering II

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

1. Beskriv generellt hur databaskopplingar i CGI fungerar
2. Visa med ett kod-exempel hur databaskopplingar i CGI fungerar

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2000-08-25

Tentamen

Internetprogrammering II

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Beskriv först generellt och med figur hur databaskopplingar i CGI fungerar. Visa sedan detta med ett exempel i Perl-kod.

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2000-12-16

Tentamen

Internetprogrammering II

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Beskriv ingående 3 sätt att implementera sessionshantering. Exemplifiera med kod.

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2001-03-10

Tentamen

Internetprogrammering II

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Beskriv GET och POST ingående.

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2001-04-21

Tentamen

Internetprogrammering II

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Beskriv först generellt ditt gesällarbete (sista uppgiften). Beskriv sedan i större detalj hur ditt gesällarbete är kodat (du behöver givetvis inte beskriva all kod).

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2001-08-25

Tentamen

Internetprogrammering II

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Vilken typ av information kan komma från en HTTP-klient till ett HTTP-server-sides-program och hur kan detta program komma åt denna information?

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2001-11-17

Tentamen

Internetprogrammering II

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Beskriv generellt (och med figur) hur databaskopplingar på HTTP-serversidan fungerar. Visa även med ett minimalt kodexempel.

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2001-12-17

Tentamen

Internetprogrammering II

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Beskriv generellt (med text och figur) hur file-uploads fungerar. Visa även med ett minimalt (men exakt) kodexempel.

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2002-03-09

Tentamen

Internetprogrammering II

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Visa med ett minimalt (men exakt) kodexempel hur man sätter och läser en kaka (cookie).

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2002-08-15

Tentamen

Internetprogrammering II

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Visa med ett minimalt (men exakt) kodexempel hur ett HTTP-server-sides-program skriver ut värdet på de parametrar som anges i följande anrop: `http:// ... osv ... /program_namn?x=1&y=2`

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2002-11-19

Tentamen

Internetprogrammering II

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Beskriv generellt (och med figur) hur databaskopplingar på HTTP-serversidan fungerar. Visa även med ett minimalt kodexempel (det behöver inte vara så omfattande som uppgift 5a på kursen, gör det minimalt).

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2002-12-11

Tentamen

Internetprogrammering II

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Beskriv först generellt ditt gesällarbete (sista uppgiften). Beskriv sedan i större detalj hur de centrala delarna i ditt gesällarbete är kodat (man ska givetvis inte beskriva all kod).

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2003-01-10

Tentamen

Internetprogrammering II

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Beskriv först generellt ditt gesällarbete (sista uppgiften). Beskriv sedan i större detalj hur de centrala delarna i ditt gesällarbete är kodat (man ska givetvis inte beskriva all kod).

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2003-08-21

Tentamen

Internetprogrammering II

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Beskriv detaljerat fördelar och nackdelar med GET och POST. Visa även med ett enkelt kodexempel hur man i ett serversides-program kommer åt information som sänts från klientsidan med GET och POST.

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2003-11-15

Tentamen

Internetprogrammering II

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Beskriv generellt den teknik du använt för att separera HTML-kod och HTTP-serversides-kod. Exemplifiera med kod.

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2003-12-19

Tentamen

Internetprogrammering II

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Beskriv först generellt ditt gesällarbete (sista uppgiften). Beskriv sedan i större detalj hur de centrala delarna i ditt gesällarbete är kodat (man ska givetvis inte beskriva all kod).

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2004-01-09

Tentamen

Internetprogrammering II

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Visa med ett minimalt (men exakt) kodexempel hur ett HTTP-server-sides-program skriver ut (text/plain) värdet på de två text-fält (med namn x och y) som finns i ett formulär på en HTML-sida.

Inga hjälpmedel. Pierre A. I. Wijkman ◻ 2004-08-18

Tentamen

Internetprogrammering II

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga.

Vilken viktig åtgärd bör man vidta om ett HTTP-serversidesprogram ska spara data på fil (så att filen inte blir korrupt)?

Inga hjälpmedel. Pierre A. I. Wijkman ◻ 2004-11-27

Tentamen

Internetprogrammering II

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Beskriv ingående alla sätt att implementera sessionshantering. Exemplifiera ett av sätten med kod.

Inga hjälpmedel. Pierre A. I. Wijkman ◻ 2004-12-18

Tentamen

Internetprogrammering II

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Konstruera ett minimalt HTTP-serversidesprogram och tillhörande HTML-dokument som tillsammans implementerar sändning av e-post.

Inga hjälpmedel. Pierre A. I. Wijkman ◻ 2005-03-30

Externa källor

På denna sida finns länkar till externa källor. Tipsa gärna om fler bra länkar.

Servlets

- » [SUN: Java Servlet Technology](#)«
- » [SUN: Java Servlet Technology Documentation](#)«
- » [SUN: Servlets 2.3 API Specification](#)«
- » [Servlets.com](#)«
- » [Servlets.com: The Problems with JSP](#)«
- » [ObjectWeb: Enhydra XMLC](#)«

Lektioner

- » [SUN: Java Servlet Technology](#)«
- » [SUN: Fundamentals of Java Servlets](#)«
- » [O'Reilly: Web Server Java: Servlets and JSP](#)«
- » [Core Servlets and Java Server Pages](#)«

Tomcat 4 konfig

- » [Configuring & Using Apache Tomcat 4](#)«
- » [Using Apache Tomcat 4](#)«

Perl

- » [Comprehensive Perl Archive Network \(CPAN\)](#)«
- » [Comprehensive Perl Archive Network \(CPAN\) Search](#)«
- » [Perldoc.com](#)«
- » [Perl.org: The Perl Directory](#)«
- » [O'Reilly: Perl.com](#)«
- » [Perl Monks](#)«
- » [PERL 5.6 Manual and Documentation](#)«

Databaser med DBI/DBD

- » [MySQL and Perl for the Web](#)«
- » [Programming the Perl DBI](#)«
- » [Introduction to MySQL \(using Perl DBI\)](#)«

Grafik med GD.pm

- » [GD.pm](#)«

Lektioner

- » [Introduction to Perl](#)«
- » [LearnPerl.org: The site for people learning Perl](#)«
- » [Perl Regular Expression Tutorial](#)«
- » [Perl Regular Expressions](#)«

CGI

- » [CGI Programming FAQ](#)«
- » [Webreference: Perl](#)«
- » [CGI.pm - a Perl5 CGI Library](#)«
- » [Perl Module: CGI.pm](#)«

Uppgifter

På denna sida finns uppgifterna till kursen. Läs först noga alla [krav](#) och sedan alla [tips](#). Siffror inom () är kapitelhänvisningar.

Output

Text

- 1.1.1 [Säker filhantering](#) obligatorisk (1-3)
- 1.1.2 [Omgivningsvariabler](#) obligatorisk (1-4)

Bild

- 1.2 [Grafikgenerering](#) obligatorisk (6)

Omladdningar

- 1.3.1 [Klientstyrd](#) obligatorisk (5, 6)
- 1.3.2 [Serverstyrd: Tidsstyrd](#) frivillig (6)
- 1.3.3 [Serverstyrd: Tidsstyrd: Olika mime-typer](#) frivillig (6)
- 1.3.4 [Serverstyrd: Händelsestyrd](#) frivillig (6)

Input

URL och länkar

- 2.1 [GET](#) obligatorisk (1-5)

Formulär

- 2.2.1 [GET](#) obligatorisk (1-5)
- 2.2.2 [POST](#) obligatorisk (1-5)
- 2.2.3 [Uppladdning av fil](#) obligatorisk (4)

Kodseparation

- 3.1 [Statisk mängd data](#) obligatorisk (1-3)
- 3.2 [Variabel mängd data](#) obligatorisk (1-5)

Sessionshantering

Omskrivning

- 4.1.1 [Länkar](#) obligatorisk (7)
- 4.1.2 [Formulär](#) obligatorisk (7)

Lagra på klienten

- 4.2.1 [Kakor](#) obligatorisk (7)
- 4.2.2 [Säkra kakor](#) frivillig IP2, obligatorisk IP2+ (7)

E-post

Sändning

- 5.1.1 [Utan bifogade filer](#) obligatorisk (19)
- 5.1.2 [Med bifogade filer](#) obligatorisk (4, 19)

Mottagning

- 5.2.1 [Utan bifogade filer](#) frivillig IP2, obligatorisk IP2+
- 5.2.2 [Med bifogade filer](#) frivillig

Databaser

Lättvikts

- 6.1 [DBM](#) frivillig IP2, obligatorisk IP2+

Relations/SQL

- 6.2.1 [I/O](#) obligatorisk (9)
- 6.2.2 [Transaktioner](#) frivillig IP2, obligatorisk IP2+ (4, 9)
- 6.2.3 [Effektivitet](#) frivillig (9)

Innehållshantering

Manuellt innehåll

- 7.1.1 [Publiceringssystem](#) frivillig
- 7.1.2 [Syndikering](#) frivillig

Automatiskt innehåll

- 7.2.1 [Kanaler](#) frivillig
- 7.2.2 [Fritextsökning](#) frivillig

Gesällprov

- 8 [Gesällprov](#) obligatorisk

Krav

På denna sida finns de krav som gäller för uppgifterna.

Samarbete och betyg

- Alla uppgifter utom gesällprovet ska göras i grupp om 1-2 personer, gesällprovet ska göras ensam
- Vissa av uppgifterna är frivilliga men desto fler av dessa som man gör desto större blir sannolikheten för att få betyget VG/5
- Man kan givetvis ändå få betyget VG/5 om man gör ett väldigt bra gesällprov
- Bästa gesällprov till deadline 1 vinner ett pris

Kodens utseende

En vanlig orsak till betyget R är att nedanstående krav inte är uppfyllda.

All kod

- All kod ska vara välstrukturerad
- Man får själv bestämma stil på koden men man måste vara konsekvent
- Använd inte en editor som använder kontrolltecken och tabbar för indentering, koden ska kunna läsas och se bra ut i en vanlig enkel editor som exempelvis Notepad och därmed även i en browser
- Designen på det (eventuella) grafiska användargränssnittet är fri men försök göra det så snyggt som möjligt (man bör alltså inte göra programmen identiska med de ofta lite tråkiga exempelprogram som ges för varje uppgift)

HTTP-serversides kod

- All HTTP-serversides kod ska vara välkommenterad med en kommentar per klass och metod
- HTML- och HTTP-serversides kod ska vara:
 - Fullständigt separerade för teknikerna Servlets och CGI, det får alltså inte finnas:
 - HTTP-serversides kod i HTML-dokument
 - HTML-kod i HTTP-serversides kod
 - Så separerade som möjligt för övriga tekniker (ASP.NET, JSP, PHP med flera)

Tips

På denna sida finns tips för uppgifterna.

Ordning och organisation

- Använd en katalog för varje huvuduppgift
- Under hela konstruktionsprocessen:
 - Välstrukturerad kod
 - Bra namn på klasser, variabler, konstanter

Evolutionär utveckling

Många moderna programmeringsspråk består av:

- En liten språkkärna
- Ett mycket stort bibliotek av färdigkonstruerade klasser

Den självklara fördelen är att man slipper konstruera en stor mängd saker som annars skulle ta mycket lång tid att göra men nackdelen är att det kan vara svårt att veta och hitta vad som finns samt att man inte får samma grundläggande förståelse.

Matematik är ingenting man förstår, matematik är något som man vänjer sig vid

von Neumann

Till att börja med bör man skaffa sig en övergripande förståelse för det område man ska ge sig in på. Detta görs bäst med hjälp av böcker, tidningar och/eller de tutorials och artiklar som finns på Internet. När man har skaffat sig en denna övergripande förståelse så bör man så fort som möjligt

testa några av de exempel man stött på. Det är ofta först när man praktiskt tillämpar ett område som man lär sig, eller vänjer sig vid, området och därmed får en förankring eller förståelse av beteendet hos objekten man laborerar med.

Man bör inte skriva av koden "rakt av" utan man bör istället:

- Konstruera en minimal körbar kärna (med sin egen stil på koden)

Dett gör man för varje nytt område man ger sig in på. Man kan sedan:

- Förändra denna minimala körbara kärna mot ett önskbart mål med **små förändringar** av koden

Detta är mycket viktigt! Man går mot ett önskat mål med:

- Små förändringar av koden
- Kompilering
- Testkörning

Man släpper aldrig kontakten med det fungerande programmet.

We are like sailors who on the open sea must reconstruct their ship but are never able to start afresh from the bottom. Where a beam is taken away a new one must at once be put there, and for this the rest of the ship is used as support. In this way, by using the old beams and driftwood the ship can be shaped entirely anew, but only by gradual reconstruction.

Otto Neurath

Detta är en mycket viktig tumregel. Försöker man förändra koden i stora steg så blir det fel och dessa fel är ofta svåra att lokalisera och rätta till. Ofta samverkar flera fel och detta gör det givetvis ännu svårare.

Ofta är stora program konstuerade av många lager/nivåer där varje nivå består av program konstruerade enligt ovanstående enkla principer. Man använder sedan program på en nivå som enheter för konstruktion av program på nästa nivå, och även denna process följer de ovan anvisade principerna.

Övrigt

- Detta är inte övningsuppgifter och man måste läsa kursboken och viss information som finns på Internet för att klara uppgifterna
- Alla program är testade i DSVs miljö för både Win32 och (det överlägsna :-) Unix/Linux
- Alla program går att utveckla lokalt ej uppkopplad (på Win32 och Unix/Linux) men man måste ha TCP/IP installerat på sin dator (vilket man har om man exempelvis surfar på Internet)
- Man kan använda IP-adressen 127.0.0.1 eller localhost som motsvarar den egna maskinen

Output: Text: Säker filhantering

Allmänt

Uppgiften belyser hur man genererar (en statisk mängd) dynamisk text output till användaren. Den tar även upp säker filhantering.

Uppgift

Gör en räknare som visar totala antalet besök av sidan. Värdet på antalet träffar ska lagras i en fil på serversidan och detta värde ska visas med mime-typen `text/plain`.

Eftersom HTTP-serversides-program kan anropas av många HTTP-klienter samtidigt så måste man i denna uppgift tänka på att undvika krockar i filhanteringen.

Exempel

Ett [enkelt exempel](#).

Tips

CGI/Perl

Använd funktionen `flock` för att undvika krockar.

Lås en fil runt hela programmet som ska köras:

```
open (LOCK, ">lock");
flock LOCK, 2;

...

close LOCK;
```

Ofta är det bra om man använder en speciell fil enbart får låsningen.

Servlets

För att hitta en fil, använd:

```
File f = new File(getServletContext().getRealPath(fileName));
```

Använd synkronisering för att undvika krockar.

Output: Text: Omgivningsvariabler

Allmänt

Uppgiften belyser hur man genererar (en variabel mängd) dynamisk text output till användaren. Den tar även upp vilken information som ett HTTP-serversides-program alltid får som input.

Uppgift

Gör ett HTTP-serversides-program som skriver ut samtliga omgivningsvariabler. För varje variabel ska både namnet och dess värde skrivas ut. Varje sådant par ska skrivas ut på en egen rad. Svaret ska vara av mime-typen `text/plain`.

Exempel

Ett [enkelt exempel](#). Notera att det inte ser exakt likadant ut för Servlets som för CGI. Det sänds ganska mycket information till HTTP-serversides-programmet, bland annat anger `REMOTE_ADDR` varifrån uppkopplingen skett `HTTP_USER_AGENT` anger vilken typ av browser som anropat HTTP-serversides-programmet.

Tips

CGI/Perl

Variabeln `%ENV` är användbar.

Servlets

Det räcker om man skriver ut den information som finns i `System.getenv()` (kräver J2SE 5.0+). Notera dock att man kan nå mer och intressant information med speciala metoder, se `jakarta-tomcat-4.1.24/webapps/ROOT/WEB-INF/classes/Test03.java` som finns i den för kursen konfigurerade Tomcat.

Output: Bild: Grafikgenerering

Allmänt

Uppgiften belyser hur man gör grafik dynamiskt på serversidan.

Uppgift

Gör ett HTTP-serversides-program som vid anrop returnerar en dynamiskt konstruerad JPG- eller PNG-bild. Bilden ska vara annorlunda vid varje anrop.

Exempel

Ett [enkelt exempel](#) samt ett annat [enkelt exempel](#).

Tips

CGI/Perl

Perl-modulen `GD.pm` finns enbart installerad på `atlas.dsv.su.se`. På grund av detta så kan man inte testa programmet via kommandofönstret utan får testa direkt via browsern.

Perl-modulen `GD.pm` är bra men det finns en stor mängd andra bra Perl-moduler för dynamisk grafikgenerering, exempelvis:

- PerlMagick vilken är enormt kraftfull (men kräver en hel del av maskinen)
- GIFgraph för att rita diagram och grafer
- Gimp

Servlets

Skriv följande överst:

```
System.setProperty("java.awt.headless", "true");
```

Detta gör att grafiken ritas internt i Java och inte i det underliggande systemet (X11 på Unix/Linux) och fungerar bara när HTTP-servern använder JDK 1.4 eller senare.

Output: Omladdningar: Klientstyrd

Allmänt

Uppgiften belyser hur klientstyrd omladdning (client pull) görs.

Uppgift

Gör ett HTTP-serversides-program som implementerar klientstyrd omladdning via HTTP.

Exempel

Ett [enkelt exempel](#).

Tips

CGI/Perl

Använd modulen `CGI.pm`.

Servlets

Inget tips ges.

Output: Omladdningar: Serverstyrd: Tidsstyrd

Allmänt

Uppgiften belyser hur tidsstyrd serverstyrd omladdning (server push) görs.

Uppgift

Gör ett HTTP-serversides-program som implementerar serverstyrd omladdning. HTTP-serversides-programmet ska skriva ut följande information med mime-typen `text/plain`:

- Tid
- En räknare som räknar ner från 10 till 1

Sidan ska laddas om en gång i sekunden.

Exempel

Ett [enkelt exempel](#). Använd exempelvis browsern Firefox, Internet Explorer hanterar inte mime-typen `multipart/x-mixed-replace` ännu.

Tips

Vill man göra något liknande i Internet Explorer eller någon annan browser som ännu inte hanterar mime-typen `multipart/x-mixed-replace` så kan man använda följande teknik på HTTP-klient-sidan:

- Ett JavaScript-program anropar med jämna mellanrum:
 - Ett äpple som i sin tur anropar:
 - Ett HTTP-serversides-program som avgör om något ska förändras och svarar till:
 - Äpplet som svarar till:
 - JavaScript-programmet som beroende på svaret kan förändra HTML-sidan

Men detta är ju mer likt klientstyrd omladdning.

CGI/Perl

Man måste här använda "non parsed headers", det vill säga vi vill inte att HTTP-servern ska lägga till några HTTP-headers. Därför måste HTTP-serversides-programmet börja med `nph-`, exempelvis är namnet `nph-x.cgi` OK.

Använd modulen `CGI.pm`.

Servlets

Använd `»com.oreilly.servlet«`.

Output: Omladdningar: Serverstyrd: Tidsstyrd: Olika mime-typer

Allmänt

Uppgiften belyser hur tidsstyrd serverstyrd omladdning (server push) med olika mime-typer görs.

Uppgift

Gör ett HTTP-serversides-program som implementerar serverstyrd omladdning med olika mime-typer. HTTP-serversides-programmet ska omväxlande skriva ut innehåll av mime-typerna:

- `text/plain`
- `image/gif` (eller någon annan bildtyp)
- `text/html`

Exempel

Ett [enkelt exempel](#). Använd exempelvis browsern Firefox, Internet Explorer hanterar inte mime-typen multipart/x-mixed-replace ännu.

Tips

Se uppgiften [Output: Omladdningar: Serverstyrd: Tidsstyrd](#).

Output: Omladdningar: Serverstyrd: Händelsestyrd

Allmänt

Uppgiften belyser hur händelsestyrd serverstyrd omladdning (server push) görs.

Uppgift

Gör ett HTTP-serversides-program som implementerar händelsestyrd serverstyrd omladdning. Sidan ska alltså laddas om om någon specifik händelse inträffar på HTTP-serversidan.

Exempel

Ett [enkelt exempel](#). Detta är en besöksräknare som hela tiden visar korrekt antal besök och för att testa att det fungerar kan man i ett annat browserfönster köra [exemplet från uppgiften Output: Text: Säker filhantering](#) och ladda om sidan i detta fönster upprepade gånger. Använd exempelvis browsern Firefox, Internet Explorer hanterar inte mime-typen multipart/x-mixed-replace ännu.

Tips

Se uppgiften [Output: Omladdningar: Serverstyrd: Tidsstyrd](#).

Input: URL och länkar: GET

Allmänt

Uppgiften belyser hur man sänder information från en HTTP-klient till ett HTTP-serversides-program med URL och länkar med HTTP-metoden GET.

Uppgift

Gör ett HTML-dokument med en länk:

- Som pekar på ett HTTP-serversidesprogram
- Som har en mängd namn/värde-par som argument

Gör sen det HTTP-serversides-program som tar emot denna information och sedan sänder tillbaka denna information till HTTP-klienten som text/plain. Både namnet och värdet för allt som sänds in till HTTP-serversides-programmet ska skrivas ut på egna rader. HTTP-serversides-programmet ska fungera för ett godtyckligt antal godtyckliga namn/värde-par.

Testa även HTTP-serversides-programmet genom att anropa det direkt från browserns adressrad med olika namn/värde-par.

Exempel

Ett [enkelt exempel](#).

Tips

CGI/Perl

Använd CGI.pm-modulens param-funktion.

Servlets

Använd:

```
Enumeration names = request.getParameterNames();
```

Input: Formulär: GET

Allmänt

Uppgiften belyser hur man sänder information från en HTTP-klient till ett HTTP-serversides-program med formulär med HTTP-metoden GET.

Uppgift

Gör ett HTML-dokument med ett formulär bestående av diverse olika komponenter (se exemplet) som länkar till ett HTTP-serversides-program och använder metoden GET.

Gör sen det HTTP-serversides-program som tar emot denna information och sedan sänder tillbaka denna information till HTTP-klienten som text/plain. Både namnet och värdet för allt som sänds in till HTTP-serversides-programmet ska skrivas ut på egna rader. HTTP-serversides-programmet ska fungera för ett godtyckligt antal godtyckliga namn/värde-par.

Exempel

Ett [enkelt exempel](#).

Tips

Titta på hur HTML-formuläret är uppbyggt i HTML-dokumentet, exempelvis:

```
<form method="get" action="2.2.1.cgi">  
...  
</form>
```

Input: Formulär: POST

Allmänt

Uppgiften belyser hur man sänder information från en HTTP-klient till ett HTTP-serversides-program med formulär med HTTP-metoden POST.

Uppgift

Gör ett HTML-dokument med ett formulär bestående av diverse olika komponenter (se exemplet) som länkar till ett HTTP-serversides-program och använder metoden POST.

Gör sen det HTTP-serversides-program som tar emot denna information och sedan sänder tillbaka denna information till HTTP-klienten som text/plain. Både namnet och värdet för allt som sänds in till HTTP-serversides-programmet ska skrivas ut på egna rader. HTTP-serversides-programmet ska fungera för ett godtyckligt antal godtyckliga namn/värde-par.

Exempel

Ett [enkelt exempel](#).

Tips

Inga tips ges.

Input: Formulär: Uppladdning av fil

Allmänt

Uppgiften belyser hur man sänder en fil från en HTTP-klient till ett HTTP-serversides-program med formulär.

Uppgift

Gör ett HTML-dokument med ett formulär bestående av en komponent för uppladdning av en fil (se exemplet) som länkar till ett HTTP-serversides-program.

Gör sen det HTTP-serversides-program som tar emot denna fil och svarar på följande sätt:

- Om filens mime-typ är text/plain, image/jpeg eller image/x-png:
 - Innehållet i filen, enkel text ska visas som text/plain, bilder ska visas som image/jpeg eller image/x-png
- Annars:
 - Filens namn (utan sökväg)
 - Filens mime-typ
 - Filens storlek

Var noga med att sätta en övre gräns för hur mycket information HTTP-serversides-programmet får ta emot. Frivillig utökning är att även hantera mime-typen image/gif och att göra en kontroll om användaren sänder med någon fil.

Exempel

Ett [enkelt exempel](#). Testa gärna med både text- och bild-filer.

Tips

Betrakta noga exemplets speciella HTML-formulär. HTML-formuläret måste kodas enligt:

```
<form method="post" enctype="multipart/form-data" action="2.2.3.cgi">
```

Det viktiga här är `enctype="multipart/form-data"`.

Fälten för nerladdning av filer ser ut så här:

```
<input type="file" name="file" size="35">
```

Det viktiga här är `type="file"`.

CGI/Perl

Man får tag på en referens till den uppladdade filen (det vill säga ett fil-handtag) med:

```
$fh = param('file');
```

Man får tag på mime-typen med:

```
$mimetype = uploadInfo($fh)->{'Content-Type'};
```

När man ska ta bort sökvägen från filnamnet måste man tänka på att användaren kan sitta vid olika operativsystem:

```
$filename = $fh;  
$filename =~ s/.*\\//g; # Win32  
$filename =~ s/.*\\//g; # Unix  
$filename =~ s/.*\\//g; # Mac
```

Servlets

Använd »[com.oreilly.servlet](#)«. Några spridda kodrader som kan vägleda:

```
File dir = (File)getContext().getAttribute("javax.servlet.context.tempdir");  
MultipartRequest mreq = new MultipartRequest(request, dir.getAbsolutePath(), 500000);  
String contentType = mreq.getContentType("file");  
File file = mreq.getFile("file");
```

Kodseparation: Statisk mängd data

Allmänt

Uppgiften belyser hur man genererar (en statisk mängd) dynamisk HTML output till användaren på ett sätt där HTTP-serversides-koden är så separerad som möjligt från HTML-koden. Att separera koden är bra eftersom:

- Författare och designers kan arbeta isolerat med HTML-dokumentet
- Programmerare kan arbeta isolerat med HTTP-serversides koden

Uppgift

Gör om uppgiften [Output: Text: Säker filhantering](#) så att den svarar med HTML istället för text. Gör detta utan att blanda HTML- och HTTP-serversides-kod.

Exempel

Ett [enkelt exempel](#).

Tips

CGI/Perl

Använd markörer i HTML-koden som liknar variablernas namn i HTTP-serversides-programmet.

Man kan använda markeringar i HTML-dokumentet, exempelvis `---0---`, `---1---` och så vidare. Sedan substituerar man dessa markeringar mot någonting annat i CGI-programmet:

```
open(HTML, "<x.html");
while($row = <HTML>) {
    $row =~ s/---0---/$time/;
    $row =~ s/---1---/$hits/;
    print "$row\n";
}
close(HTML);
```

Problemet med detta är att:

- Man måste ha en rad kod för varje substituering, vilket är ineffektivt
- Kopplingen mellan orden/markeringarna och variablerna i CGI-programmet är svag

Det är bättre att använda markeringar som `---$hits---`, `---$time---` och så vidare. Här motsvarar `$hits` och `$time` de variabler i CGI-programmet som ska ersätta orden/markeringarna. Man får då en rad kod för alla substitutioner (vilket är effektivare) och man får en bra koppling mellan orden/markeringarna och variablerna i CGI-programmet. Själva substitutionen blir nu:

```
$row =~ s/---(.*?)---/eval $1/eg;
```

Snyggt men vad är detta?

Servlets

Använd exempelvis »[Mixer](#)« men tänk på att programmet är i beta version och troligtvis har brister. Man behöver då JAR-filen »[mixer.jar](#)« som redan finns i den för kursen konfigurerade Tomcat-servern. Man kan använda en annan teknik eller göra ett eget liknande program istället om man vill.

Kodseparation: Variabel mängd data

Allmänt

Uppgiften belyser hur man genererar (en variabel mängd) dynamisk HTML output till användaren på ett sätt där HTTP-serversides-koden är så separerad som möjligt från HTML-koden.

Ofta vill man kunna sända en variabel mängd output från ett HTTP-serversides-program och

därmed skapa HTML-dokument av variabel storlek. Man vill kanske presentera svaret från en fråga till en databas i en HTML-tabell med variabel storlek. Vi ska nu titta på ett knep för hur man gör detta utan att ge upp vårt krav på att HTTP-serversides-program och HTML-dokument ska vara separerade.

Uppgift

Gör om uppgiften [Output: Text: Omgivningsvariabler](#) så att den svarar med HTML istället för text. Gör detta utan att blanda HTML- och HTTP-serversides-kod.

Exempel

Ett [enkelt exempel](#).

Tips

CGI/Perl

Använd två markörer runt de ställen i HTML-dokumentet för HTML-kod som ska upprepas, exempelvis:

```
<!-- ==xxx== -->
...
<!-- ==xxx== -->
```

HTTP-serversides-programmet kan då:

- Läs in all HTML-kod från en fil till en sträng
- Använda funktionen `split` med avseende på markörerna ovan och på så sätt få tre delsträngar
- Skriva ut första delsträngen
- Skriva ut andra delsträngen upprepade gånger med substitution enligt uppgiften [Kodseparation: Statisk mängd data](#) (använd en temporär sträng vid substitutionen annars försvinner markörerna vid första substitutionen)
- Skriva ut tredje delsträngen

Se HTML-dokumentet i exemplet.

Servlets

Se tips för uppgiften [Kodseparation: Statisk mängd data](#).

Sessionshantering: Omskrivning: Länkar

Allmänt

Uppgiften belyser ett sätt att implementera sessionshantering vid användning av länkar.

Uppgift

Använd två HTML-dokument:

- På det första HTML-dokumentet ska det finnas två länkar till samma HTTP-serversides-program men med två olika namn/värde-par argument
- På det andra HTML-dokumentet ska det finnas två länkar till ett annat HTTP-serversides-program med två olika namn/värde-par argument

När man klickat på en av länkarna på den första HTML-dokumentet ska den andra HTML-dokumentet visas. Då ska informationen om vilket namn man klickat på i första HTML-dokumentet föras över till länkarna i det andra HTML-dokumentet. När man till slut klickar på en av länkarna i det andra HTML-dokumentet så ska informationen från båda dessa klick visas.

Exempel

Ett [enkelt exempel](#).

Tips

Inget tips ges.

Sessionshantering: Omskrivning: Formulär

Allmänt

Uppgiften belyser ett sätt att implementera sessionshantering vid användning av formulär.

Uppgift

Läs in information från en klient från två HTML-formulär på två olika HTML-dokument. Använd dolda fält i det andra HTML-formuläret för att spara informationen som sändes från det första HTML-formuläret. Informationen ska i båda fallen skickas med POST.

Exempel

Ett [enkelt exempel](#).

Tips

Formuläret i det andra HTML-dokumentet har dolda fält med markörer:

```
<input type="hidden" name="name" value="---$name---">
```

Det är `type="hidden"` och `value="---$name---"` som är viktigt här.

Sessionshantering: Lagra på klienten: Kakor

Allmänt

Uppgiften belyser sessionshantering med hjälp av kakor. När man använder kakor måste man tänka på att:

- Kak-funktionen kan vara avstängd hos klienterna vilket medför att man inte enbart kan förlita sig till kakor
- En kaka sätts per dator/konto och om det är flera personer som använder samma dator/konto så kan det vara farligt

Att informationen som kan lagras i en kaka är begränsad är inget problem eftersom det räcker med att man sparar ett unikt sessions-ID, det vill säga ett långt slumpmässigt nummer, i kakan. Man kan sedan använda detta sessions-ID som en nyckel på serversidan för att nå godtyckligt mycket information associerad till detta sessions-ID. Vare sig man än använder omskrivning eller kakor så är det bäst att använda just ett sessions-ID. Vill man ha hög säkerhet så använder man detta sessions-ID enbart för en enda session, nästa gång användaren besöker webbplatsen så får användaren ett annat sessions-ID. Detta eftersom browsrar ofta cashar besökta HTML-dokument och då sparas även sessions-ID't om man använder omskrivningar.

Uppgift

Gör ett HTTP-serversides-program som sätter och läser följande två kakor hos klienten:

- time = <tiden då kakan skapades>
- name = <ett namn>

Se exemplet. Kakan ska "leva" i 3 timmar.

Exempel

Ett [enkelt exempel](#).

Tips

CGI/Perl

Använd CGI.pm-modulens cookie-funktion.

Servlets

Man kan välja om man vill använda rena kakor eller det inbyggda stöd som finns i Session Tracking API. Använder man rena kakor så använd klassen `javax.servlet.http.Cookie`.

Sessionshantering: Lagra på klienten: Säkra kakor

Allmänt

Uppgiften belyser sessionshantering med hjälp av säkra kakor som enbart sänds om överföringen är säker via HTTPS/SSL.

Uppgift

Gör ett HTTP-serversides-program som sätter och läser följande två säkra kakor hos klienten:

- time_secure = <tiden då kakan skapades>
- name_secure = <ett namn>

Se exemplet. Kakan ska "leva" i 3 timmar.

Exempel

Ett [enkelt exempel](#). Testa att kakorna enbart sänds då man använder https i adressfältet på exemplet.

Tips

Se uppgiften [Sessionshantering: Lagra på klienten: Kakor](#).

E-post: Sändning: Utan bifogade filer

Allmänt

Uppgiften belyser hur man sänder e-post från ett HTML-formulär.

Uppgift

Gör ett HTML-formulär och ett HTTP-serversides-program som möjliggör sändning av e-post från en browser. Användaren ska kunna skriva in följande:

- Till
- Cc
- Bcc
- Från
- Ärende
- Meddelande
- Lösenord

När användaren har sänt brevet så ska en kopia av det som sändes visas som svar (se exemplet men det behöver inte se exakt likadant ut).

Lösenord används för att ingen ska kunna missbruka detta, det har tidigare hänt en del olyckor med just dessa uppgifter. Om användaren skriver in fel lösenord så ska användaren upplysas om detta och inget brev ska sändas. I slutet på alla meddelanden ska även följande information finnas:

Observera! Detta meddelande är sänt från ett formulär på Internet och avsändaren kan vara felaktig!

Notera att man **inte kan sända anonym e-post** eftersom det är mycket enkelt att spåra en användare som skriver in felaktiga avsändare i from-fältet.

Exempel

Ett [enkelt exempel](#). Lösenordet är 1234.

Tips

CGI/Perl

Man kan börja med att skapa ett brev på en fil men var då noga med att undvika krockar. Brevet har tre delar:

```
Mime-Version: 1.0
Content-Type: text/plain; charset=ISO-8859-1
Content-Transfer-Encoding: 8bit
To: x1@dsv.su.se
Cc: x2@dsv.su.se
Bcc: x3@dsv.su.se
From: y@dsv.su.se
Subject: Test
Date: Fri Jan 14 13:16:00 2004
```

```
Hejsan!
Hoppas Ni gillar kursen!
Mvh Pierre :-)
```

De olika delarna är:

- Ett mime-huvud med information om avsändare, mottagare med mera (egentligen räcker det med To, From och Date fälten)
- En tom rad mellan mime-huvudet och kroppen
- En kropp med meddelandet

För att sända brevet kan man använda Unix/Linux-programmet sendmail:

```
system('/usr/lib/sendmail -t -oi < message');
```

Här är message namnet på den fil som vi skrev brevet på.

Servlets

Använd `oreilly.servlet.MailMessage`.

E-post: Sändning: Med bifogade filer

Allmänt

Uppgiften belyser hur man sänder e-post med en bifogad godtycklig fil från ett HTML-formulär.

Uppgift

Utvidga uppgiften [E-post: Sändning: Utan bifogade filer](#) så att man även kan sända med minst två bifogade godtyckliga filer. Precis som innan så ska en kopia av det som sändes visas som svar när användaren har sänt brevet men för just de bifogade filerna kan man välja om man vill skriva ut

filerna eller bara filernas namn och storlek. Programmet ska fungera även om användaren inte väljer att sända med några filer alls.

Exempel

Ett [enkelt exempel](#). Lösenordet är 1234.

Tips

De bifogade filerna sänds från HTTP-klienten till HTTP-servern som i uppgiften [Input: Formulär: Uppladdning av fil](#).

CGI/Perl

Modulen MIME::Lite bör användas, se [MIME-Lite-1.135.readme](#), som installeras enligt följande:

- Skapa katalogen MIME i katalogen för uppgiften: `mkdir MIME`
- Lägg filen [Lite.pm](#) i denna katalog

När man väl satt sig in i denna modul blir programmet relativt lätt att göra. Notera att Content-Transfer-Encoding default är binary och att man bör sätta denna till base64.

Servlets

Använd Java-Mail API.

E-post: Mottagning: Utan bifogade filer

Allmänt

Uppgiften belyser hur man hämtar e-post via en HTML-sida.

Uppgift

Gör ett HTML-formulär och ett HTTP-serversides-program som låter en användare hämta all sin e-post. Användaren ska kunna skriva in följande från ett HTML-formulär:

- Användare
- Lösenord
- Mail-servers adress
- Mail-servers port

Exempel

Ett [enkelt exempel](#). Inga lösenord sparas givetvis men tänk på att de sänds okrypterade över Internet. Som svar anges här hela kommunikationen med POP-servern men det behöver man inte ta med.

Tips

CGI/Perl

Innan man börjar så kan man koppla sig "manuellt" till en POP-server för att se vilka kommandon man ska ge och vilka svar man får. Det gör man via telnet:

```
telnet mail.dsv.su.se 110
telnet mail.nada.kth.se 110
```

Här ska man använda Perl's socket funktioner och för denna uppgift ser koden för just detta ut så här:

```
use Socket;
socket SOCKET, AF_INET, SOCK_STREAM, 0;
connect(SOCKET, sockaddr_in($port, inet_aton($server)));
```

Man bör givetvis göra detta säkrare med `or die` och så vidare. Här är `$server` en sträng som anger den server som man anknyter till och `$port` är porten för tjänsten på denna server. Nu kan man skriva och läsa till `SOCKET` som till ett vanligt filhandtag.

Glöm inte att stänga av buffringen:

```
select SOCKET; $| = 1;
```

När man letar efter sista raden i ett brev (som avslutas med en ensam punkt) så är följande användbart:

```
last if $row =~ /\^\./;
```

eller säkrare:

```
last if $row =~ /\^\.\r\n$/;
```

Servlets

Använd Java-Mail API.

E-post: Mottagning: Med bifogade filer

Allmänt

Uppgiften belyser hur man sänder e-post med en bifogad godtycklig fil från ett HTML-formulär.

Uppgift

Utvidga uppgiften [E-post: Mottagning: Utan bifogade filer](#) så att man även kan hämta bifogade godtyckliga filer.

Exempel

Inget exempel ges.

Tips

CGI/Perl

Se modulen MimeTools.

Servlets

Inget tips ges.

Databaser: Lättvikts: DBM

Allmänt

Uppgiften belyser hur man använder en DBM-databas. Ofta behöver man inte använda en fullständig relations/SQL-databas (som kräver separat installation, konfiguration med mera) utan kan använda denna typ av mycket snabba lättviktsdatabas.

Uppgift

Gör ett HTTP-serversides-program som implementerar en enkel loggningsfunktion så att följande information sparas för alla som besöker sidan:

- Tid
- REMOTE_ADDR
- HTTP_USER_AGENT

Detta ska för alla besökare läggas in i en DBM-databas och alla loggar ska visas för besökaren.

Frivillig utökning är att även ta fram namnet som motsvarar REMOTE_ADDR.

Exempel

Ett [enkelt exempel](#). Ett annat exempel är kursens webbplats.

Tips

CGI/Perl

Använd modulen DB_File.

Servlets

Man behöver här ett tillbehör som implementerar DBM, se exempelvis »[Welcome to the JDBM project](#)« som redan finns i den för kursen konfigurerade Tomcat-servern.

Databaser: Relations/SQL: I / O

Allmänt

Uppgiften belyser hur man använder en SQL-databas från ett HTML-formulär. Detta är en mycket vanlig teknik som det är viktigt att man kan bra:

- Ett HTML-formulär i en browser anropar:
 - En HTTP-server som anropar:
 - Ett HTTP-serversides-program som anropar:
 - En SQL-server, som sedan svarar:
 - HTTP-serversides-programmet, som svarar:
 - HTTP-servern, som svarar:
 - Klientens browser

I den här uppgiften är det databasservern MySQL som anropas.

Uppgift

Gör en enkel gästbok. Varje gäst ska i ett formulär på en HTML-sida kunna skriva in (minst):

- Namn
- E-post adress
- Hemsida
- Kommentar

Detta ska läggas in i en SQL-databas där varje inlägg ska tidsstämplas och sedan presenteras enligt samma format som i exemplet. För säkerheten så ska all insänd data i formuläret kontrolleras så att all eventuell:

- HTML-kod ska substitueras bort
- SQL-kod inte ska kunna köras (SQL-injection)

När inläggen visas så ska varje gästs e-post adress och hemsida vara "klickbara" (se exemplet).

Man måste inte (som i exemplet) göra allt i ett enda HTML-dokument utan man kan använda två olika HTML-dokument:

- Ett HTML-dokument för att lägga in inlägg
- Ett HTML-dokument för att visa inlagda inlägg

Om någonting går fel så ska information om felet skrivas ut till användaren.

Lägg gärna till sortering efter namn och tid eller andra funktioner, men detta är frivilligt.

Exempel

Ett [enkelt exempel](#).

Tips

För att kunna lösa denna uppgift måste man använda en databas, se [MySQL](#). Om man redan har gjort det på Internetprogrammering I så kan man använda samma databas och samma tabell guestbook i denna uppgift.

Använder man DSV's MySQL-databasserver så kör den på default-porten och man behöver då inte ange portnummret när man ansluter.

CGI/Perl

För de som arbetar på DSV så måste dessa uppgifter testas med browsern eftersom DBD/DBI-modulerna enbart är installerade på atlas.dsv.su.se. De som arbetar hemma måste installera DBD/DBI-modulerna.

Anslutningen görs så här om databasservern kör på samma dator som HTTP-servern:

```
$dbh = DBI->connect("DBI:mysql:$db", $user, $passwd) or die print "COULD NOT CONNECT: " . DBI->errstr;
```

Annars:

```
$dbh = DBI->connect("DBI:mysql:$db:<dator adress>", $user, $passwd) or die print "COULD NOT CONNECT: "
```

Man kan inte skriva en variabel direkt i en SQL-fråga utan variabelns värde måste omges med citattecken:

```
$name = '\'' . param('name') . '\'';
```

Det är bättre att använda funktionen `DBI::quote` men för att undvika SQL-injection så är det dock bäst att använda funktionen `bind_param`:

```
...
$statement = "INSERT INTO $table (r1, r2, r3, r4, r5) VALUES (?, ?, ?, ?, ?)";
$sth = $dbh->prepare($statement) or die print "COULD NOT PREPARE STATEMENT: " . $dbh->errstr;
$sth->bind_param(1, $x1);
$sth->bind_param(2, $x2);
$sth->bind_param(3, $x3);
$sth->bind_param(4, $x4);
$sth->bind_param(5, $x5);
...
```

Fälten får man enligt följande:

```
@fields = @{$sth->{NAME}};
```

Vill man använda andra databaser är MySQL så räcker det med att ändra texten `mysql` till exempelvis `oracle` på ett enda ställe i programmet samt installera de DBD/DBI-moduler som hör till Oracle. Man kan även använda flera olika typer av databaser samtidigt.

Servlets

Man behöver här ett JDBC-gränssnitt mot MySQL. JAR-filen [mm.mysql-2.0.14-bin.jar](#) finns redan i den för kursen konfigurerade Tomcat-servern.

Anslutningen görs med:

```
Class.forName("org.gjt.mm.mysql.Driver").newInstance();
Connection dbConnection = DriverManager.getConnection(url, username, password);
```

där:

```
o url = "jdbc:mysql://" + computer + "/" + db_name;
```

där `computer` är `localhost` om HTTP-servern kör på samma dator som MySQL-servern, annars datorns adress.

För att undvika SQL-injection är `PreparedStatement` bra.

Databaser: Relations/SQL: Transaktioner

Allmänt

Uppgiften belyser hur man använder en SQL-databas med transaktioner.

Uppgift

Bygg ut uppgiften [Databaser: Relations/SQL: I/O](#) så att gästboken fungerar med bilder:

- Lägg till ett fält för uppladdning av en bild-fil i HTML-formuläret
- Gör två tabeller i databasen:
 - En tabell (som tidigare) för namn, e-post adress, hemsida och kommentar
 - En tabell för bilden och bildens mime-typ
- Använd transaktionshantering när inläggen ska sparas i databasen:
 - Anropen på de båda tabellerna ska grupperas så att båda anropen blir en atomär enhet
 - Om inte båda anropen går igenom så ska en rollback-operation utföras och HTTP-klienten ska informeras om det

Exempel

Ett [enkelt exempel](#).

Tips

För att transaktioner ska fungera på MySQL bör man använda MySQL version 4.0 eller senare. Man måste köra MySQL med lagringsmotorn/tabelltypen InnoDB.

För att skapa en tabell av typen InnoDB ska man skriva:

```
CREATE TABLE IF NOT EXISTS table_1 ( ... ) TYPE=InnoDB;
```

Har man redan en tabell av en annan typ kan man ändra den enligt:

```
ALTER TABLE table_1 TYPE=INNODB;
```

Se vidare »[Dev Shed: Using Transactions In MySQL \(Part 1\)](#)« och »[Dev Shed: Using Transactions In MySQL \(Part 2\)](#)«.

För att lagra bilder kan man använda SQL-typen MEDIUMBLOB.

CGI/Perl

Man måste stänga av autocommit och det kan man göra så här:

```
$dbh = DBI->connect("DBI:mysql:$db", $user, $passwd, {RaiseError => 1, AutoCommit => 0});
```

För transaktionshanteringen kan man använda eval-block runt stället där man lägger in data i databasen så att fel kan fångas och man då kan göra rollback:

```
eval {
```

```
...
};

$dbh->commit();

if ($@) {
    print "Transaction aborted: $@";
    $dbh->rollback();
}
```

Bilderna kan läsas in så här:

```
if(param('file')) {
    $file_handle = param('file');
    $mime_type = uploadInfo($file_handle)->{'Content-Type'};

    # Read in the contents
    while (<$file_handle>) {
        $image_data .= $_;
    }
} else {
    $mime_type = "none";
    $image_data = "";
}
```

Servlets

För transaktionshanteringen kan man använda:

```
dbConnection.setAutoCommit(false);
...
dbConnection.commit();
```

Om något fel uppstått (i ...) så anropar man:

```
dbConnection.rollback();
```

För bilderna kan man använda något i stil med:

```
preparedStatement2 = dbConnection.prepareStatement("INSERT INTO GuestBookImage " + "VALUES (?, ?, ?, ?");
FileInputStream fis = new FileInputStream(file);
byte[] data=new byte[(int) file.length()];
fis.read(data);
fis.close();
preparedStatement2.setBytes(1, data);
```

Databaser: Relations / SQL: Effektivitet

Allmänt

Uppgiften belyser hur man använder en teknik kallad connection pooling för att uppnå större snabbhet när man ansluter till en SQL-databas. Detta gör man för att anslutningen till en databas brukar ta tid och en lösning till det är att låta ett HTTP-serversides-program låna ett redan anslutet objekt från en pool av sådana objekt.

Uppgift

Bygg ut uppgiften [Databaser: Relations/SQL: I/O](#) eller uppgiften [Databaser: Relations/SQL: Transaktioner](#) så att gästboken fungerar med connection pooling.

Exempel

Inget exempel ges.

Tips

Inget tips ges.

Innehållshantering: Manuellt innehåll: Publiceringssystem

Allmänt

Uppgiften belyser hur man konstruerar ett enkelt publiceringssystem för smidig manuellt uppdatering av innehåll.

Uppgift

Gör ett program som implementerar en BLOG som använder ett minimalt egendefinerat WIKI-baserat språk för att låta en användare lägga till information (exempelvis text och bilder) i en HTML-sida direkt från en browser.

Frivillig överkurs är att göra BLOG'en användbar även från mobila klienter med exempelvis text, bild, ljud och film.

Exempel

Kurserna i Internetprogrammering och Internetsäkerhet använder ett sådant system.

Tips

Titta på uppbyggnaden och språkreglerna hos några olika WIKI och hämta inspiration från dem.

Innehållshantering: Manuellt innehåll: Syndikering

Allmänt

Uppgiften belyser hur man använder automatisk syndikering för att meta-beskriva information dynamiskt.

Uppgift

Utvidga uppgiften [Innehållshantering: Manuellt innehåll: Publiceringssystem](#) så att man automatiskt får en meta-beskrivning av varje inlägg i BLOG'en enligt någon av följande kanalstandarder:

- Really Simple Syndication (RSS 0.91 och 0.92)
- RDF Site Summary (RSS 1.0)

Exempel

Inget exempel ges.

Tips

Inget tips ges.

Innehållshantering: Automatiskt innehåll: Kanaler

Allmänt

Uppgiften belyser hur man skapar automatiskt innehåll från en mängd ordnade meta-beskrivna källor.

Uppgift

Gör ett program som dynamiskt (on demand) skapar en HTML-sida med innehåll genom att hämta visst innehåll från ett antal på förhand specificerade meta-beskrivna HTML-sidor. Välj själv vilken kanalstandard som programmet ska fungera med, exempelvis:

- Really Simple Syndication (RSS 0.91 och 0.92)
- RDF Site Summary (RSS 1.0)

Frivillig överkurs är att implementera så att användaren på ett enkelt sätt även kan lägga till och ta bort de meta-beskrivna källor som ska användas av programmet.

Exempel

Se exempelvis »[Google News](#)«.

Tips

Inga tips ges.

Innehållshantering: Automatiskt innehåll: Fritextsökning

Allmänt

Uppgiften belyser hur man skapar automatiskt innehåll från en mängd oordnade länkade källor.

Uppgift

Gör ett program som dynamiskt (on demand) skapar en HTML-sida med innehåll genom att hämta visst innehåll från ett antal på förhand specificerade HTML-sidor och de HTML-sidor som dessa sidor länkar till.

Välj själv vilka specifika rekursiva sökalgoritmer (djupet-först, bredden-först och så vidare) och sökkriterier som ska användas.

Denna uppgift är mycket lik en sökspindel och man kan om man vill fokusera på en implementation av en sådan. Frivillig överkurs är att bygga ut uppgiften till en komplett sökmotor genom att använda flera samtidiga sök-spindel-processer som gemensamt uppdaterar en databas.

Exempel

Ett [enkelt exempel](#) på just en sökspindel. Detta exempel letar efter en angivet ord med start på en given URL och fortsätter därefter rekursivt på alla länkar som hittas i dokumentet (till den nivå som angivits).

Tips

Observera följande hänsyn vid uttestningen:

- Sök inte på sidor från DSV utan sök på andra och olika ställen
- De som testkör sitt program på DSV bör begränsa sökningen till 2 - 3 nivåer eftersom antalet sidor växer mycket snabbt

CGI/Perl

CPAN-modulerna LWP::Simple och HTML::LinkExtor är användbara. Dessa moduler och flera andra finns zippade i filen [ext.zip](#). Hämta den och zippa upp den i samma katalog som dessa moduler skall användas i. För att sedan kunna nå modulerna skriver man följande i Perl-programmen:

```
use lib "ext";
use LWP::Simple;
use HTML::LinkExtor;
```

Bygg sökspindeln stegvis:

- Börja med att göra ett HTTP-serversides-program som bara hämtar en annan sida, exempelvis som detta [enkla exempel](#)
- Bygg sedan ut HTTP-serversides-programmet så det kan hitta, exempelvis som detta [enkla exempel](#):
 - Om ett angivet ord finns på sidan
 - Alla länkar på sidan

En användbar subrutin:

```
sub get_links {
    $parser = HTML::LinkExtor->new(undef, $url);
    $parser->parse($html)->eof;
    @links = $parser->links;
    foreach $linkarray (@links) {
        @element = @$linkarray;
        $elt_type = shift @element;
        while(@element) {
            ($attr_name, $attr_value) = splice(@element, 0, 2);
            $seen{$attr_value}++;
        }
    }
}
```

```
}  
for(sort keys %seen) {  
    @ret = (@ret, $_);  
}  
return @ret;  
}
```

Gör man en riktig sökspindel som hämtar information från många ställen på Internet bör man använda `LWP::RobotUA` istället för `LWP::UserAgent` eftersom denna modul tar hänsyn till olika webbplatserns önskemål (vilka länkar man ska följa) och även lägger in en fördröjning mellan förfrågningarna så att man inte sänker någon HTTP-server med för intensiva förfrågningar.

Servlets

Inget tips ges.

Gesällprov

Allmänt

Gesällprovet är en uppgift för den egna kreativiteten och är ett fritt valt (men obligatoriskt) arbete där du får chansen att visa vad du går för konstruktionsmässigt (istället för minnesmässigt som på en tenta) i "lugn och ro".

Uppgift

Följande krav måste uppfyllas:

- Ska baseras på de tekniker som kursen bygger på men får givetvis även innehålla tekniker från andra områden
- Ska vara nätverksbaserat
- Ska vara bra både till både form och funktion
- Ska inte vara en av de tidigare uppgifterna på kursen men gärna en utökning av en eller flera av dessa

Tips

Huvudsaken är att man gör något som är kul! Några exempel:

- Inloggningssystem som använder HTTPS/SSL, säkra kakor, sessions-ID och sessionshantering
- Kompletta SQL-genererade e-affär med:
 - Ingång för kund med sessionshantering, shoppingvagn, kassa, e-post-lista, kaka för automatisk identifiering med mera
 - Ingång för innehavare med lager, kunder, omsättning, vinst med mera
- Sökmotor och databas:
 - Lokal filsystems-sökmotor
 - SQL-baserad sökmotor för Internet
- News:
 - Webbaserat diskussionsforum för Internetprogrammeringskurserna
- Spel:
 - Ett roligt nätverksbaserat spel
 - File upload av två bilder och svara med en (på servern) morfad moving-GIF av dessa

Hjälp

På denna sida finns alla hjälpresurser på kursen.

Handledning

Kör man fast så får man hjälp via news och e-post. Notera dock att det inte är någon handledning på:

- Kunskap som ingår i förkunskapskraven (kompileringsfel med mera)
- Gesällprovet

News är tänkt som en plattform för diskussion mellan studenterna men handledarna kommer att besvara de frågor som fortfarande är obesvarade senare under de dagar som det är handledning. E-post går direkt till handledarna och de kommer att försöka svara snabbt men eftersom de ibland får väldigt många frågor kan svaren dröja. Notera att handledarna enbart arbetar inom angivna handledningstider vilket medför att man kanske inte hinner få svar samma dag på en sent ställd fråga.

News

News bör användas för frågor som kan intressera andra:

- Läs först igenom de tidigare frågorna och svaren
- Hittar man inte en lösning så:
 - Ange ett väl beskrivande ämne för brevet
 - Var mycket tydlig vid beskrivningen av problemet
 - Sänd inte in mycket kod eller fullständiga program

News-server:

- Adress: `atlas.dsv.su.se`
- Port: 1199 (notera att detta inte är default numret och därför måste sättas manuellt)

News-grupper:

- `public.ip2.a.1` för frågor om uppgift 1
- `public.ip2.a.2` för frågor om uppgift 2
- ... och så vidare ...
- `public.ip2.misc` för övriga frågor

Se även [news](#) för mer detaljerad information om hur man konfigurerar en news-klient.

E-post

E-post bör användas för specifika frågor utan intresse för andra:

- Sänd inte e-post direkt till lärarna utan använd länken nedan
- Om man ska sända med kod:
 - Sänd endast med de delar som är relevanta för frågan
 - Koden ska vara välkommenterad och välstrukturerad
- Använd "svara till alla" på eventuella följdfrågor

E-post-server:

- Ingående e-brev (POP3):
 - Adress: `mail.dsv.su.se`
 - Port: 110
- Utgående e-brev (SMTP):

- Adress: mail.dsv.su.se
- Port: 25

Notera att om man sitter uppkopplad utanför DSV's domän (exempelvis via Telia Bredband) så ska man använda leverantörers SMTP-server för utgående brev.

E-post länk:

- [Klicka här för handledning via e-post](#)

Se även [e-post](#) för mer detaljerad information om hur man konfigurerar en e-post-klient.

FAQ

Här följer de frågor och svar som brukar vara vanligast:

HTTP-serversidan

Fråga/svar 1

Man kan använda två HTTP-metoder, GET och POST. Vilken av dessa två ska man använda?

GET:

- Anrop via GET syns i URL-adressfältet, man kan exempelvis skapa en egen länk till Google med en viss specifik förfrågan direkt till sökmotorn
- Informationen sänds i HTTP-huvudet via omgivningsvariabeln QUERY_STRING (GET sänder bara ett HTTP-huvud)
- En URL-adress får enligt standarden vara hur lång som helst men olika webb-läsare stöder ofta inte mer än 1024 tecken så mängden information som kan sändas är begränsad

POST:

- Anrop via POST syns inte i URL-adressfältet
- Informationen sänds i HTTP-kroppen (POST sänder både ett HTTP-huvud och en HTTP-kropp)
- Mängden information som kan sändas är godtyckligt stor

Ursprungligen var HTTP-metoden GET tänkt till att hämta dokument och POST att sända dokument.

Fråga/svar 2

Hur förhindrar man att browsern använder cachad information?

Man infogar följande i HTTP-huvudet:

Dokumenthuvudsrubrik	Kommentar
Expires: Mon, 26 Jul 1997 05:00:00 GMT	Gammalt datum
Last-Modified: x	Modifiera x för varje access
Cache-Control: no-cache, must-revalidate	För HTTP/1.1
Pragma: no-cache	För HTTP/1.0

Fråga/svar 3

Hur skickar man text till en fax från ett HTML-formulär på nätet?

Man kan använda en fax-gateway, exempelvis »[SUNET: Faxförmedling](#)«.

Ett [enkelt exempel](#).

CGI/Perl

Fråga/svar 1

När jag ska skriva ut en enkel text så fungerar det inte alls?

50% av frågorna brukar komma i samband med att man skriver ut enligt:

```
print <<_TEXT_
... osv ...
_TEXT_
```

Det är lätt att få ett mellanslag eller ^M sist efter strängen _TEXT_ vilket det inte ska vara. Använd inte detta sätt att skriva ut utan använd hellre en print-sats för varje sak som ska skrivas ut.

Fråga/svar 2

Varför fungerar mitt program på Win32 men inte på Unix/Linux?

När man gör textfiler i Win32 så får man ibland osynliga tecken (Carriage Return, CR) i textfilen som gör att koden inte kan köras på Unix. Man tar bort alla CR-tecken från en fil med:

```
tr -d \r <filnamn >nytt_filnamn
```

Observera att filnamn och nytt_filnamn måste vara olika, annars försvinner all information.

En annan vanlig felkälla är när man använder FTP. När man för över CGI-filer så ska man använda ASCII-transfer-mode (inte binary-transfer-mode).

Fråga/svar 3

Hur tar man reda på vilka moduler som är installerade på en dator?

Enligt CPAN kan man använda sig av »[SUNET, CPAN Frequently Asked Questions: perldoc perllocal](#)«. Se även »[SUNET, CPAN Frequently Asked Questions: perldoc perlmodlib](#)«.

E-post

På denna sida beskrivs hur man konfigurerar e-post-klienten. Följande är ett exempel på hur man konfigurerar e-post i Outlook Express, förfarandet är ungefär detsamma i andra moderna e-post/news-klienter:

- Klicka på Verktyg/Konton
- Klicka på Add/Mail
- Skriv in namn och e-post-adress
- Skriv in adressen för inkommande och utgående e-post-servrar
- Skriv in användarnamn och lösenord
- Klicka på Avsluta

Vill man använda exempelvis hotmail istället så kan man göra följande för att få all e-post som kommer till DSV vidareänd till hotmail-kontot:

- telnet triton.dsv.su.se
- emacs .forward
- Skriv in hotmail-adressen, spara och avsluta emacs
- Logga ut

De som använder ett e-postkonto på DSV kan även använda »[DSV: Webmail](#)«.

News

På denna sida beskrivs hur man konfigurerar en news-klient. Följande är ett exempel på hur man konfigurerar news i Outlook Express, förfarandet är ungefär detsamma i andra moderna e-post/news-klienter:

- Klicka på Verktyg/Konton
- Klicka på Lägg till/News
- Skriv in namn och e-post-adress
- Ange adressen för news-servern
- Klicka på Avsluta
- Klicka på mappen Alla och markera aktuell news-server
- Klicka på Egenskaper
- Klicka på Avancerat
- Ange port nummer till news-servern
- Klicka på ok och sen Stäng
- Svara Ja på frågan Vill du ladda ner nyhets-grupper från news-kontot
- Dubbelklicka på alla grupper som är av intresse
- Klicka på ok

Default så visar Outlook Express bara de senaste meddelandena. Vill man ändra på det så:

- Klicka på Verktyg/Alternativ
- Klicka på mappen Underhåll
- Ändra vid Ta bort news meddelanden efter 5 dagar (exempelvis genom att ta bort markeringen helt)

Anledningen till att vi använder news istället för FirstClass är att news är en öppen mycket vanlig standard på Internet och att det även ingår som några förslag på gesällprov.

Redovisning i detalj

Denna beskrivning utgår från att man använder DSV's datorer men man kan använda andra ställen än DSV för redovisningen. Notera dock att andra ställen kan använda en annan huvudkatalog för webpublicering och andra rättigheter på kataloger och filer.

Steg för steg

1. Starta ett [filöverföringsprogram](#) och koppla dig till ditt DSV/Unix/Linux-konto
2. Skapa katalogstrukturen nedan och ladda upp filerna i denna struktur
3. Sätt rättigheterna på kataloger och filer så att allt blir läsbart från en browser
4. Testa att allt kan nås från en browser och att all källkod ser bra ut från en browser
5. Ange på redovisningssidan:
 - URL'en till denna katalogstruktur: `http://www.dsv.su.se/~x/ip2/`, där x är användarnamnet
 - Att allt är klart för redovisning

Katalogstruktur

Man måste ha **exakt följande katalogstruktur** men använder man en annan plats än DSV så måste inte www katalogen ha det namnet.

```
www
  ip2
    ip2.zip (ip2 zippad, ej innehållande ip2.zip givetvis)
    1.1.1
      källkodsfiler för uppgift 1.1.1
    1.1.2
      källkodsfiler för uppgift 1.1.2
    1.2
      källkodsfiler för uppgift 1.2
    1.3.1
      källkodsfiler för uppgift 1.3.1

    ... osv ...

  8
    källkodsfiler för gesällprovet
    description
      beskrivning i text och bild av gesällprovet
    docs (för de som valt Servlets)
      javadoc-beskrivning
```

Observera:

- Katalogen www måste ha stora/versala bokstäver
- Skapa inte kataloger för frivilliga uppgifter som inte är gjorda
- Lägg ingen fil med namnet index.html eller index.htm direkt i uppgiftskatalogerna eftersom man då inte kan lista innehållet i dessa kataloger från Internet
- Källkoden för HTTP-serversides-programmen måste vara läsbara från en browser och för vissa tekniker, som exempelvis CGI, ASP.NET, JSP och PHP, innebär det att man måste kopiera filerna för HTTP-serversides-programmen till TXT-filer

Beskrivning av gesällprovet

Gesällprovet ska beskrivas generellt med både text och bild (på programmet under körning) på något av följande sätt:

- HTML-dokument med bilderna inkluderade
- PDF-dokument med bilderna inkluderade
- TXT-dokument med bilderna separata

Ett HTML-dokument kan göras antingen för hand i en vanlig text-editor eller med ett högnivåverktyg.

För att skapa PDF-filer i Win32 kan man exempelvis:

1. Lägg till en skrivare, exempelvis HP Color LaserJet PS
2. Högerklicka på skrivaren och välj Egenskaper, klicka på Portar och kryssa i FILE
3. Skriv ut Word-dokumentet till denna skrivare varefter en fil <namn>.prn skapas
4. Gå till »PS2PDF.com« och konvertera filen till PDF

En bild (använd inte BMP-format) av ett program under körning fås enligt:

- Win32: Tryck på <CTRL><ALT><PRINT SCR> samtidigt, vilket medför att bilden på programmet läggs i urklippsbufferten som sedan lätt kan klistras in i ett HTML-dokument i en HTML-editor
- Linux/Unix: Skriv i kommandofönstret `import dump.jpeg` och klicka sen på det fönster som man vill ha en bild på (ändrar man till en annan extension så får man bilden som en annan bildtyp)

All eventuell java-kod i gesällprovet ska beskrivas med verktyget javadoc.

Efter erhållet betyg

När man fått G/VG eller 3/4/5 på ett arbete så bör man ändra rättigheterna på källkodsfilerna så att ingen annan kan läsa dem.

Nyheter

På denna sida visas alla uppdateringar.

Tid	Rum	Uppdatering
2004-10-14 12:05	Allt	Nya kursen klar
2004-11-02 12:00	Tomcat	Tips för HTTPS/SSL
2004-11-02 13:45	Uppgifter	Uppgift 2.2.3 mer tips
2004-11-03 13:00	Uppgifter	Uppgift 2.2.3 finslipad
2004-11-24 03:05	Uppgifter	Uppgift 6.2.2 mer tips + ett exempel

Internetprogrammering III

På denna sida finns hela kursen utom utom OH-bilderna och exemplen.

Intro

Välkommen Pierre! På denna sida beskrivs kursen generellt.

Kurskoder

Kursen är på 5 poäng och går för följande universitet/högskolor:

- SU med kod IP3
- KTH med kod 2I1803
- KTH-FoV med kod 2I4112
- Nätuniversitetet med kod IB9040

Kursen ges även i en utökad fördjupande form, kallad IP3+, på 5 + 2 poäng för följande universitet/högskolor:

- SU med kod IP3+

Notera att man **inte senare kan komplettera** från IP3 till IP3+ utan att man måste välja IP3+ direkt från start.

Innehåll

Denna kurs lär ut Internetprogrammering fördjupat mot HTTP-klient-sidan:

- Struktur med Extended Hypertext Markup Language (XHTML):
 - XHTML 1.1 för stationära enheter

- XHTML Basic för mobila enheter (1/2 i WML 2.0)
- Modularization of XHTML för egendefinerade XHTML-standarder
- XHTML och teckenuppsättningar
- XHTML 2.0
- Riktlinjer för tillgänglighet (accessability)
- Stilsättning med Cascading Style Sheets (CSS):
 - CSS 2.1 för stationära enheter
 - CSS Mobile Profile för mobila enheter (2/2 i WML 2.0)
 - Dynamiskt anpassning
- Innehållshantering:
 - Separation av data och XHTML med Extensible Markup Language (XML), Extensible Stylesheet Language Transformations (XSLT) och XML Path Language (XPath)
 - Modularisering av data
 - Syndikering med Really Simple Syndication (RSS 0.91 och 0.92) och RDF Site Summary (RSS 1.0)
- Bild:
 - Bitmappad:
 - PNG/JPEG/p-JPEG/GIF/Moving-GIF/WBMP
 - Java-baserad 2D
 - Vektorbaserad med Scalable Vector Graphics (SVG) och Java 3D:
 - SVG 1.1 för stationära enheter
 - SVG 1.1 för stationära enheter uppspelad med Java
 - SVG Tiny och SVG Basic för mobila enheter
 - Java-baserad 3D
- Audio och video:
 - Inspelning/redigering/konvertering:
 - Audio med MP3 och Ogg Vorbis
 - Video med Ogg Theora
 - Strömmande via RTSP/RTP/ISMA:
 - Audio med MP3 spellistor
 - Audio med MP3 spellistor uppspelad med Java
 - Video med MPEG-4 för stationära enheter
 - Video med MPEG-4 för stationära enheter uppspelad med Java
 - Video med 3GPP för mobila enheter
 - Semi-strömmande via HTTP-interleaving/muxing:
 - Video med RealMedia uppspelad med RealPlayer
 - Video med MPEG-4-MUX för stationära enheter uppspelad med Java
 - Integrering med Synchronized Multimedia Integration Language (SMIL):
 - SMIL 2.0 för stationära enheter
 - SMIL 2.0 för stationära enheter uppspelad med Java
 - SMIL i MMS för mobila enheter
- Dynamisk HTML (DHTML) med JavaScript 1.5 och Object Model (OM) där Document Object Model 2 (DOM 2) ingår:
 - Objekt-modellen:
 - Egenskaper
 - Händelser
 - Metoder
 - Dokument-objekt-modellen:
 - Dynamisk struktur
 - Dynamisk stil
 - Dynamiskt innehåll
 - Formulär validering
 - JavaScript-Java kopplingar
 - Databindning

Språk: XHTML, CSS, SVG, SMIL, JavaScript, DOM, XML, XPath, XSL, XSLT, Java. Kursen följer W3C's standarder.

Undervisningsform

Pedagogiken för denna kurs är:

- (1) Frihet, (2) tävling och (3) samarbete --> glädje och motivation --> ett bra resultat!

Frihet, tävling och samarbete gör att man går från passivt mottagande till aktivt sökande. Genom frihet tillåts man själv bli nyfiken och ställa frågor innan man får svar. Genom tävling motiveras man att göra sitt bästa. Detta leder fram till glädje och motivation och därmed till ett bra resultat.

Observera! Detta är en distanskurs och kräver förmåga till självständigt arbete. Kursen har dock, till skillnad mot vanliga distanskurser, kontinuerligt och direkt stöd via handledning på distans. Kursen är krävande och man bör inte läsa andra kurser samtidigt. Kursen fokuserar starkt på praktiskt arbete och examinationen bygger på detta arbete. Digitalslörd!

Examination

Examinationen på kursen består av:

- Praktiska uppgifter, 4 poäng (R/G/VG) (R/3/4/5):
 - Flera bestämda/styrda
 - Ett fritt valt gesällprov
- Tentamen, 1 poäng (U/G) (U/3)

För fördjupningskursen gäller utöver ovanstående även:

- Extra praktiska uppgifter, 2 poäng (R/G) (R/3):
 - Flera bestämda/styrda

Totalbetyget på båda kurserna bestäms av de praktiska uppgifterna med tyngdpunkt på det fritt valda gesällprovet.

Notera att man måste lämna in lösningar på de praktiska uppgifterna som stämmer med **nuvarande läsårs praktiska uppgifter**. Kurserna utvecklas dock bara i små steg med gradvisa förfiningar så det innebär inte att man måste göra om allt om man inte har fått betyg på de praktiska uppgifterna vid den sista deadline i Augusti.

Litteratur

Litteraturen är:

- »[Using HTML and XHTML \(speciell utgåva\)](#)« skriven av Holzschlag och utgiven av »[QUE](#)«

Notera att man istället kan använda:

- »[W3 WSchools](#)« tutorials
- »[W3C](#)«'s specifikationer
- »[Adobe: SVG Zon: Tutorials](#)«
- »[Real: RealNetworks Production Guide](#)«
- Audio och video länkar under sidan Externa källor

För de som går IP3+ även:

- »[JavaScript The Definitive Guide \(fjärde utgåvan\)](#)« skriven av Flanagan och utgiven av »[O'Reilly](#)«

Billigaste boken hittas med hjälp av »[Bokfynd.nu](#)«. Vi rekommenderar »[Amazon](#)« och »[WHSmith](#)« i England som brukar ha bra priser och snabba leveranser. Köp inte böcker utanför EU eftersom det tar lång tid (2-3 veckor) att få böckerna och sen får man en ganska stor extra avgift från tullen.

4 steg för att klara denna kurs

Gör följande för att klara denna kurs:

1. Registrera dig hos »[DSV: Studievägledningen](#)« som har e-post svl@dsv.su.se
2. Köp och läs kurslitteraturen
3. Lös och redovisa uppgifterna
4. Anmäl dig till tentamen (minst en vecka innan den går) och klara denna

Lärare

Vi önskar alla en givande och rolig kurs!

- »[Pierre A. I. Wijkman](#)«, ansvarig
- Handledarna

Lycka till!

Schema

På denna sida finns alla tider och platser för kursen. Alla tider är preliminära och kan ändras.

Föreläsningar

Detta är en distanskurs och vi ger inte en hel föreläsningsserie. Den inledande föreläsningen fungerar enbart som en kick-off eller frösättning. Notera att innehåll markerat som reserv innebär att ingen föreläsning ges men att detta kan ändras vid behov.

Tid	Rum	Föreläsare	Innehåll
2005-01-17 kl 15-19	F3, Electrum, Kista	Pierre A. I. Wijkman	Intro, XHTML, CSS, DHTML med JavaScript/OM

Handledning

All handledning sker via e-post och news (se sidan Hjälp).

Samtliga studenter har handledning:

- Kursens 1:a - 5:e vecka: samtliga vardagar kl 10-17

Studenter som läser +-kursen (7 poäng) har även handledning på +-delen:

- Kursens 6:e - 7:e vecka: samtliga vardagar kl 10-17

Studenter från Nätuniversitetet som läser på halvfart har även handledning:

- Kursens 6:a - 10:e vecka: samtliga vardagar kl 10-17

För de som arbetar på DSV finns info om prio-tider för labbsalarna under »[DSV: Scheman](#)«.

Examination

Notera följande angående tid för rättning:

- Det brukar ta upp till **3 veckor tills rättningen är klar**
- Ansvarig kan **inte rätta vissa personers uppgifter före andras**, exempelvis de som har bråttom på grund av CSN
- När man fått betyg på kursen så dröjer det **ytterligare tid** tills allt är infört i Daisy och Ladok

Var god **e-posta inte ansvarig om detta** för att försöka påskynda processen, det är väldigt många studenter på samtliga kurser.

Man får välja tid för tenta och deadline oberoende av om man går på SU eller KTH och så vidare. Typen nedan är enbart en rekommendation.

Deadlines för uppgifterna

Det finns flera deadlines för inlämning av uppgifterna och man väljer själv en som passar. Man redovisar samtliga uppgifter direkt eftersom det av praktiska skäl tyvärr inte går att delrätta och ge återkoppling efter hand som deluppgifterna blir klara. De som går en + kurs kan dock lämna in ordinarie uppgifter och + uppgifter separat vid olika deadlines. Varje deadline är normalt daterad till dagen innan tentamen.

Tid	Rum	Typ
2005-02-21 kl 24	Internet	Deadline 1 SU kurs IP3
2005-03-07 kl 24	Internet	Deadline 1 KTH och SU kurs IP3+
2005-05-20 kl 24	Internet	Deadline 2
2005-08-10 kl 24	Internet	Sommardeadline

Observera:

- Redovisade uppgifter **rättas endast i samband med dessa deadlines och uppskov ges aldrig**
- Får man **rest** så **rättas den igen först vid nästa deadline**
- Betyget påverkas inte beroende på vilken deadline man väljer
- Man måste inte välja en deadline som ligger i anslutningen till en tenta
- Handledningen slutar ofta innan deadline 1 (det finns tyvärr inga resurser för en längre handledningsperiod)

Tentamen

Inga hjälpmedel är tillåtna på tentamen. Man bör ha gjort klart de obligatoriska uppgifterna för 5 poängskursen (exklusive gesällprovet) innan tentamen eftersom tentamen kommer att fråga om dessa (detta är dock inget strikt krav för att få gå upp på tentamen). Vissa frågor kan vara mer generella och vissa kan vara mer tekniska och fråga om hur viss kodning (med avseende på uppgifterna) görs.

Tid	Rum	Typ
2005-02-22 kl 16-20	Forum, Kista	Tentamen SU
2005-03-08 kl 16-20	Forum, Kista	Tentamen KTH
2005-05-21 kl 10-14	Forum, Kista	Omtentamen
2005-08-11 kl 09-13	Forum, Kista	Sommartentamen

Observera:

- Man måste anmäla sig i Daisy **senast en vecka innan tentamen går**, se »[DSV: Att tentera vid DSV](#)«
- Man måste inte välja en tenta som ligger i anslutningen till en deadline

- Sal meddelas i Daisy eller på telefonsvarare 08 674 70 04 senast en dag innan tentamen, se även »[DSV: Scheman](#)«

Var god **e-posta inte ansvarige** (han har inga befogenheter här) **om detta**, utan kontakta istället »[DSV: Studentexpeditionen](#)«.

Kursmaterial

På denna sida finns information om arbetssätt och vilka mjukvaror man behöver. Här finns även OH-bilder och tidigare tentor.

Arbetssätt och mjukvaror

På denna kurs kan man arbeta på följande sätt:

- På DSV:
 - [Darwin](#)
- På distans online kopplad via telnet eller SSH till sitt DSV/Unix-konto:
 - [Terminalprogram](#)
 - [Darwin](#)
- På distans offline med det egna systemet konfigurerat för allt som behövs:
 - [Editor](#)
 - Browser: »[Mozilla Firefox 1.0](#)« med plugin »[Adobes SVG plug-in för Firefox](#)«
 - Bildbehandling:
 - »[Gimp 2.0.6 med stöd för GIF-bilder](#)«
 - [GIFAnimator](#)
 - Audio-redigering: »[Audacity](#)«
 - Video-redigering:
 - Unix/Linux: »[Kino](#)«
 - Win32: »[VirtualDub](#)«
 - Audio/video-konvertering:
 - »[Real Producer Basic 10](#)«
 - »[Helix Producer 10](#)« med plugin »[Xiph Producer Plugin](#)«
 - »[Helix Mobile Producer](#)« (installeras just när man behöver använda programmet eftersom det endast går att använda 30 dagar)
 - [Darwin](#)
 - Media-spelning: »[Real Player Basic 10](#)« med plugin »[Xiph Player Plugin](#)«
 - [Filöverföringsprogram](#)

Det senare är det klart roligaste och mest lärorika sättet eftersom man då måste installera allt som behövs på den egna datorn. Notera att ingen handledning ges på detta eftersom det finns alltför många olika varianter av konfigurationer.

OH-bilder

OH-bilderna finns enbart här:

- [00_intro.pdf](#)
- [01_content.pdf](#)
- [02_dhtml.pdf](#)
- [03_xhtml.pdf](#)
- [04_css.pdf](#)
- [05_javascript_om.pdf](#)

Tidigare tentor

Tidigare tentor finns enbart här:

- [Tidigare tentor](#)

Notera att de är baserade på de uppgifter som var obligatoriska vid tidpunkten för tentan och som möjligtvis inte är det nu.

Terminalprogram

På denna sida beskrivs hur man använder terminalprogram.

Terminalserver

Om man behöver logga in och köra kommandon på en avlägsen dator så använder man ett terminalprogram/klient. På DSV används följande dator som terminalserver:

- `triton.dsv.su.se`

Denna dator kör Linux och har både telnet- och SSH-server igång.

Om man inte släpps in kan det bero på att DSV's brandvägg inte släpper in just det IP-nummer som man använder. Då kan man kontakta [Christer Jansson](#) som kan registrera IP-nummret i brandväggen så att man kommer förbi den.

Telnet

I Win32 och Linux så kan man köra ett Terminal-program direkt i ett kommando-fönster:

- `telnet triton.dsv.su.se`

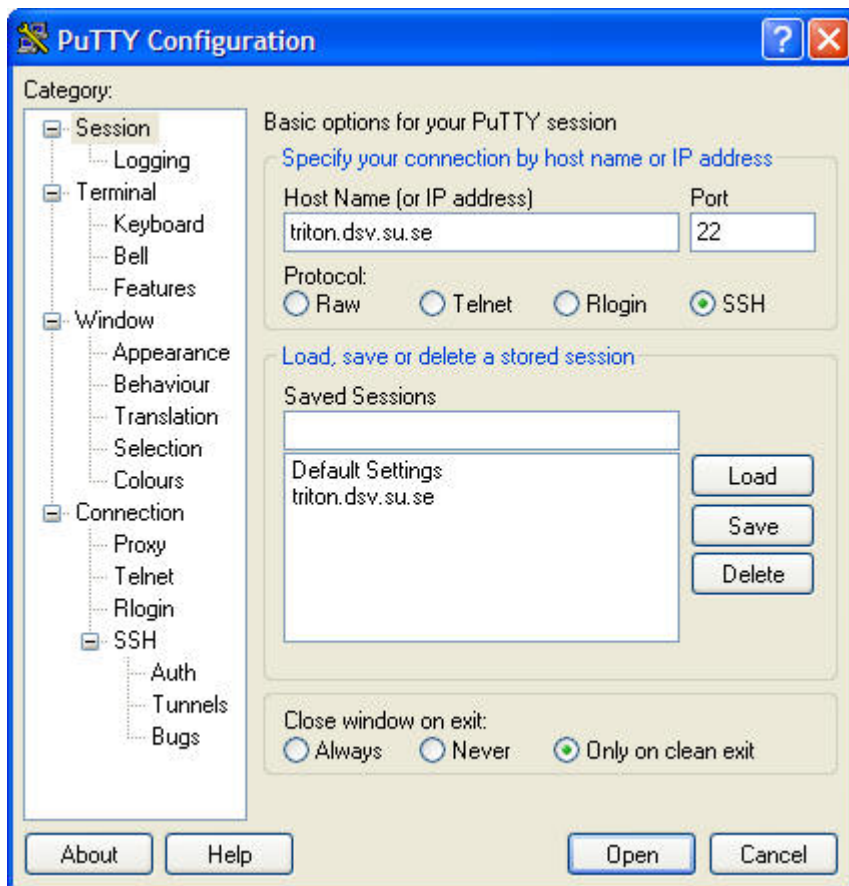
Man anger sedan användarnamn och lösenord.

SSH

När man använder telnet så sänds all information okrypterad över Internet och det kan vara farligt. Vill man undvika det kan man istället använda SSH och följande program rekommenderas:

- Win32: »[Putty](#)«
- Linux: `ssh`

Så här ser det ut när man startar Putty:



Editor

På denna sida beskrivs det verktyg man använder för att skapa kod med.

Intro

Man kan använda allt från en enkel editor till en avancerad grafisk utvecklingsmiljö. Valet är fritt! En avancerad grafisk utvecklingsmiljö kan vara något svårare att komma igång med men är överlägsen när man väl lärt sig den. En grafisk utvecklingsmiljö kan generera vissa delar av koden i ett program automatiskt, exempelvis den kod som tar han om det grafiska användargränssnittet, vilket snabbar upp konstruktionsarbetet betydligt. En stor nackdel med detta är att koden tenderar att bli omfattande och främmande för en nybörjare och det blir mycket svårt att anknyta sin egen kod till den kod som verktyget har skapat. Det bästa är att börja med en enkel editor och sedan gå över till en grafisk utvecklingsmiljö.

Observera att koden ska vara läsbar direkt i en vanlig browser för enklare rättning. Man bör därför inte använda en editor som använder kontrolltecken och tabbar för att indentera koden.

Enkla

- Notepad (Win 32)
- »[Pico](#)« (Unix)
- »[Emacs](#)« (Win32 och Unix)

Använder man Notepad så måste man spara filerna genom att skiva citationstecken runt filnamnet "X.java" när man namnger och sparar filen, annars sparas filen som X.java.txt.

Använder man emacs för Java bör man inte använda JDE-tillägget eftersom det lätt blir fel med classpath med mera. Kompilera, kör, sätt classpath med mera i ett kommandofönster istället:

- Skriv koden i en editor och spara filen (här x.java)
- Starta ett kommandofönster och gå till katalogen där koden ligger sparad
- Kompilera med `javac x.java`
- Kör med `java x`

Tips Win32:

- Har man Windows95/98/ME så bör man börja med att skriva doskey i kommandofönstret så att man kan använda "uppåtpil" och "nedåtpil" istället för att skriva om allt man skrivit tidigare
- Man kan lägga en genväg till DOS-promten i varje katalog och konfigurera (höger-klicka och välj Egenskaper) så att den startar i samma katalog:
 - Windows 98: Välj Program/Arbetsmapp och skriv en .
 - Windows 2000/XP: Välj Shortcut/Start in och gör detta text-fält tomt
- När det är långa krångliga kommandon så kan man i DOS använda BAT-filer genom att skriva kommandot i en fil med namnet run.bat så kan man sedan köra kommandot med run

Vill man strukturera upp sin XHTML-kod så rekommenderas:

- »[W3C: Tidy](#)«
- »[SourceForge: Tidy](#)«
- »[Charlie's Tidy Add-Ons](#)«

Mer komplexa

- »[SciTe](#)«, se även »[SciTe Bundled](#)«-versionen
- »[JEdit](#)« (för Java-kod)
- »[HTML-kit](#)« (för XML- och XHTML-kod)

Konfiguration av SciTE:

- Ladda den fil som bestämmer SciTE's funktion: Options / Load Global Options File
- Sen kan man sätta lite olika saker, exempelvis:
 - Använd 3 mellanslag vid intendering: `indent.size=3`
 - Använd inte tabbar vid intendering: `use.tabs=0`
- Spara sen filen

Sessionshantering med SciTE:

- Om man har många filer öppna och vill fortsätta med dessa senare så sparar man sessionen med: File / Save Session ...
- När man vill öppna alla filer igen sen så går man till: File / Load Session ...

Kompleta utvecklingsmiljöer

- »[Eclipse](#)« (för Java-kod)

Filöverföringsprogram

På denna sida beskrivs hur man använder filöverföringsprogram.

Intro

För att kunna sända filer mellan olika datorer använder man filöverföringsprogram och vanliga standarder för det är FTP, SFTP och FTPS.

Ett vanligt fall är när man har HTML-filer som man vill sända till DSV för att synliggöra dem på Internet. Man använder då sitt DSV/Unix/Linux-konto och lägger allt under `www`-katalogen. När filerna är överförda måste man även sätta rättigheterna på filerna och katalogerna:

- Kataloger: `chmod 755`
- De flesta filtyperna: `chmod 744 *`
- CGI-filer: `chmod 711 *.cgi`

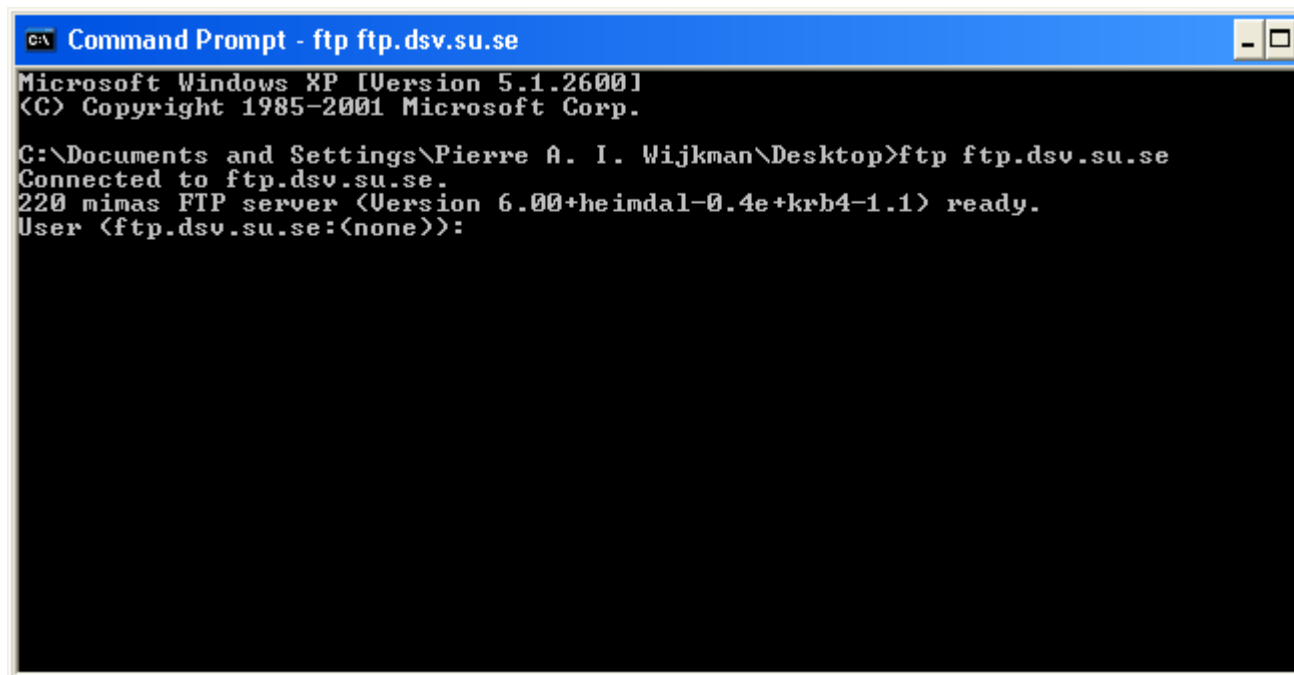
Notera att `www`-katalogen inte ska vara med i URL'ens sökväg när man sedan ska hämta filer i en browser:

- En fil `z.html` på DSV/Unix/Linux-kontot under katalogen `www/x/y` når man i en browser med `http://dsv.su.se/~<användarnamn>/x/y/z.html`

Man kan även nå sina filer på DSV/Win32 men det är endast DSV/Unix/Linux som har en tillförlitlig HTTP-server.

FTP

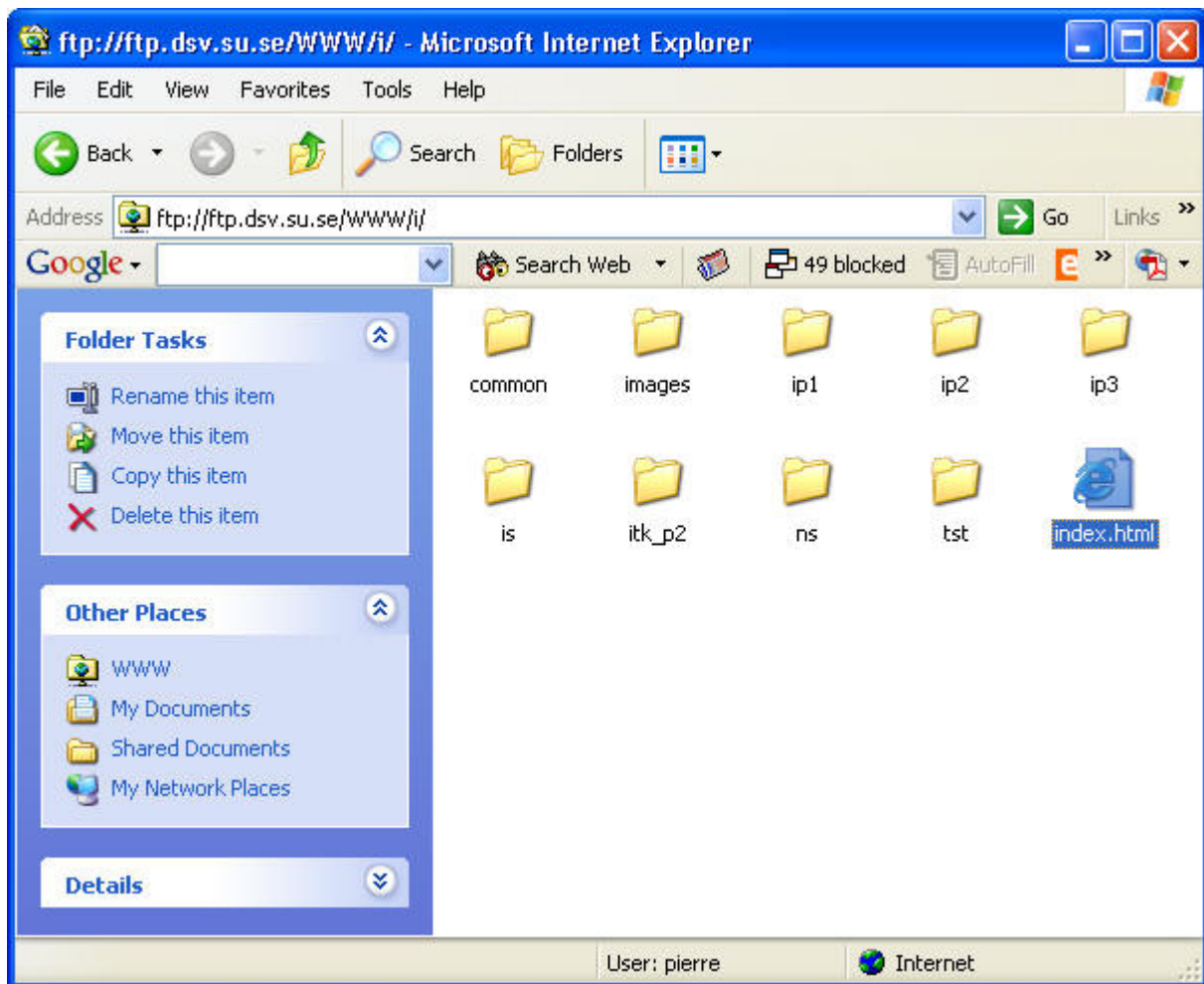
I Win32 och Linux så kan man köra en FTP-klient direkt i ett kommando-fönster:



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Pierre A. I. Wijkman\Desktop>ftp ftp.dsv.su.se
Connected to ftp.dsv.su.se.
220 mimas FTP server (Version 6.00+heimdal-0.4e+krb4-1.1) ready.
User (ftp.dsv.su.se:(none)):
```

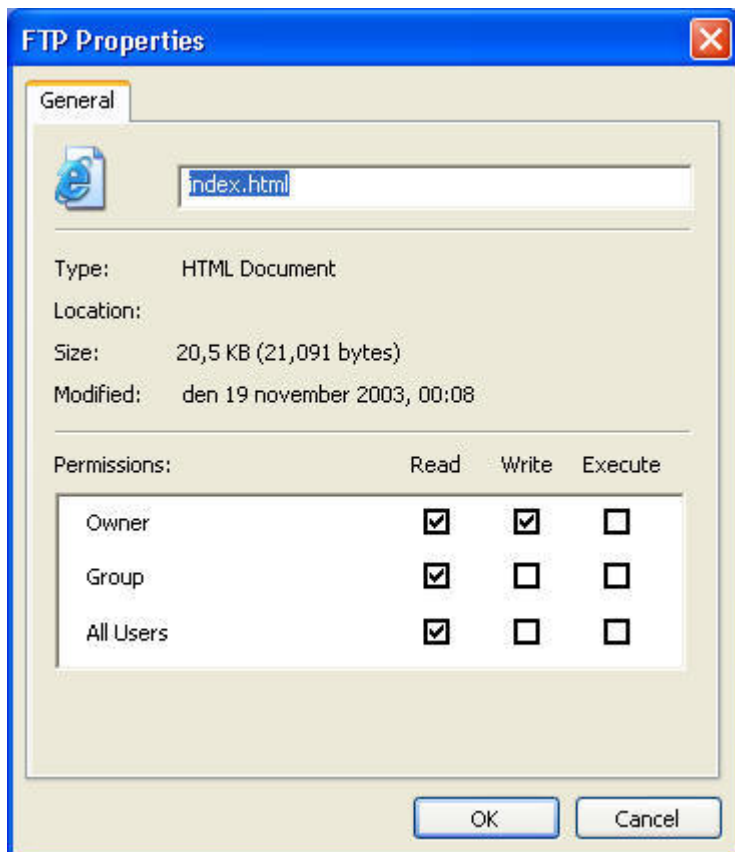
Det är dock ofta lättare att använda en GUI-baserad FTP-klient. Det finns många olika program och »Shareware.com« är ett ställe som man kan leta på. Man även använda en vanlig browser, exempelvis Internet Explorer:



Man använder sen drag-and-drop. Man kan även sätta fil- och katalogrättigheter från detta program:

- Högerklicka på fil eller katalog
- Välj Egenskaper
- Sätt rättigheter

Så här kan det se ut:



Notera dock att Internet Explorer inte stöder ASCII-mode.

Man ansluter sig till FTP-servern på följande dator:

- DSV/Unix/Linux: `ftp://ftp.dsv.su.se`

Använder man Internet Explorer så kan man skapa en genväg enligt:

- `ftp://<användarnamn><lösenord>@ftp.dsv.su.se`

Notera att man bör sända:

- Rena text-filer (som CGI-program med mera) i ASCII-mode
- Binära filer (som bilder, kompillerade program med mera) i BINARY-mode

Använder man en brandvägg så bör man se till att initieringen av dataöverföringskanalen sker på klient-sidan och det gör man genom att använda `passive mode` som kan ställas in i de flesta FTP-klienter.

SFTP och FTPS

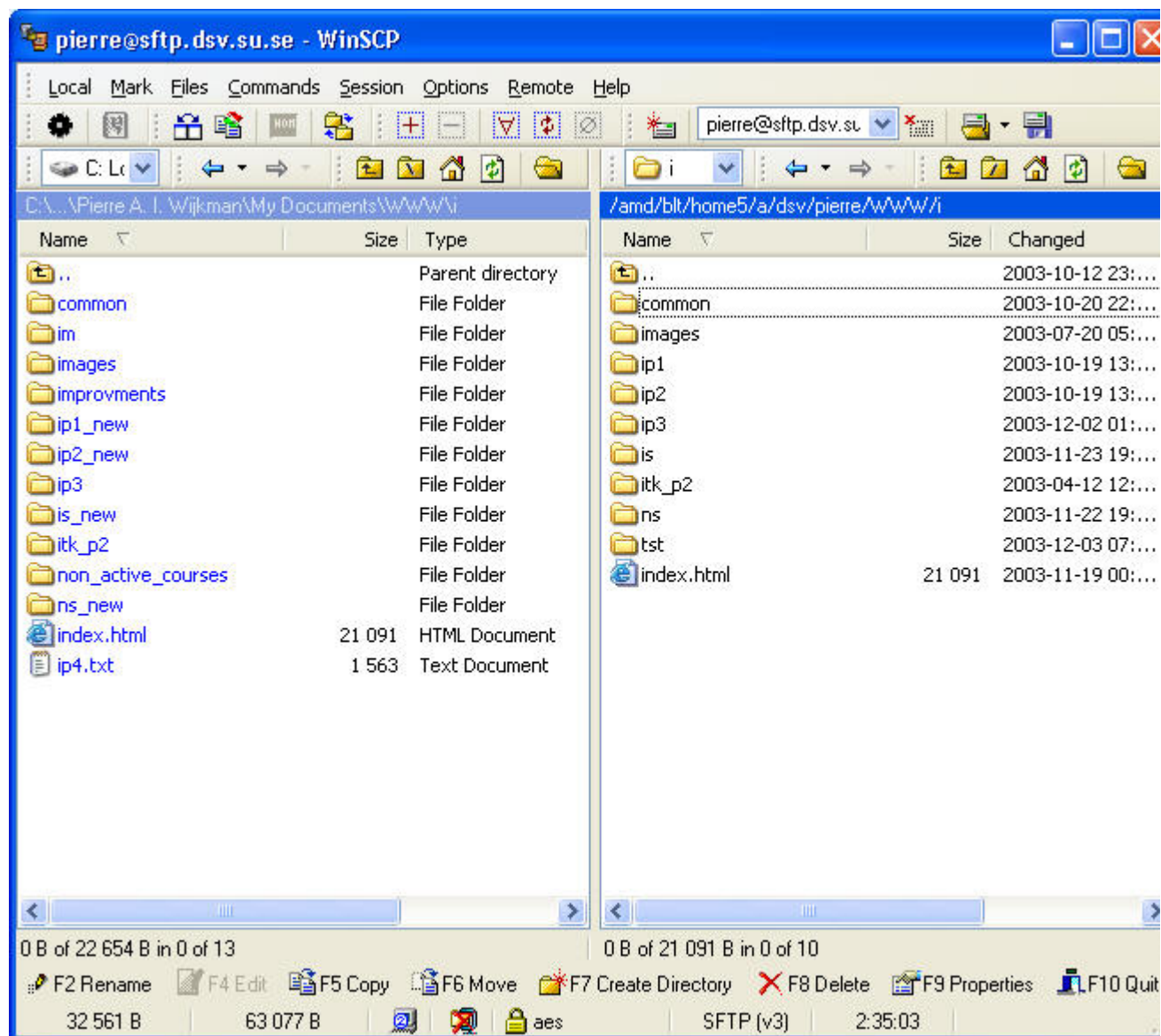
När man använder FTP så sänds all information, även lösenordet, okrypterat över Internet och det kan vara farligt. Man bör istället använda någon av följande tekniker:

- SFTP som använder sig av tunnling via SSH och först upprättar en SSH-anslutning som sedan används för att ansluta till en server via FTP
- FTPS som ansluter som vanlig FTP enligt RFC959 för att sedan begära en säker anslutning via SSL/TLS enligt RFC2228

DSV stöder enbart SFTP så man måste använda ett sådant program och bra program för det är:

- Unix/Linux: sftp som ingår i OpenSSH
- Win32: »[WinSCP](#)«

Exempelvis så ser WinSCP ut så här:



Man ansluter till SFTP-servern på följande datorer:

- DSV/Unix/Linux: sftp.dsv.su.se
- DSV/Win32: wftp.dsv.su.se

Detta säkrare sätt rekommenderas starkt.

Darwin

På denna sida finns hela Darwin beskrivningen.

Intro

På denna sida introduceras RTSP/RTP-servern Darwin Streaming Server som används för att

leverera strömmande media. Det finns två sätt att göra detta:

- Använda DSV's centrala Darwin Streaming Server
- Använda en egen Darwin Streaming Server

Både DSV's centrala Darwin Streaming Server och en egen Darwin Streaming Server kan användas för utveckling och testning men vi rekommenderar att man börjar med DSV's centrala Darwin Streaming Server.

@DSV

På denna sida beskrivs hur man använder DSV's RTSP/RTP-server för strömmande media.

Adress

På DSV körs en Darwin Streaming Server här:

- Host: atlas.dsv.su.se
- Port: 8000

Förfarande

Gör följande för att strömma en media-fil på DSV's Darwin Streaming Server:

- På ditt DSV/Unix/Linux-konto, skapa katalogen: `www/media`
- Lägg media-filerna i katalogen: `www/media`
- Accessa media-filerna via: `rtsp://atlas.dsv.su.se/pierre/users/<användarnamn>/<media-fil>`

Notera att man måste ha loggat in på denna webbplats med en DSV e-postadress för att detta ska fungera.

FAQ

Här följer de frågor och svar som brukar vara vanligast:

Fråga/svar 1

Media-filen spelas inte upp, vad ska jag göra?

Det vanligaste felen är:

- Man använder en media-typ som inte stöds av Darwin Streaming Server
- Man använder en media-typ som inte stöds av mediaspelaren

@Hemma

På denna sida beskrivs hur man använder en egen RTSP/RTP-server för strömmande media.

Installation och körning

- Ladda ner »[Darwin Streaming Server](#)«

- Packa upp filen
- Dubbelklicka på Install.bat och ange ett nytt användarnamn och nytt lösenord
- Eventuellt får man gå till Kontrollpanelen / Administrative Tools / Services och starta Darwin Streaming Server manuellt

Konfiguration

- Öppna en browser och ange URL: `http://localhost:1220/`

Tidigare tentor

Tentamen

Internetprogrammering III

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

1. Vilka tekniker ingår i DHTML? Beskriv dessa och hur de relaterar till varandra. Visa även med ett enkelt exempel.
2. Beskriv vad begreppen valid och well-formed står för i XML

Inga hjälpmedel. Pierre A. I. Wijkman ◻ 2000-05-02

Tentamen

Internetprogrammering III

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

1. Visa med ett exempel vilka tekniker som ingår i DHTML?
2. Beskriv vad begreppen valid och well-formed står för i XML

Inga hjälpmedel. Pierre A. I. Wijkman ◻ 2000-08-11

Tentamen

Internetprogrammering III

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Vilka tekniker ingår i DHTML? Beskriv dessa och hur de relaterar till varandra. Visa även med ett enkelt exempel där alla dessa tekniker ingår.

Inga hjälpmedel. Pierre A. I. Wijkman ◻ 2001-04-11

Tentamen

Internetprogrammering III

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Redogör för begreppen XML, DTD och XSLT.

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2001-04-18

Tentamen

Internetprogrammering III

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Till vad används XSLT och varför är XSLT bra?

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2001-05-29

Tentamen

Internetprogrammering III

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Beskriv först generellt Ditt gesällarbete (sista uppgiften). Beskriv sedan i större detalj hur Ditt gesällarbete är kodat, beskriv vilka klasser Du använt och hur dessa samarbetar (Du behöver givetvis inte beskriva all kod i dessa klasser).

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2001-08-30

Tentamen

Internetprogrammering III

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Vad är SVG? Visa ett enkelt exempel.

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2002-05-06

Tentamen

Internetprogrammering III

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Vad är SMIL? Visa ett enkelt kod-exempel.

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2002-05-30

Tentamen

Internetprogrammering III

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Beskriv ditt gesällprov ingående och visa med kod de delar som är extra intressanta.

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2002-08-20

Tentamen

Internetprogrammering III

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Vilka tekniker ingår i DHTML? Beskriv dessa och hur de relaterar till varandra. Visa även med ett enkelt exempel.

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2003-04-25

Tentamen

Internetprogrammering III

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Beskriv först generellt ditt gesällarbete (sista uppgiften). Beskriv sedan i större detalj hur de centrala delarna i ditt gesällarbete är kodat (man ska givetvis inte beskriva all kod).

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2003-08-26

Tentamen

Internetprogrammering III

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Gör ett DHTML-program som implementerar följande. När användaren för muspekaren över en text så ska:

1. Texten få en annan stilsättning
2. Ett alert-fönster visas

Man behöver inte ta hänsyn till om användaren använder Internet Explorer.

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2004-04-24

Tentamen

Internetprogrammering III

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Förklara följande HTML-taggar: `blockquote`, `br`, `code`, `div`, `em`, `h1`, `h2`, `h3`, `h4`, `h5`, `h6`, `p`, `span`, `strong`.

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2004-05-26

Tentamen

Internetprogrammering III

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 3 sidor.

Förklara följande CSS-egenskaper: `visibility`, `background`, `margin`, `border`, `padding`.

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2004-08-16

Tentamen

Internetprogrammering III

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 1 sida.

Redogör för RSS (eller RDF Site Summary).

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2005-02-22

Tentamen

Internetprogrammering III

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 2 sidor.

Redogör för SVG och SMIL.

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2005-03-08

Tentamen

Internetprogrammering III

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att

förtydliga. Totalt bör hela svaret omfatta minst 2 sidor.

Redogör för XSLT.

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2005-05-21

Externa källor

På denna sida finns länkar till externa källor. Tipsa gärna om fler bra länkar.

Struktur

- » [W3C: HyperText Markup Language Home Page](#)«
 - » [W3C: Validate Your Markup](#)«
 - » [W3C: XHTML 1.1 - Module-based XHTML](#)«
 - » [W3C: XHTML Basic](#)«
 - » [W3C: Modularization of XHTML](#)«
 - » [W3C: XHTML 2.0](#)«
 - » [W3C: Web Accessibility Initiative](#)«
- » [O'Reilly: XHTML: The Clean Code Solution](#)«
- » [ZVON](#)«

Stilsättning

- » [W3C: Cascading Style Sheets home page](#)«
 - » [W3C: CSS Validation Service](#)«
 - » [W3C: Cascading Style Sheets, level 2 revision 1 CSS 2.1 Specification](#)«
 - » [W3C: CSS Mobile Profile 1.0](#)«
 - » [W3C: Style Guide for online hypertext](#)«
 - » [W3C: Web Accessibility Initiative](#)«
- » [HWG: CSS Frequently Asked Questions](#)«
- » [TopStyle](#)«
- » [WebReference: Tutorial 24: Fixing Frames with Fixed Positioning](#)«
 - » [WebReference: Tutorial 24: Fixing Frames with Fixed Positioning: Example](#)«
- » [O'Reilly: Cascading Style Sheets: An Interview with Eric A. Meyer](#)«
- » [Glish: CSS Layout Techniques: for Fun and Profit](#)«
- » [Digital Web Magazine: Eric Meyer](#)«
- » [CSS Zen Garden](#)«
- » [Zen Garden: Resource Guide](#)«
- » [Microsoft: WEFT](#)«
- » [Font Editor](#)«

Innehållshantering

XSL/XSLT/XSL-FO

- » [W3C: The Extensible Stylesheet Language Family](#)«
- » [CSSToXSLFO](#)«

RSS och RDF

- » [UserLand: RSS 0.91](#)«
- » [UserLand: RSS Validator](#)«
- » [W3C: Resource Description Framework \(RDF\)](#)«
- » [O'Reilly: RSS](#)«

- » [O'Reilly: RSS and You](#)«
- » [WebReference: Introduction to RSS](#)«

Bild

SVG

- » [W3C: Scalable Vector Graphics](#)«
 - » [W3C: Scalable Vector Graphics 1.1 Specification](#)«
 - » [W3C: Mobile SVG Profiles: SVG Tiny and SVG Basic](#)«
 - » [W3C: SVG Embedding and Referencing Issues](#)«
- » [Adobe: SVG Zone](#)«
 - » [Adobe: SVG Zone: Tutorials](#)«
 - » [Adobe: SVG Zone: Demos](#)«
- » [Apache XML: Batik SVG Toolkit](#)«
- » [O'Reilly: SVG At the Movies](#)«
- » [WebReference: SVG: Modularized and mobile](#)«
- » [WebReference: SVG: The Art is in the Code](#)«
- » [Square: Web Vector Graphics Gallery](#)«
- » [TheOpenSourcery: Overview of Smart Graphics](#)«

2D

- » [SUN: Java 2D API](#)«
- » [SUN: Java Advanced Imaging API](#)«
- » [SUN: Java Tutorial: Trail: 2D Graphics](#)«
- » [JFreeChart](#)«
- » [Hexidec: Java Apps](#)«
- » [Anfys](#)«
- » [W3C: PNG](#)«
- » [WebReference: Web Color Reference](#)«
- » [WebMonkey: Death of the Websafe Color Palette?](#)«
- » [Ekdahl: Olika filformat och hur dom användes?](#)«
- » [JPEG2000](#)«
- » [O'Reilly: JPEG2000: the Killer Image File Format for Lossless Storage](#)«
- » [EETimes: JPEG2000 wavelet compression spec approved](#)«
- » [Mandrixx](#)«

3D

- » [SUN: Java 3D API](#)«
 - » [SUN: Java 3D API: Collateral](#)«
 - » [SUN: Java 3D API Specification](#)«
- » [J3D: The Java 3D Community Site](#)«
- » [LiveGraphics3D Homepage](#)«
- » [Web 3D Consortium](#)«
- » [Burning Pixel](#)«
- » [JOGL](#)«

Audio och video

Ogg

- » [Xiph](#)«
- » [Theora](#)«
- » [Vorbis](#)«
- » [Icecast](#)«

- » [NetJuke](#)«
- » [Flumotion](#)«
- » [Speex](#)«
- » [Lame](#)«
- » [Ogg Vorbis Help](#)«
- » [FFMPEG2 Theora](#)«
- » [Tobias: OggDS](#)«
- » [Ogg Vorbis Player for Symbian OS phones](#)«
- » [Fridgesoft: HaddiskOgg](#)«

MPEG-4

- » [Apple: QuickTime](#)«
 - » [Apple: QuickTime: Tutorials](#)«
 - » [Apple: Developer Connection QuickTime](#)«
- » [Apple: Darwin Streaming Server](#)«
- » [IBM: IBM Toolkit for MPEG-4](#)«
- » [IBM: IBM Toolkit for MPEG-4: FAQ](#)«
- » [IBM: Now showing: Your own online MPEG-4 cinema](#)«
- » [IBM Research: IBM MPEG-Technologies](#)«
- » [O'Reilly: Making Media from Scratch: Part 1](#)«
- » [O'Reilly: IBM's Java MPEG-4 Efforts Bear Fruit](#)«
- » [Linux Journal: Streaming MPEG-4 with Linux](#)«
- » [StreamingMedia: Creating Interactive Video With MPEG4](#)«
- » [StreamingMedia: Hinting For Quicktime And MPEG4](#)«
- » [MPEG4IP: Open Source, Open Standards, Open Streaming](#)«
- » [MpegAble X4 live](#)«
- » [Manifest Technologies: Next-Generation Video: MPEG-4 and Streaming Media](#)«
- » [Streaming Live MPEG-4](#)«
- » [Gromkov's Software: What is 3ivX codec?](#)«
- » [3ivx D4 4.5](#)«
- » [DivX](#)«

Real

- » [Real: Resources and Support](#)«
- » [Real: Production Topics](#)«
- » [Real: Review Documentation](#)«
- » [Real: Content Production and Authoring Documentation](#)«

Java

- » [SUN: Java Media Framework API](#)«
 - » [SUN: Java Media Framework API: JMF 2.1.1 Supported Formats](#)«

SMIL

- » [W3C: Synchronized Multimedia](#)«
- » [W3C: Displaying SMIL Basic Layout with a CSS2 Rendering Engine](#)«
- » [Real: RealNetworks Production Guide](#)«
- » [IBM: Get up to speed with SMIL 2.0](#)«
- » [O'Reilly: SMIL and CSS2 Rendering](#)«
- » [O'Reilly: Spicing-Up RealPlayer with JavaScript](#)«
- » [Empirenet: Learn SMIL with a SMIL](#)«
- » [Learning to Smil](#)«
- » [Helio: The SMIL Tutorial](#)«
- » [A List Apart: SMIL When You Play That](#)«
- » [WebMonkey: SMIL 2.0: Codeless Animation in HTML](#)«
- » [Oratrix: GRiNS Player for SMIL 2.0](#)«

- » [Web Adaptation Multimédia: LimSee2](#)«
- » [Inria: Des exposés scientifiques multimédia](#)«

Övrigt

- » [FFMPEG](#)«
- » [VideoLAN: VLC media player](#)«
- » [Intel: Intel IPP Video Samples](#)«
- » [MPlayer](#)«
- » [Cliprex Video Software: Cliprex DVD Player Professional Free DVD player software](#)«
- » [xine](#)«
- » [Buzzmachines](#)«
- » [AllForMP3: Glossary](#)«
- » [VideoHelp](#)«
- » [JMcgowan: AVI Overview](#)«
- » [JMcgowan: John McGowan's AVI Overview: Audio and Video Codecs](#)«

Dynamiska dokument

- » [W3C: Document Object Model](#)«
- » [Ecma International: Standard ECMA-327](#)«
- » [Mozilla: The DOM and JavaScript](#)«
- » [Mozilla: DOM Element Reference](#)«
- » [Mozilla: Gecko DOM Reference](#)«
- » [Mozilla: Introduction to the DOM](#)«
- » [Mozilla: DOM Sample Code](#)«
- » [Microsoft: Scripting](#)«
- » [Microsoft: JScript Features: ECMA](#)«
- » [O'Reilly: JavaScript: The Definitive Guide, 4th Edition: Sample Chapter 17](#)«
- » [O'Reilly: JavaScript: The Definitive Guide, 3rd Edition: Sample Chapter 13](#)«
- » [QuirksMode](#)«
- » [ZVON: DOM2 reference with examples](#)«
- » [HowToCreate: JavaScript tutorial](#)«
- » [EchoEcho: JavaScript Tutorial](#)«
- » [Jibbering: comp.lang.javascript FAQ](#)«
- » [JavaScript Kit: Introduction to the DOM of IE 5/ NS 6](#)«

Övrigt

- » [The Internet Engineering Task Force](#)«
- » [W3C](#)«
- » [DSV: Datormiljö](#)«
- » [KTHB Forumbiblioteket](#)«
- » [WebReference.com](#)«
- » [Webmonkey](#)«
- » [W3Schools](#)«
- » [Dev Shed Open Source Web Development](#)«
- » [CMD.exe](#)«

Uppgifter

På denna sida finns uppgifterna till kursen. Läs först noga alla [krav](#) och sedan alla [tips](#). Siffror inom () är kapitelhänvisningar.

Struktur

- 1.1 [XHTML 1.1 för stationära enheter](#) obligatorisk (XHTML: 1-8, 10, 13, 17)
- 1.2 [XHTML Basic för mobila enheter](#) obligatorisk (XHTML: 1-8, 10, 13, 17)
- 1.3 [Modularization of XHTML för egendefinerade XHTML-standarder](#) frivillig IP3, obligatorisk IP3+ (XHTML: 19, 20)
- 1.4 [XHTML och teckenuppsättningar](#) frivillig
- 1.5 [XHTML 2.0](#) frivillig

Stilsättning

- 2.1 [CSS 2.1 för stationära enheter](#) obligatorisk (XHTML: 14-17)
- 2.2 [CSS Mobile Profile för mobila enheter](#) obligatorisk (XHTML: 14-17)
- 2.3 [Dynamiskt anpassning](#) obligatorisk (XHTML: 14-17)

Innehållshantering

- 3.1 [Separation av data och XHTML med XML/XSLT/XPath](#) obligatorisk (XHTML: 21, 22)
- 3.2 [Modularisering av data](#) frivillig (XHTML: 21, 22)
- 3.3 [Syndikering med RSS och RDF Site Summary](#) obligatorisk

Bild

Bitmappad

- 4.1.1 [PNG/JPEG/p-JPEG/GIF/Moving-GIF/WBMP](#) obligatorisk (XHTML: 11)
- 4.1.2 [Java-baserad 2D](#) frivillig

Vektorbaserad

- 4.2.1 [SVG 1.1 för stationära enheter](#) obligatorisk
- 4.2.2 [SVG 1.1 för stationära enheter uppspelad med Java](#) frivillig
- 4.2.3 [SVG Tiny och SVG Basic för mobila enheter](#) frivillig
- 4.2.4 [Java-baserad 3D](#) frivillig

Audio och video

Inspelning/redigering/konvertering

- 5.1.1 [Audio med MP3 och Ogg Vorbis](#) obligatorisk
- 5.1.2 [Video med Ogg Theora](#) obligatorisk

Strömmande via RTSP/RTP/ISMA

- 5.2.1 [Audio med MP3 spellistor](#) obligatorisk
- 5.2.2 [Audio med MP3 spellistor uppspelad med Java](#) obligatorisk
- 5.2.3 [Video med MPEG-4 för stationära enheter](#) obligatorisk
- 5.2.4 [Video med MPEG-4 för stationära enheter uppspelad med Java](#) obligatorisk
- 5.2.5 [Video med 3GPP för mobila enheter](#) obligatorisk

Semi-strömmande via HTTP-interleaving/muxing

- 5.3.1 [Video med RealMedia uppspelad med RealPlayer](#) obligatorisk (XHTML: 12-13)
- 5.3.2 [Video med MPEG-4-MUX för stationära enheter uppspelad med Java](#) frivillig

Integrering

- 5.4.1 [SMIL 2.0 för stationära enheter](#) obligatorisk
- 5.4.2 [SMIL 2.0 för stationära enheter uppspelad med Java](#) frivillig
- 5.4.3 [SMIL i MMS för mobila enheter](#) frivillig

Dynamiska dokument

Objekt-modellen

- 6.1.1 [Egenskaper](#) frivillig IP3, obligatorisk IP3+ (JavaScript: 1-9, 12-14, 16, 17)
- 6.1.2 [Händelser](#) frivillig IP3, obligatorisk IP3+ (JavaScript: 1-9, 12-14, 17, 19)
- 6.1.3 [Metoder](#) frivillig IP3, obligatorisk IP3+ (JavaScript: 1-9, 12-14, 17)

Dokument-objekt-modellen

- 6.2.1 [Dynamisk struktur](#) frivillig IP3, obligatorisk IP3+ (JavaScript: 12-14, 17-19)
- 6.2.2 [Dynamisk stil](#) frivillig IP3, obligatorisk IP3+ (JavaScript: 12-14, 17-19)
- 6.2.3 [Dynamiskt innehåll](#) frivillig IP3, obligatorisk IP3+ (JavaScript: 14, 17)
- 6.2.4 [Formulär validering](#) frivillig IP3, obligatorisk IP3+ (JavaScript: 10)
- 6.2.5 [JavaScript-Java kopplingar](#) frivillig (JavaScript: 22)
- 6.2.6 [Databindning](#) frivillig (JavaScript: 17)

Gesällprov

- 7 [Gesällprov](#) obligatorisk

Krav

På denna sida finns de krav som gäller för uppgifterna.

Samarbete och betyg

- Alla uppgifter utom gesällprovet ska göras i grupp om 1-2 personer, gesällprovet ska göras ensam
- Vissa av uppgifterna är frivilliga men desto fler av dessa som man gör desto större blir sannolikheten för att få betyget VG/5
- Man kan givetvis ändå få betyget VG/5 om man gör ett väldigt bra gesällprov
- Bästa gesällprov till deadline 1 vinner ett pris

Kodens utseende

En vanlig orsak till betyget R är att nedanstående krav inte är uppfyllda.

All kod

- All kod ska vara välstrukturerad
- Man får själv bestämma stil på koden men man måste vara konsekvent
- Använd inte en editor som använder kontrolltecken och tabbar för indentering, koden ska kunna läsas och se bra ut i en vanlig enkel editor som exempelvis Notepad och därmed även i en browser
- Designen på det (eventuella) grafiska användargränssnittet är fri men försök göra det så

snyggt som möjligt (man bör alltså inte göra programmen identiska med de ofta lite tråkiga exempelprogram som ges för varje uppgift)

HTML-kod

- All HTML ska vara W3C-validerad med W3C's HTML validator enligt någon av följande standarder (se uppgifterna för vilken standard som ska följas i respektive uppgift):
 - XHTML 1.1
 - XHTML Basic
 - Modularisation of XHTML
 - XHTML 2.0 (kan ej valideras än)
- I XHTML-dokumentet får ingen annan kod finnas (ingen CSS-, JavaScript/OM-, JSP-, PHP-, ASP-kod med mera)
- All HTML-kod ska följa W3C riktlinjer för tillgänglighet (man får exempelvis inte stilsätta med tabeller)

CSS-kod

- All CSS ska vara validerad med W3C's CSS validator enligt antingen (se uppgifterna för vilken standard som ska följas i respektive uppgift):
 - CSS 2.1
 - CSS Mobile Profile 1.0
- All CSS kod ska finnas på en separat fil
- All CSS-kod ska följa W3C riktlinjer för tillgänglighet (man får exempelvis inte använda fonter med en fixerad storlek)

JavaScript/OM-kod

- All JavaScript/OM-kod ska vara välkommenterad med en kommentar per klass och metod
- All JavaScript ska följa ECMAScript 3
- All DOM-kod ska följa W3C DOM 2
- All JavaScript ska finnas på en separat fil
- All JavaScript/OM-kod ska fungera i Mozilla Firefox 1.0+
- I JavaScript koden får ingen HTML- eller CSS-kod finnas

Copyright

Om man letar efter material (bilder, audio, video med mera) på Internet måste man vara noga med att ta reda på om materialet är tillåtet att hämta hem och använda på sin egen sida. Material copyrightskyddas genom att:

- Det står uttryckligen direkt i samband med materialet
- Det står uttryckligen för hela webbsidan eller webbplatsen

Man får även se upp med webbplatser där material har lagts upp för gratis nedladdning eftersom det finns oseriösa webbplatser där påstått gratis material har hämtats från copyright skyddade källor. Notera dock att man ofta får använda material för skolprojekt och liknande och att ansvarig för kursen inte kommer att göra några kontroller på detta.

Tips

På denna sida finns tips för uppgifterna.

Ordning och organisation

- Använd en katalog för varje huvuduppgift
- Under hela konstruktionsprocessen:
 - Välstrukturerad kod
 - Bra namn på klasser, variabler, konstanter

Evolutionär utveckling

Många moderna programmeringsspråk består av:

- En liten språkkärna
- Ett mycket stort bibliotek av färdigkonstruerade klasser

Den självklara fördelen är att man slipper konstruera en stor mängd saker som annars skulle ta mycket lång tid att göra men nackdelen är att det kan vara svårt att veta och hitta vad som finns samt att man inte får samma grundläggande förståelse.

Matematik är ingenting man förstår, matematik är något som man vänjer sig vid

von Neumann

Till att börja med bör man skaffa sig en övergripande förståelse för det område man ska ge sig in på. Detta görs bäst med hjälp av böcker, tidningar och/eller de tutorials och artiklar som finns på Internet. När man har skaffat sig en denna övergripande förståelse så bör man så fort som möjligt testa några av de exempel man stött på. Det är ofta först när man praktiskt tillämpar ett område som man lär sig, eller vänjer sig vid, området och därmed får en förankring eller förståelse av beteendet hos objekten man laborerar med.

Man bör inte skriva av koden "rakt av" utan man bör istället:

- Konstruera en minimal körbar kärna (med sin egen stil på koden)

Dett gör man för varje nytt område man ger sig in på. Man kan sedan:

- Förändra denna minimala körbara kärna mot ett önskbart mål med **små förändringar** av koden

Detta är mycket viktigt! Man går mot ett önskat mål med:

- Små förändringar av koden
- Kompilering
- Testkörning

Man släpper aldrig kontakten med det fungerande programmet.

We are like sailors who on the open sea must reconstruct their ship but are never able to start afresh from the bottom. Where a beam is taken away a new one must at once be put there, and for this the rest of the ship is used as support. In this way, by using the old beams and driftwood the ship can be shaped entirely anew, but only by gradual reconstruction.

Otto Neurath

Detta är en mycket viktig tumregel. Försöker man förändra koden i stora steg så blir det fel och dessa fel är ofta svåra att lokalisera och rätta till. Ofta samverkar flera fel och detta gör det givetvis ännu svårare.

Ofta är stora program konstruerade av många lager/nivåer där varje nivå består av program konstruerade enligt ovanstående enkla principer. Man använder sedan program på en nivå som enheter för konstruktion av program på nästa nivå, och även denna process följer de ovan anvisade principerna.

Övrigt

- Detta är inte övningsuppgifter och man måste läsa kursboken och viss information som finns på Internet för att klara uppgifterna
- Alla program är testade i DSVs miljö för både Win32 och (det överlägsna :-) Unix/Linux
- Alla program går att utveckla lokalt ej uppkopplad (på Win32 och Unix/Linux) men man måste ha TCP/IP installerat på sin dator (vilket man har om man exempelvis surfar på Internet)
- Man kan använda IP-adressen 127.0.0.1 eller localhost som motsvarar den egna maskinen

Struktur: XHTML 1.1 för stationära enheter

Allmänt

Uppgiften introducerar till och går igenom det mesta inom XHTML 1.1 som är den senaste standarden för att strukturera innehåll i browsrar för stationära enheter.

Uppgift

Gör en webbplats som beskriver dig själv som består av flera olika (en förstasida, en CV-sida och så vidare), ej stilsatta, XHTML-dokument där varje XHTML-dokument följer standarden XHTML 1.1. XHTML-dokumenterna måste tillsammans innehålla minst de taggar nedan som är markerade med fet stil från respektive modul:

- Struktur: **body, head, html, title**
- Text: abbr, acronym, address, **blockquote**, br, cite, **code**, dfn, **div, em, h1, h2, h3**, h4, h5, h6, kbd, **p**, pre, q, samp, **span, strong**, var
- Hypertext: **a**, testa:
 - Relativ och absolut länk
 - Länk till ankare
 - Bild-länk
 - Länk till olika protokoll
 - Mailto-länk
- Listor: **dl, dt, dd, ol, ul, li**
- Objekt: **object**, param
- Presentation: b, big, hr, i, small, sub, sup, tt
- Editering: del, ins
- Dubbelriktad text: <hoppa över>
- Formulär: **button, fieldset, form, input, label, legend, select, optgroup, option, textarea** (man behöver inte länka till ett fungerande HTTP-serversides program)
- Tabell: **caption, col, colgroup, table, tbody, td, tfoot, th, thead, tr**
- Bild: **img**
- Bildkartor, klient-sidan: area, map
- Bildkartor, server-sidan: <hoppa över>
- Händelser: <hoppa över>
- Metainformation: meta-taggar med name-attributen: **author, description, generator, http-equiv, keywords, robots**
- Skript: **noscript, script** (länka till ett tomt JavaScript-program)

- Stil: <hoppa över>
- Länk: **link** (länka till ett tomt CSS-dokument)
- Bas: base

Om man länkar till bilder med mera så lägg bilderna lokalt på samma ställe som XHTML-dokumentet eftersom det annars kostar bandbredd för andra. Notera att man inte får placera ut data/innehåll med tabeller, se sidan Krav.

Testa allt med W3C's validator.

Exempel

Se sidan »[Pierre A. I. Wijkman](#)« och klicka på [Click here to disable CSS](#) för att ta bort stilsättningen (denna sida använder dock inte alla de taggar som ska användas för uppgiften).

Tips

Om man testar map-taggen så kan man få problem. Enligt XHTML 1.1 så ska man använda bildkartor enligt:

```

<map id="x">
  <area shape="circle" coords="78, 123, 50" href="x.html" alt="B"/>
  <area shape="rect" coords="0, 0, 50, 50" href="y.html" alt="C"/>
</map>
```

Här är värdet för attributet usemap ett id-värde. Detta fungerar inte än i alla browsrar som är vana vid att usemap använder ett name-värde. En temporär lösning är att använda:

```

<map id="x" name="x">
  <area shape="circle" coords="78, 123, 50" href="x.html" alt="X"/>
  <area shape="rect" coords="0, 0, 50, 50" href="y.html" alt="Y"/>
</map>
```

Dokumentet validerar då inte som XHTML 1.1. En bra taktik är då att försöka följa XHTML 1.1 så långt det går och först använda det korrekta sättet för att få ett validerat XHTML 1.1-dokument och sen ändra just map-taggen till ett sätt som går att använda. Notera dock att denna uppgift ska validera som XHTML 1.1.

Struktur: XHTML Basic för mobila enheter

Allmänt

Uppgiften introducerar till och går igenom det mesta inom XHTML Basic som är den senaste standarden för att strukturera innehåll i browsrar för mobila klienter. WML 2.0 består av just XHTML Basic och CSS Mobile Profile.

Uppgift

Kopiera XHTML-dokumenterna från uppgiften [Struktur: XHTML 1.1 för stationära enheter](#) och ändra i dessa så att de nu följer XHTML Basic. Testa allt med W3C's validator.

Exempel

Kursens webbplats använder XHTML Basic (dock med stilsättning och inte med alla taggar som ska användas i uppgiften).

Tips

XHTML Basic 1.0 har följande moduler och taggar:

- Struktur: body, head, html, title
- Text: abbr, acronym, address, blockquote, br, cite, code, dfn, div, em, h1, h2, h3, h4, h5, h6, kbd, p, pre, q, samp, span, strong, var
- Hypertext: a
- Listor: dl, dt, dd, ol, ul, li
- Formulär: form, input, label, select, option, textarea
- Tabell: caption, table, td, th, tr
- Bild: img
- Objekt: object, param
- Metainformation: meta
- Länk: link
- Bas: base

Man kan man testa hur WML-sidor ser ut i mobila enheter på »[Gelon](#)«.

Struktur: Modularization of XHTML för egendefinerade XHTML-standarder

Allmänt

Uppgiften introducerar till hur man kan skapa egna XHTML-standarder från den mängd moduler som finns i Modularization of XHTML.

Uppgift

Gör en egen XHTML standard som består av följande moduler:

- Samma moduler som finns XHTML Basic
- Target modulen

Gör den DTD som definierar denna standard nåbar från Internet.

Kopiera sen XHTML-dokumentet från uppgiften [Struktur: XHTML Basic för mobila enheter](#) och lägg till:

- Den nya DOCTYPE-deklarationen
- Attributet target="_blank" på alla länkar som går till platser utanför den egna webblasen (så att dessa öppnas i ett nytt fönster)

Testa allt med W3C's validator.

Exempel

Ett [enkelt exempel](#) fast med XHTML 1.1 och target modulen.

Tips

Modularization of XHTML består av följande moduler:

- Kärnmoduler:
 - Structure
 - Text
 - Hypertext
 - List
- Övriga moduler:
 - Applet (deprecated)
 - Presentation
 - Edit
 - Bi-Directional Text
 - Basic Forms
 - Forms
 - Basic Tables
 - Tables
 - Image
 - Client-Side Image Map
 - Server-Side Image Map
 - Object
 - Frames
 - Target
 - Iframe
 - Intrinsic Events
 - Metainformation
 - Scripting
 - Style Sheet
 - Style Attribute (deprecated)
 - Link
 - Base
 - Name Identification (deprecated)
 - Legacy

Se vidare:

- » [W3C: XHTML Abstract Modules](#)«
- » [W3C: Developing DTDs with defined and extended modules](#)«
- » [Accessify: Standards-Compliant New Windows](#)«

Struktur: XHTML och teckenuppsättningar

Allmänt

Uppgiften visar hur man skapar ett XHTML-dokument med en annan teckenuppsättning.

Uppgift

Skapa ett XHTML 1.1-dokument som har:

- Tecken från exempelvis farsi, arabiska eller andra teckenuppsättningar
- En (svensk :) beskrivning på hur du gjort detta

Exempel

Ett [enkelt exempel](#) utan beskrivning.

Tips

Inget tips ges.

Struktur: XHTML 2.0

Allmänt

Uppgiften introducerar till XHTML 2.0.

Uppgift

Kopiera XHTML-dokumenterna från uppgiften [Struktur: XHTML 1.1 för stationära enheter](#) och ändra i dessa så att de nu följer (den nuvarande ej helt färdiga standarden) XHTML 2.0.

Exempel

Inget exempel ges.

Tips

Inget tips ges.

Stilsättning: CSS 2.1 för stationära enheter

Allmänt

Uppgiften introducerar till och går igenom det mesta inom CSS 2.1 som är den senaste standarden för att stilsätta XHTML-dokument i browsrar för stationära enheter.

Uppgift

Kopiera XHTML-dokumenterna från uppgiften [Struktur: XHTML 1.1 för stationära enheter](#) och stilsätt dessa XHTML-dokument med en extern CSS 2.1-fil som är gemensam för alla XHTML-dokument så att XHTML/CSS-dokumentet får:

- Dynamisk anpassning efter fönstrets bredd utan tomma ytor på kanterna och utan utfyllnad med bakgrund och så vidare
- En proportionerlig stlsättning som fungerar för alla användarval av teckenstorlek, om användaren ändrar till en större textstorlek så ska även bilder med mera ändras i samma grad
- En övre rubrikarea
- Minst tre kolumner under denna rubrikarea
- Text bredvid bild som inte innehåller rubriker

CSS-filen måste innehålla minst de egenskaper nedan som är markerade med fet stil från respektive modul:

- Teckensnitt: **font**, **font-family**, font-stretch, **font-size**, font-size-adjust, font-style, font-variant, **font-weight**
- Text: **letter-spacing**, word-spacing, **line-height**, text-align, **text-decoration**, text-shadow, text-indent, **text-transform**, white-space
- Listor: list-style, list-style-image, list-style-position, **list-style-type**, marker-offset
- Färg: **color** (testa med både egna färger och med fördefinierade systemfärger)
- Synlighet: visibility, clip, display, overflow
- Bakgrund: **background**, background-attachment, **background-color**, background-image, background-position, background-repeat
- Marginaler: **margin**, margin-top, margin-right, margin-bottom, margin-left
- Kantlinjer:
 - **border**, border-top, border-right, border-bottom, border-left
 - **border-color**, border-top-color, border-right-color, border-bottom-color, border-left-color
 - **border-style**, border-top-style, border-right-style, border-bottom-style, border-left-style
 - **border-width**, border-top-width, border-right-width, border-bottom-width, border-left-width
- Outlines: outline, outline-color, outline-style, outline-width
- Paddning: **padding**, padding-top, padding-right, padding-bottom, padding-left
- Tabeller: table-layout, border-collapse, border-spacing, caption-side, empty-cells
- Storlek: **width**, **height**, max-width, max-height, min-width, min-height, vertical-align
- Position: **position**, **top**, **left**, bottom, **right**, **z-index**, **float**, **clear**
- Övrigt: **cursor**

Använd olika selektorer:

- id
- class
- tagg, testa även a:link, a:active, a:visited, a:hover, p:first-line, p:first-letter

Följande enheter finns men man måste mest använda realiva enheter som em för att kunna uppfylla kraven ovan:

- Relativa: em, ex, px, %
- Absoluta: in, cm, mm, pt, pc

Notera att vissa CSS 2.1 egenskaper inte är implementerade i alla browsers än. Man kan behöva lägga till viss XHTML-kod (exempelvis id-märkta div-taggar för att definiera olika områden) och om man tycker att vissa XHTML-taggar från det tidigare XHTML 1.1-dokumentet inte behövs så är det ok att ta bort dessa. Testa allt med W3C's validator.

Exempel

Se sidan »[Pierre A. I. Wijkman](#)« (denna sida använder dock inte alla de egenskaper som ska användas för uppgiften).

Tips

När man använder font-family så måste man tänka på att en klient endast stöder vissa typsnitt. Det är ett fåtal typsnitt som alltid stöds:

- Linux/Unix:
 - Serif: Times
 - Sans-serif: Helvetica
 - Monospace: Courier
- Mac Classic:
 - Serif: Times
 - Sans-serif: Helvetica
 - Monospace: Courier
- Mac OS X:
 - Serif: Times, Times New Roman
 - Sans-serif: Arial, Helvetica
 - Monospace: Courier New, Courier
- Win32:
 - Serif: Times New Roman
 - Sans-serif: Arial
 - Monospace: Courier New

Se vidare »[Upsdell: Resources Fonts](#)«. För att lösa detta problem kommer man i framtiden att kunna använda nerladdningsbara fonter. Just nu finns tyvärr ingen standard klar för detta, det är två tekniker som tävlar: Embedded OpenType och TrueDoc.

Stilsättning: CSS Mobile Profile för mobila enheter

Allmänt

Uppgiften introducerar till och går igenom det mesta inom CSS Mobile Profile som är den senaste standarden för att stilsätta XHTML-dokument i browsrar för mobila enheter.

Uppgift

Kopiera:

- XHTML-dokumentet från uppgiften [Struktur: XHTML Basic för mobila enheter](#)
- CSS-filen från uppgiften [Stilsättning: CSS 2.1 för stationära enheter](#)

Ändra CSS-filen så att den nu följer CSS Mobile Profile med bibehållen stilsättning så långt det går och stilsätt XHTML-dokumentet med denna nya CSS-fil.

Man kan behöva lägga till viss XHTML-kod (exempelvis id-märkta div-taggar för att definiera olika områden) och om man tycker att vissa XHTML-taggar från det tidigare XHTML Basic-dokumentet inte behövs så är det ok att ta bort dessa. Testa allt med W3C's validator. Notera att W3C's validator för CSS Mobile Profile kan vara bristfällig och att man då istället får gå efter listan i »[W3C: CSS Mobile Profile 1.0](#)«.

Exempel

Kursens webbplats använder CSS Mobile Profile (dock inte med alla egenskaper som ska användas i uppgiften).

Tips

CSS Mobile Profile har följande moduler:

- Teckensnitt: font, font-family, font-size, font-style, font-variant, font-weight
- Text: text-align, text-decoration, text-indent, text-transform, white-space
- Lister: list-style, list-style-image, list-style-position, list-style-type
- Färg: color
- Synlighet: visibility, display
- Bakgrund: background, background-color, background-image, background-position, background-repeat
- Marginaler: margin, margin-top, margin-right, margin-bottom, margin-left
- Kantlinjer:
 - border, border-top, border-right, border-bottom, border-left
 - border-color, border-top-color, border-right-color, border-bottom-color, border-left-color
 - border-style, border-top-style, border-right-style, border-bottom-style, border-left-style
 - border-width, border-top-width, border-right-width, border-bottom-width, border-left-width
- Paddning: padding, padding-top, padding-right, padding-bottom, padding-left
- Storlek: width, height, vertical-align
- Position: float, clear

Stilsättning: Dynamiskt anpassning

Allmänt

Uppgiften visar hur man stilsätter samma XHTML-dokument med olika CSS-dokument beroende på vilken typ av enhet det är som ska visa XHTML-dokumentet.

Uppgift

Kopiera:

- XHTML-dokumentet från uppgiften [Struktur: XHTML Basic för mobila enheter](#)
- CSS-filen från uppgiften [Stilsättning: CSS 2.1 för stationära enheter](#) och kalla den för screen.css
- CSS-filen från uppgiften [Stilsättning: CSS Mobile Profile för mobila enheter](#) och kalla den för handheld.css

Skapa två nya CSS-filer:

- En CSS Mobile Profile-fil kallad color.css dit man flyttar allt som har med färgsättning att göra från screen.css och handheld.css
- En CSS 2.1-fil kallad print.css speciellt gjord för utskrift där man exempelvis kan testa modulen för sidbaserade egenskaper: page, page-brake-after, page-brake-before, page-brake-inside, size, orphans, widows, marks (men tänk på att det inte stöds av alla browsers)

Stilsätt nu XHTML-dokumentet så att:

- För stationära enheter används: color.css, screen.css
- För mobila enheter används: color.css, handheld.css
- För skrivare används: print.css

Notera att vi genom att modularisera CSS-filerna kan återanvända moduler för olika enheter. Man kan naturligtvis föra modulariseringen längre än i denna uppgift. Testa att det fungerar på:

- En stationär enhet
- En mobil enhet (om möjlighet finns)
- Papper

Testa allt med W3C's validator.

Exempel

Kursens webbplats använder denna teknik. Ett [enkelt exempel](#) som inte är av den omfattning som krävs i denna uppgift.

Tips

Titta närmare på `link`-taggens `media`-attribut.

Innehållshantering: Separation av data och XHTML med XML / XSLT / XPath

Allmänt

I uppgiften [Stilsättning: Dynamiskt anpassning](#) så var vi tvugna att använda ett XHTML-dokument som följde den mest strikta standarden (i det här fallet XHTML Basic) av de varianter på XHTML-standarder som skulle kunna visas på de (i det här fallet) tre olika typerna av enheter. Det är förstås en nackdel om man exempelvis vill använda mer avancerade XHTML 1.1 på stationära enheter. Den här uppgiften visar hur man kan lösa det problemet genom att separera data och XHTML med hjälp av XSLT och XPath. Det är även mycket bra att ha data separat på ett enda ställe för att lätt kunna uppdatera datat.

Uppgift

Skapa ett XML-dokument med egna beskrivande taggar och attribut för det data/innehåll som finns för huvud XHTML-dokumentet i uppgiften [Struktur: XHTML 1.1 för stationära enheter](#) (frivilligt att ta med samtliga XHTML-dokument i webbplatsen). Detta XML-dokument är alltså en ren datafil för detta huvud XHTML-dokument.

Skapa sedan följande två XSLT-dokument:

- Ett som transformerar XML-dokumentet till det stilsatta XHTML-dokumentet i uppgiften [Stilsättning: CSS 2.1 för stationära enheter](#)
- Ett som transformerar XML-dokumentet till det stilsatta XHTML-dokumentet i uppgiften [Stilsättning: CSS Mobile Profile för mobila enheter](#)

Man kopierar givetvis respektive CSS-fil till katalogen för denna uppgift. Länka till sist till dessa två XSLT-dokument från XML-dokumentet och använd `media` attributet för att få det att fungera på olika typer av enheter.

Notera att inte alla browsrar stöder XSLT ännu och att exempelvis Firefox har vissa brister så man bör även testa med andra browsrar som exempelvis IE. Notera särskilt att mobila klienter ofta inte har något stöd alls för XSLT och om man senare arbetar med denna teknik i industrin så bör man utföra transformationen på HTTP-serversidan istället. Det finns mycket bra stöd för det i de flesta

HTTP-serversides tekniker.

Frivillig överkurs (utan handledning):

- Titta på hur man kan modularisera XSLT-dokument med `import`-taggen
- Titta på hur man kan stilsätta med XSL-FO istället för med CSS

Testa allt med W3C's validator. XML-dokumentet kan man testa direkt i browsern som har en inbyggd XML-parser.

Exempel

Ett [enkelt exempel](#) som inte är av den omfattning som krävs i denna uppgift.

Tips

Länkningen görs med:

```
<?xml-stylesheet type="text/xsl" href="a_3_1_handheld.xsl" media="handheld"?>
<?xml-stylesheet type="text/xsl" href="a_3_1_screen.xsl" media="screen"?>
```

Notera att Firefox 1+ inte verkar ha så bra stöd än för olika media-typer.

Innehållshantering: Modularisering av data

Allmänt

Uppgiften tar upp hur man delar upp data för att kunna återanvända och kombinera delar av data i olika kombinationer för olika ändamål.

Uppgift

Ofta återanvänder man data/innehåll på en webbplats, exempelvis:

- Header
- Footer
- Navigation
- Användarinformation

Men även annat data/innehåll brukar återanvändas på stora webplatser. Diskutera hur detta kan implementeras på ett bra sätt och peka på och beskriv etablerade tekniker (exempelvis XInclude) och/eller egna lösningar om sådana finns.

Exempel

Kursens webbplats använder en variant av detta tema. Data/innehåll återanvänds och data/innehåll finns lagrat på text-filer som använder ett egendefinierat minimalt WIKI-språk (alltså används inte XML som bas för data/innehåll) och dessa textfiler kan kombineras ihop genom att man i WIKI-språket kan importera andra textsidor. Detta kan även göras rekursivt.

Tips

Inga tips ges.

Innehållshantering: Syndikering med RSS och RDF Site Summary

Allmänt

Uppgiften tar upp hur man meta-beskriver data/innehåll i en XHTML-sida.

Uppgift

Använd någon av standarderna RSS och/eller RDF Site Summary för att meta-beskriva innehållet i uppgiften [Innehållshantering: Separation av data och XHTML med XML/XSLT/XPath](#).

Kopiera sedan filerna från uppgiften [Innehållshantering: Separation av data och XHTML med XML/XSLT/XPath](#) och lägg till en länk till denna meta-beskrivning i XSLT-filen så att det genererade XHTML-dokumentet även får denna länk.

Testa allt med W3C's validator.

Exempel

Se sidan »[Pierre A. I. Wijkman](#)« och klicka på den orange bilden med texten XML.

Tips

Inga tips ges.

Bild: Bitmappad: PNG / JPEG / p-JPEG / GIF / Moving-GIF / WBMP

Allmänt

Uppgiften introducerar till bitmappade bildformat. De flesta browsrar har inbyggt stöd för olika format för bitmappade bilder och dessa format tas upp i denna uppgift. Se »[NIST: Digital Media File Types: Survey of Common Formats](#)« för en jämförelse mellan några av de vanligaste bild-formaten.

Uppgift

Utvidga och uppdatera dina båda webplatser från uppgifterna [Stilsättning: CSS 2.1 för stationära enheter](#) och [Stilsättning: CSS Mobile Profile för mobila enheter](#) så att de innehåller finslipade bilder med formaten PNG, JPEG, p-JPEG, GIF och (en egengjord) Moving-GIF. Notera att olika format är bra för olika saker.

Frivillig överkurs är att titta på formatet WBMP som är optimerat för att ha en minimal filstorlek och fungera bra på enklare mobila enheter.

Exempel

Inget exempel ges.

Tips

- Icke-förstörande kompression har PNG (som utvecklades på grund av att GIF är patenterad) samt GIF och man kan exempelvis experimentera med:
 - Transparens
 - Sammanflätning
 - Animation
 - Olika bitdjup och storlekar på bilden för att se hur det påverkar bildens filstorlek
- Förstörande kompression har JPEG samt p-JPEG och man kan exempelvis experimentera med:
 - Progressivitet
 - Olika komprimeringsgrader och storlekar på bilden för att se hur det påverkar bildens filstorlek

En Moving-GIF bild kan göras exempelvis med [GIFAnimator](#) för Win32 eller roliga »[Morpheus](#)« för Win32.

Bild: Bitmappad: Java-baserad 2D

Allmänt

Vill man använda dynamiska 2D bilder är Java ett bra alternativ och denna uppgift introducerar till detta.

Uppgift

Gör ett XHTML 1.1-dokument med ett egengjort Java-äpple som visar dynamiska 2D bilder.

Exempel

Ett [enkelt exempel](#).

Tips

Se »[SUN: Java 2D API](#)« och »[SUN: Java Advanced Imaging \(JAI\) API](#)«.

Notera att man med denna teknik även kan koppla tillbaka till serversidan (som äpplet laddades från) för att hämta dynamiska data för bildgenereringen.

Slutligen några tips om [Mandelbrots mängd](#) för de som vill implementera uppgiften på detta sätt.

Bild: Vektorbaserad: SVG 1.1 för

stationära enheter

Allmänt

SVG är ett vektorbaserad bildformat med inbyggda funktioner för tidsstyrd och användarstyrd animering (både via inbyggda funktioner hämtade från SMIL-standarden och JavaScript) och många andra saker. SVG är likt Flash med skillnaden:

- SVG är en W3C standard, öppet och gratis
- SVG baseras på XML och man behöver inget speciellt (dyrt) utvecklingsverkty

Uppgiften introducerar till vektorgrafik för stationära enheter. Notera att vi nu kommer in på ett område där vi använder format som ännu inte stöds direkt i alla browsrar än och att man då måste använda en SVG-plugin.

Uppgift

Gör ett XHTML 1.1-dokument med en egengjord SVG 1.1 bild. Experimentera med följande:

- Gränsboxen (the bounding box)
- Kurvor och former
- Text
- CSS
- Fyllnader
- Streck
- Gradienter
- Filtereffekter
- Bitmappade bilder
- Transformationer

Testa allt med W3C's validator. Frivillig överkurs är att titta på dynamisk addering och borttagning av SVG-element (kräver JavaScript).

Exempel

Ett [enkelt exempel](#) som inte är av den omfattning som krävs i denna uppgift.

Tips

Man bör även här använda en separat CSS-fil som man länkar till med en absolut adress:

```
<?xml-stylesheet href="http://atlas.dsv.su.se/~pierre/i/05_ass/ip3/a_4_2_1/a_4_2_1.css" type="text/css"
```

Om man vill ha en enhetlig design på webbplatsen så kan man låta vissa CSS-filer vara gemensamma för XHTML- och SVG-dokumenterna.

För de som vill testa JavaScript tillsammans med en SVG-plugin så kan man göra det på två sätt:

- Man kan länka till JavaScript-filen från XHTML-dokumentet vilket medför att browserns JavaScript-motor används som i sin tur anropar funktioner i SVG-plugin'en
- Man kan länka till JavaScript-filen från SVG-dokumentet vilket medför att SVG-plugin'ens JavaScript-motor används direkt

Vissa browsrar har inte stöd för JavaScript-anrop till en plugin och då måste man använda den

andra varianten med en absolut adress i länken till JavaScript-filen:

```
<script xmlns:xlink="http://www.w3.org/1999/xlink" xlink:href="http://atlas.dsv.su.se/~pierre/i/05_as
```

Funktionen `clearInterval` fungerar inte i Adobes SVG-plug-in.

Bild: Vektorbaserad: SVG 1.1 för stationära enheter uppspelad med Java

Allmänt

Uppgiften belyser hur man använder Java för att spela upp SVG-bilder.

Uppgift

Gör ett XHTML 1.1-dokument med ett Java-äpple som spelar upp SVG 1.1-bilden från uppgiften [Bild: Vektorbaserad: SVG 1.1 för stationära enheter](#). Möjligtvis måste SVG 1.1-bilden modifieras något.

Exempel

Inget exempel ges.

Tips

Se exempelvis »[Apache XML: Batik](#)«.

Bild: Vektorbaserad: SVG Tiny och SVG Basic för mobila enheter

Allmänt

Uppgiften introducerar till vektorgrafik för mobila enheter.

Uppgift

Gör ett XHTML Basic-dokument med en egengjord SVG Tiny och/eller en egengjord SVG Basic bild. Kopiera filerna från uppgiften [Bild: Vektorbaserad: SVG 1.1 för stationära enheter](#) och ändra i SVG-dokumentet så att det validerar som SVG Tiny eller SVG Basic med W3C's validator.

Exempel

Inget exempel ges.

Tips

Inga tips ges.

Bild: Vektorbaserad: Java-baserad 3D

Allmänt

Vill man använda dynamiska 3D bilder är Java ett bra alternativ och denna uppgift introducerar till detta.

Uppgift

Gör ett XHTML 1.1-dokument med ett egengjort Java-äpple som visar dynamiska 3D bilder.

Exempel

Ett [enkelt exempel](#) som enbart är förberett för Win32 men som går att utöka så att det även fungerar på andra operativsystem. 3D-bilden är dynamisk genom att bilden av jorden är momentan som den ser ut nu, se vidare »[Tellus 3D Live](#)«.

Tips

Se »[SUN: Java 3D API](#)«.

Java 3D är beroende på en extension som är gjord med plattformsspecifik kod och man kan läsa »[Java extensioner med nativelib-filer](#)« för att se hur man gör så att denna fungerar på flera plattformar.

Notera att man med denna teknik även kan koppla tillbaka till serversidan (som äpplet laddades från) för att hämta dynamisk data för bildgenereringen.

Audio och video: Inspelning / redigering / konvertering Audio med MP3 och Ogg Vorbis

Allmänt

När det gäller format för audio och video finns för nuvarande många tävlande format och stödet för dessa finns oftast inte inbyggt i browsrarna utan man får använda speciella program för uppspelningen. Det finns tre program som är mycket vanliga för detta:

- RealPlayer för Linux/Unix, Win32 och Mac OS
- Quicktime för Win32 och Mac OS
- MS MediaPlayer för Win32

Dessa program kan användas både fristående och inbäddat i en XHTML-sida och ofta kan man även styra dem med JavaScript. Ett bra alternativ är att använda Java för uppspelning av audio och video.

Vi börjar i den här uppgiften med att titta på audio och kvaliteten beror främst på:

- Samplings-frekvensen
- Antalet bitar per sample
- Antalet kanaler

Se »[NIST: Digital Media File Types: Survey of Common Formats](#)« för en jämförelse mellan några av de vanligaste audio-formaten. Vi kommer att fokusera på formaten MP3, eftersom det är mycket vanligt, och Ogg Vorbis, eftersom det är ett nytt öppet fritt format som kommer att bli stort.

Uppgift

Använd programmet »[Audacity](#)« (eller andra bra program, exempelvis »[Pro Logic 7](#)« för Mac OS) för att:

- Redigera röst-audio som man spelar in
- Redigera musik som kommer från högkvalitativa ljud-filer så det blir en (cool) remix

Använd klipp och klistra, olika effekter, med mera. Spara sen filerna som:

- Format MP3 med olika kvalitet
- Format Ogg Vorbis med olika kvalitet

Gör ett XHTML 1.1-dokument med:

- Länkar till original-filerna (om dessa filer är väldigt stora behöver man inte redovisa dem)
- Länkar till de redigerade audio-filerna med olika format och kvalitet
- Beskrivning av format och kvalitet för varje länk

Om original-filerna är väldigt stora så behöver man inte redovisa dem eftersom det inte finns plats på DSV's filserver. Frivillig överkurs är att även titta på »[Speech](#)« som är en speciell codec för röst-audio.

Exempel

Ett [enkelt exempel](#) som inte är tillräckligt bra remixat och inte innehåller ett exempel med röst-audio.

Tips

De som arbetar på DSV behöver ta med sig hörlurar för att kunna göra denna och flera andra efterföljande uppgifter.

Audio och video: Inspelning / redigering / konvertering Video med Ogg Theora

Allmänt

Vi ska i den här uppgiften titta på video. Se »[NIST: Digital Media File Types: Survey of Common Formats](#)« för en jämförelse mellan några av de vanligaste video-formaten. Vi kommer att fokusera på formatet Ogg Theora eftersom det är ett nytt öppet fritt format som kommer att bli stort.

Uppgift

Använd programmet »[VirtualDub](#)« för Win32 eller »[Kino](#)« för Unix/Linux (eller andra bra program, exempelvis »[Adobe Premiere Pro 1.5](#)«) för att:

- Redigera en eller flera videor med klipp och klistra och eventuellt andra finesser
- Ta bort eventuell audio från videon och lägg till en från uppgiften [Audio och video: Inspelning/redigering/konvertering: Audio med MP3 och Ogg Vorbis](#) (eventuellt omredigerad) audio-fil som varar hela videons längd

Spara sen den redigerade video-filen och konvertera den till flera video-filer:

- Format Ogg Theora med olika kvalitet

Gör sen ett XHTML 1.1-dokument med:

- Länkar till original-filerna
- Länk till den ej konverterade men redigerade video-filen
- Länkar till de med olika kvalitet konverterade video-filerna
- Beskrivning av kvalitet för varje länk

Notera att de som använder Kino inte behöver konvertera utan kan spara direkt till rätt format. Om original-filerna är väldigt stora så behöver man inte redovisa dem eftersom det inte finns plats på DSV's filserver.

Exempel

Ett [enkelt exempel](#).

Tips

För konverteringen från AVI-format till Ogg Theora kan man använda Helix Producer 10 med plugin Xiph Producer Plugin:

```
producer.exe -i <filnamn>.avi -o <filnamn>.ogg -ad theora.rpad -dt
```

Läs readme-filen till Xiph Producer Plugin för mer info om hur man konverterar till olika kvalitet med mera.

De som arbetar på DSV kan man använda:

- »[FFMPEG](#)« som finns på [triton.dsv.su.se](#) (det går även att kompilera FFMPEG på Win32 men det är lite knepigt)
- »[FFMPEG2Theora](#)«

Audio och video: Strömmande via

RTSP / RTP / ISMA: Audio med MP3 spellistor

Allmänt

Vi ska i den här uppgiften titta på strömmande audio vilket är mycket vanligt och används av de många radiostationer som finns på Internet.

Uppgift

På DSV körs Darwin Streaming Server och en av utsändningarna är en MP3 spellista:

- http://atlas.dsv.su.se:8000/mp3_at_dsv

Gör ett XHTML 1.1-dokument med:

- En länk till ett egengjort m3u-dokument till denna MP3 spellista
- En länk till ett egengjort pls-dokument till denna MP3 spellista

De som arbetar hemma bör även installera och köra igång en egen Darwin Streaming Server och göra en egen MP3 spellista samt länka till denna på motsvarande sätt. Frivillig överkurs är att titta på strömmande servrar för Ogg Vorbis format, exempelvis »[IceCast](#)« och »[NetJuke](#)«.

Exempel

Ett [enkelt exempel](#).

Tips

Se [Darwin](#).

Audio och video: Strömmande via RTSP / RTP / ISMA: Audio med MP3 spellistor uppspelad med Java

Allmänt

Uppgiften belyser hur man kan använda Java för att spela upp strömmande audio.

Uppgift

Gör ett XHTML 1.1-dokument med ett Java-äpple som spelar upp MP3 spellistan från uppgiften [Audio och video: Strömmande media via RTSP/RTP/ISMA: Audio med MP3 spellistor](#).

Frivillig överkurs är att göra Java-äpplet själv, se exempelvis »[SUN: JMF MP3 Plugin](#)«. Notera att detta kan göras med JavaSound utan stöd av extensionen Java Media Framework (JMF).

Exempel

Några enkla exempel (man måste inte göra alla):

- [JLayer](#)
- [TinyPlayer](#)
- [jLGui](#)

Tips

Se exempelvis någon av följande:

- »[JavaZoom: JLayer](#)«
- »[JavaZoom: TinyPlayer](#)«
- »[JavaZoom: jLGui](#)«

Se [Darwin](#).

Audio och video: Strömmande via RTSP/RTP/ISMA: Video med MPEG-4 för stationära enheter

Allmänt

Uppgiften belyser hur man strömmar MPEG-4 för stationära enheter.

Uppgift

Konvertera den ej konverterade men redigerade video-filen från uppgiften [Audio och video: Inspelning/redigering/konvertering: Video med Ogg Theora](#) till en hintad MPEG-4-fil som följer ISMA-standard.

Lägg sen ut den konverterade video-filen på ett lämpligt ställe på ditt DSV/Unix/Linux-konto så att den är nåbar via DSV's Darwin Streaming Server.

Gör sen ett HTML-dokument med:

- En inbäddad RealPlayer Basic 10 som spelar upp video-filen
- En länk till den konverterade video-filen via RTSP
- En länk till den konverterade video-filen via HTTP

Man behöver i denna uppgift inte följa XHTML 1.1-standard eftersom stödet för object-taggen med RealPlayer inte ännu är helt implementerad.

Exempel

Ett [enkelt exempel](#) med några extra varianter av MPEG-4-filer.

Tips

Använd Helix Mobile Producer för konverteringen.

De som arbetar på DSV kan man använda:

- »[FFMPEG](#)« som finns på [triton.dsv.su.se](#)

Se [Darwin](#).

Audio och video: Strömmande via RTSP/RTP/ISMA: Video med MPEG-4 för stationära enheter uppspelad med Java

Allmänt

Uppgiften belyser hur man använder Java för att spela upp strömmad MPEG-4 för stationära enheter.

Uppgift

Konvertera den ej konverterade men redigerade video-filen från uppgiften [Audio och video: Inspelning/redigering/konvertering: Video med Ogg Theora](#) till en hintad MPEG-4-fil som följer ISMA-standarderna och som kan spelas upp av IBM's Java-äpple:

- »[IBM: IBM Toolkit for MPEG-4](#)«
- »[IBM Research: MPEG-4 Technologies](#)«

Lägg sen ut den konverterade video-filen på ett lämpligt ställe på ditt DSV/Unix/Linux-konto så att den är nåbar via DSV's Darwin Streaming Server.

Gör sen ett XHTML 1.1-dokument med:

- Ett Java-äpple som spelar upp video-filen strömmande från DSV's Darwin Streaming Server
- En länk till den konverterade video-filen via RTSP
- En länk till den konverterade video-filen via HTTP

Exempel

Ett [enkelt exempel](#).

Tips

Använd Helix Mobile Producer för konverteringen. Använd storleken 320 * 240.

De som arbetar på DSV kan man använda:

- »[FFMPEG](#)« som finns på [triton.dsv.su.se](#)

IBM's äpple har default en area för kommandon längst ner som tar upp 22 pixlar på höjden och har

man en film som är av storleken `<widd> * <höjd>` så ska alltså applet ha storleken `<widd> * (<höjd> + 22)`.

Se [Darwin](#).

Audio och video: Strömmande via RTSP/RTP/ISMA: Video med 3GPP för mobila enheter

Allmänt

Uppgiften belyser hur man strömmar 3GPP för mobila enheter.

Uppgift

Konvertera den ej konverterade men redigerade video-filen från uppgiften [Audio och video: Inspelning/redigering/konvertering: Video med Ogg Theora](#) till en hintad 3GPP-fil som följer ISMA-standard.

Lägg sen ut den konverterade video-filen på ett lämpligt ställe på ditt DSV/Unix/Linux-konto så att den är nåbar via DSV's Darwin Streaming Server.

Gör sen ett XHTML Basic-dokument med:

- En länk till den konverterade video-filen via RTSP
- En länk till den konverterade video-filen via HTTP

Exempel

Ett [enkelt exempel](#) med några extra varianter av 3GPP-filer.

Tips

Använd Helix Mobile Producer för konverteringen.

De som arbetar på DSV kan man använda:

- »[FFMPEG](#)« som finns på `triton.dsv.su.se`

Se [Darwin](#).

Audio och video: Semi-strömmande via HTTP-interleaving / muxing: Video med RealMedia uppspelad

med RealPlayer

Allmänt

Uppgiften belyser hur man semi-strömmar RealMedia-formatet som stöds på:

- Windows 95/98/NT/2000/XP
- Macintosh OS 8/OS 9/OS X
- Unic och Linux
- RealPlayer for Mobile Devices:
 - Pocket PC
 - Nokia 9200 Series
 - Nokia 7650
 - Palm OS 5 Based Handheld

När man semi-strömmar så sänds inte de strömmade filerna via en streaming server, som använder effektiva UDP, utan via en HTTP-server, som använder mindre effektiva TCP, och det går bara bra så länge man inte har för många simultana anslutningar. Man måste använda speciella format och speciella mediaspelare för att detta ska fungera.

Uppgift

Konvertera den ej konverterade men redigerade video-filen från uppgiften [Audio och video: Inspelning/redigering/konvertering: Video med Ogg Theora](#) till ett RealMedia-format och lägg ut den konverterade video-filen på ett lämpligt ställe på ditt DSV/Unix/Linux-konto så att den är nåbar via DSV's HTTP-server. Skapa även en ram-fil som pekar absolut på video-filen och lägg ut även den.

Gör sen ett HTML-dokument med:

- En inbäddad RealPlayer Basic 10 som spelar upp den konverterade video-filen (semi-strömmande genom att ladda ram-filen)
- En länk till video-filen via HTTP
- En länk till ram-filen via HTTP

Testa gärna med olika kvalitet på video-filen. Man behöver i denna uppgift inte följa XHTML 1.1-standarderna eftersom stödet för object-taggen med RealPlayer inte ännu är helt implementerad.

Exempel

Ett [enkelt exempel](#) med några extra varianter av semi-strömmade Ogg Theora-filer.

Tips

Använd RealProducer Basic 10 eller Helix Producer för konverteringen:

- Konvertera till en lämplig hastighet
- Använd single-stream

SureStream är ett adaptivt format som kan växla hastigheter under sändning beroende på hur bandbredden varierar men det formatet fungerar enbart med Real Streaming Server.

Man skapar en ram-fil (som är en vanlig text-fil) för att uppspelningen ska ske semi-strömmande, utan den så laddas hela rm-filen ner innan uppspelningen börjar. Vill man först testa lokalt på den

egna datorn så ska ram-filen peka på `file:///.<rm-fil>`.

Notera att RealProducer Basic 10 även finns för Unix/Linux och kan köras i kommandofönster. Man kan alltså lätt skapa HTTP-serversides program som utför konverteringar till RealMedia-format.

Audio och video: Semi-strömmande via HTTP-interleaving / muxing: Video med MPEG-4-MUX för stationära enheter uppspelad med Java

Allmänt

Uppgiften belyser hur man semi-strömmar MPEG-4.

Uppgift

Konvertera MPEG-4-filen från uppgiften [Audio och video: Strömmande via RTSP/RTP/ISMA: Video med MPEG-4 för stationära enheter uppspelad med Java](#) till formatet M4Mux som är ett standardbaserat interleave format för gradvis nerladdning under spelning som används över en HTTP-server och kan spelas upp av IBM's Java-äpple:

- »[IBM: IBM Toolkit for MPEG-4](#)«
- »[IBM Research: MPEG-4 Technologies](#)«

Lägg sen ut den konverterade video-filen på lämpligt ställe på ditt DSV/Unix/Linux-konto så att den är nåbar via DSV's HTTP-server.

Gör sen ett XHTML 1.1-dokument med:

- Ett Java-äpple som spelar upp den konverterade video-filen semi-strömmande från DSV's HTTP-server
- En länk till den konverterade video-filen via HTTP

Notera att M4Mux troligtvis enbart stöds av IBM's Java-mediaspelare.

Exempel

Ett [enkelt exempel](#).

Tips

Använd »[IBM: IBM Toolkit for MPEG-4](#)«. Man måste sätta sig in i detta verktyg ganska mycket för att klara uppgiften.

Audio och video: Integrering: SMIL

2.0 för stationära enheter

Allmänt

Uppgiften belyser hur man integrerar media med SMIL till en enhetlig framställning.

Uppgift

Gör ett SMIL 2.0-dokument som kombinerar:

- Texter (gärna RealText)
- Bilder
- Audio
- Video

Använd både:

- Sekventiell uppspelning
- Parallell uppspelning

Uppspelningen ska vara semi-strömmad.

Testa allt med W3C's validator.

Frivillig överkurs är att titta på `switch`-taggen för alternativ uppspelning för olika typer av enheter, bandbredd med mera.

Exempel

Ett exempel kommer troligtvis senare, man kan titta på »[Inria: Des exposés scientifiques multimédia](#)« tillsvidare.

Tips

Använd RealPlayer Basic 10 för uppspelningen och en ram-fil för att få uppspelningen semi-strömmad. Om man vill att uppspelningen ska starta över hela fönstret så skriver man följande i ram-filen:

```
http://<server>/<sökväg>/<filnamn.smil>?mode="theater"
```

Använd RealMedia-format för audio och video.

Audio och video: Integrering: SMIL 2.0 för stationära enheter uppspelad med Java

Allmänt

Uppgiften belyser hur man använder Java för att spela upp SMIL.

Uppgift

Kopiera och modifiera SMIL-filen i uppgiften [Audio och video: Integrering av media: SMIL 2.0 för stationära enheter](#) så att den blir på version SMIL 1.0.

Gör sen ett XHTML 1.1-dokument med ett Java-äpple som spelar upp en denna SMIL-bild.

Testa allt med W3C's validator.

Exempel

Inget exempel ges.

Tips

Se exempelvis »[W3C: Synchronized Multimedia](#)« för att hitta lämpliga Java-tillbehör.

Notera att en Java-baserad SMIL-spelare ofta inte stöder samma media-format som RealPlayer Basic 10 vilket medför att man då måste konvertera även media-filerna så de blir på format som stöds.

Audio och video: Integrering: SMIL i MMS för mobila enheter

Allmänt

Uppgiften belyser hur MMS förhåller sig till SMIL.

Uppgift

Diskutera relationen mellan SMIL och MMS. Visa gärna med exempel.

Exempel

Inget exempel ges.

Tips

Inga tips ges.

Dynamiska dokument: Objektmodellen: Egenskaper

Allmänt

Vi ska nu titta på objekten i objekt modellen (OM) som ligger mellan browsern och JavaScript:

- window
 - window.screen
 - window.navigator
 - window.location
 - window.history
 - window.document

Av dessa är enbart `window.document` standardiserat av W3C i DOM 2. De övriga objekten är inte standardiserade än så därför tittar vi i de tre första uppgifterna enbart på de (1) egenskaper, (2) händelser och (3) metoder som är gemensamma för Firefox 1+, IE 4+ med flera, för dessa objekt.

Uppgiften belyser hur man (1) läser av värdet (och för vissa även sätter värdet) hos vissa egenskaper hos objekt-modell-objekten och (2) för in dessa värden i ett XHTML-dokument.

Uppgift

Gör ett XHTML 1.1-dokument med ett JavaScript som visar de egenskaper som finns angivna för respektive objekt-modell-objekt nedan:

- window-objektet:
 - Fönstrets namn (`window.name`)
 - Fönstrets default status-meddelande (`window.defaultStatus`)
 - Fönstrets status-meddelande (`window.status`)
 - Fönstrets öppnare (`window.opener`)
 - Fönstrets förälder (`window.parent`)
 - Fönstrets topnivå förälder (`window.top`)
 - Fönstret är stängt (`window.closed`)
- window.screen-objektet:
 - Skärmens bredd (`window.screen.width`)
 - Skärmens höjd (`window.screen.height`)
 - Skärmens färgdjup (`window.screen.colorDepth`)
 - Skärmens tillgängliga bredd (`window.screen.availWidth`)
 - Skärmens tillgängliga höjd (`window.screen.availHeight`)
- window.navigator-objektet:
 - Browserns total-info (`window.navigator.userAgent`)
 - Browserns namn (`window.navigator.appName`)
 - Browserns kodnamn (`window.navigator.appCodeName`)
 - Browserns version (`window.navigator.appVersion`)
 - Browsern tillåter cookies (`window.navigator.cookieEnabled`)
 - Datorns plattform (`window.navigator.platform`)
- window.location-objektet:
 - Browserns URL (`window.location.href`)
 - Browserns URL/protokoll (`window.location.protocol`)
 - Browserns URL/hostnamn (`window.location.hostname`)
 - Browserns URL/host (`window.location.host`)
 - Browserns URL/port (`window.location.port`)
 - Browserns URL/sökväg (`window.location.pathname`)
 - Browserns URL/bokmärke (`window.location.hash`)
 - Browserns URL/söksträng (`window.location.search`)
- window.history-objektet:
 - Historyns längd (`window.history.length`)
- window.document-objektet:
 - Dokumentets kakor (`window.document.cookie`)
 - Dokumentets referens (`window.document.referrer`)

- Dokumentets säkerhets-domän (`window.document.domain`)
- Dokumentet modifierades (`window.document.lastModified`)

Ge följande egenskaper ett värde innan avläsning:

- Fönstrets default status-meddelande
- Fönstrets status-meddelande
- Kakans värde

Exempel

En [bild av ett enkelt exempel](#) som en statisk PDF-fil. Anledningen till att inget riktigt exempel ges är att man då får hela lösningen på uppgiften.

Tips

Man kan exempelvis göra så här:

```
var textNode = document.createTextNode(window.name);
document.getElementById("v01").appendChild(textNode);
```

Dynamiska dokument: Objekt-modellen: Händelser

Allmänt

Uppgiften belyser hur man fångar och ta hand om händelser hos objekt-modell-objekt.

Uppgift

Gör ett XHTML 1.1-dokument med ett JavaScript som fångar de händelser som finns angivna för respektive objekt-modell-objekt nedan:

- window-objektet:
 - load
 - unload
 - resize
- window.document-objektet:
 - click

För varje händelse ska en dialog-ruta visas som visar vilken händelse som genererats och fångats. För händelsen `click` ska även muspekarens koordinater visas.

Exempel

Några statiska bilder av:

- [Ett enkelt exempel](#) som en statisk PDF-fil
- [Ett enkelt exempel](#) som en statisk GIF-bild
- [Ett enkelt exempel](#) som en statisk GIF-bild
- [Ett enkelt exempel](#) som en statisk GIF-bild

Anledningen till att inget riktigt exempel ges är att man då får hela lösningen på uppgiften.

Tips

Inga tips ges.

Dynamiska dokument: Objekt-modellen: Metoder

Allmänt

Uppgiften belyser hur man anropar metoder hos flera viktiga objekt-modell-objekt.

Uppgift

Gör ett XHTML 1.1-dokument med ett JavaScript som implementerar de metodanrop som finns angivna för respektive objekt-modell-objekt nedan:

- window-objektet:
 - Öppna en dialog-ruta med en godtycklig text och en ok-knapp
 - Öppna en dialog-ruta med en godtycklig text, en ok-knapp och en avbryt-knapp
 - Öppna en dialog-ruta med en godtycklig text, ett inmatnings-text-fält, en avbryt-knapp och en ok-knapp
 - Flytta fönstret relativt
 - Flytta fönstret absolut
 - Ändra fönstrets storlek relativt
 - Ändra fönstrets storlek absolut
 - Skrolla relativt
 - Skrolla absolut
 - Öppna ett nytt fönster
 - Stäng detta fönstret
 - Kör ett skript var 5:e sekund
 - Stäng av detta skript
 - Kör ett skript en gång om 5 sekunder
 - Stäng av detta skript
- window.navigator-objektet:
 - Java-enablad
- window.location-objektet:
 - Surfa vidare till ett annat ställe
 - Ladda om sidan
- window.history-objektet:
 - Surfa bakåt
 - Surfa framåt
 - Surfa bakåt/framåt

Exempel

En [bild av ett enkelt exempel](#) som en statisk PDF-fil. Anledningen till att inget riktigt exempel ges är att man då får hela lösningen på uppgiften.

Tips

Inga tips ges.

Dynamiska dokument: Dokument-objekt-modellen: Dynamisk struktur

Allmänt

Uppgiften går ut på att dynamiskt ändra strukturen i ett XHTML 1.1-dokument.

Uppgift

Gör ett XHTML 1.1-dokument med ett JavaScript som förändrar strukturen på ett XHTML 1.1-dokument dynamiskt:

- Användarstyrt
- Programstyrt

Exempel

Några bilder av ett enkelt exempel:

- [Bild 1](#)
- [Bild 2](#)
- [Bild 3](#)

Anledningen till att inget riktigt exempel ges är att man då får hela lösningen på uppgiften.

Tips

Ett program är användarstyrt när förändringen initieras av användaren, exempelvis:

- När muspekaren är över ett specifikt område så händer en förändring
- När muspekaren går ifrån ett specifikt område så händer en förändring
- När användaren klickar på ett specifikt område så händer en förändring
- När användaren släpper musknappen efter ett klick på ett specifikt område så händer en förändring

Ett program är programstyrt när förändringen initieras av programmet, exempelvis med någon av metoderna:

- `window.setInterval`
- `window.setTimeout`

Man kan exempelvis ändra lägga till och/eller ta bort text-noder och då är följande metoder användbara:

- `document.createTextNode("En text");`
- `document.getElementById('x').appendChild(textNode);`
- `document.getElementById('x').removeChild(textNode);`

Dynamiska dokument: Dokument-objekt-modellen: Dynamisk stil

Allmänt

Uppgiften går ut på att dynamiskt ändra stil på ett XHTML 1.1-dokument.

Uppgift

Gör ett XHTML 1.1-dokument med en CSS 2.1-fil och ett JavaScript som förändrar stilsättningen på ett XHTML 1.1-dokument dynamiskt:

- Användarstyrt
- Programstyrt

Exempel

Några bilder av ett enkelt exempel:

- [Bild 1](#)
- [Bild 2](#)
- [Bild 3](#)

Anledningen till att inget riktigt exempel ges är att man då får hela lösningen på uppgiften.

Tips

Ange inte stilen direkt i JavaScript-koden utan använd egenskapen `className` och växla på så sätt mellan olika stilar som är fördefinierade i CSS 2.1-filen.

Dynamiska dokument: Dokument-objekt-modellen: Dynamiskt innehåll

Allmänt

Uppgiften går ut på att dynamiskt ändra innehåll i ett XHTML 1.1-dokument.

Uppgift

Gör ett XHTML 1.1-dokument med ett JavaScript som förändrar innehållet i ett XHTML 1.1-dokument dynamiskt:

- Användarstyrt
- Programstyrt

Exempel

Några bilder av ett enkelt exempel:

- [Bild 1](#)
- [Bild 2](#)

Anledningen till att inget riktigt exempel ges är att man då får hela lösningen på uppgiften.

Tips

Man kan exempelvis förändrar innehållet:

- Användarstyrt genom att olika bilder visas beroende på om användaren har muspekaren över en bild eller ej
- Programstyrt genom att man visar aktuell tid (med hjälp av objektet `Date`)

Dynamiska dokument: Dokument-objekt-modellen: Formulär validering

Allmänt

Uppgiften belyser HTTP-klientsides validering av indata till formulär.

Uppgift

Gör ett XHTML 1.1-dokument med ett JavaScript som kontrollerar indata till ett formulär som ska ta emot:

- En e-postadress som följer ett viss mönster, exempelvis sluta på `dsv.su.se` eller `kth.su.se`
- Ett lösenord som ska vara ett tal

En dialog-box ska visa om data är på korrekt eller felaktig form.

Exempel

Några bilder av ett enkelt exempel:

- [Bild 1](#)
- [Bild 2](#)
- [Bild 3](#)
- [Bild 4](#)
- [Bild 5](#)
- [Bild 6](#)

Anledningen till att inget riktigt exempel ges är att man då får hela lösningen på uppgiften.

Tips

Använd reguljära uttryck för kontroll av e-postadressen.

Observera att kontroll av indata till ett formulär enbart är till för att underlätta för användaren, inte för att kontrollera att indata är säkert för programmet på HTTP-serversidan eftersom det är mycket lätt att kringgå sådana kontroller (genom att exempelvis använda en browser som inte stöder JavaScript).

Dynamiska dokument: Dokument-objekt-modellen: JavaScript-Java kopplingar

Allmänt

Vi ska nu se hur JavaScript kan få kontakt med ett Java-äpple, något som kan vara mycket användbart inom områden där JavaScript saknar stöd, exempelvis inom grafik och nätverkskopplingar. Uppgiften belyser kopplingen mellan JavaScript och Java-äpplen.

Uppgift

Gör ett XHTML 1.1-dokument med ett JavaScript och ett Java-äpple där JavaScriptet anropar (användarstyrt och/eller programstyrt) ett Java-äpple som i sin tur gör något och svarar med ett värde som sen JavaScriptet visar i XHTML 1.1-dokumentet.

Frivillig överkurs är att även titta på hur Java initierar anrop till JavaScript, se vidare »[SUN: Java-to-Javascript Communication](#)«.

Exempel

En [bild av ett enkelt exempel](#). Anledningen till att inget riktigt exempel ges är att man då får hela lösningen på uppgiften.

Tips

Ge äpplet ett id i XHTML-dokumentet och använd detta id i JavaScript-koden för att nå äpplet.

Stäng eventuellt alla öppna browsrar vid omtestning så att Java-äpplet laddas om.

Dynamiska dokument: Dokument-objekt-modellen: Databindning

Allmänt

Databindning är en bra teknik som väl avvägd kan göra att användaren:

- Får snabbare interaktivitet
- Kan minska ner antalet förfrågningar till HTTP-servern

Det är bra att ha data på XML-format och vi ska titta på hur man kan accessa XML-data från JavaScript genom att använda den i Firefox 1+ inbyggda XML/XSLT-processorn. Ett annat alternativ är att använda ett Java-äpple som kan parsas och processa ett XML-dokument och anropa detta äpple från JavaScript enligt uppgiften [Dynamiska dokument: Dokument-objekt-modellen: JavaScript-Java kopplingar](#). Uppgiften går ut på att visa hur man använder databindning där datat som binds är på XML-format.

Uppgift

Gör ett XHTML 1.1-dokument med ett JavaScript som läser in och (användarstyrt) visar en viss del av innehållet i ett egenskapat XML-dokument. Användaren ska kunna välja vilken del av XML-datat som ska visas. Implementera även en kontroll så att användaren inte kan accessa data som inte finns.

Exempel

En [bild av ett enkelt exempel](#). Anledningen till att inget riktigt exempel ges är att man då får hela lösningen på uppgiften.

Tips

Kontrollera att XML-dokumentet är välutformat.

DOM 2's modell för att ladda ett XML-dokument:

```
dtd = document.implementation.createDocumentType('my_root_element', null, 'my_dtd.dtd');
xml = document.implementation.createDocument("", "", dtd);
xml.async = false;
xml.load('my_xml.xml');
```

Notera att IE (vilken vi inte använder på kursen men kan vara bra att veta) inte stöder detta enligt DOM 2 men kan göras med:

```
xml = new ActiveXObject("Microsoft.XMLDOM");
xml.async = false;
xml.validateOnParse = true;
xml.load('my_xml.xml');
```

Gesällprov

Allmänt

Gesällprovet är en uppgift för den egna kreativiteten och är ett fritt valt (men obligatoriskt) arbete där du får chansen att visa vad du går för konstruktionsmässigt (istället för minnesmässigt som på en tenta) i "lugn och ro".

Uppgift

Följande krav måste uppfyllas:

- Ska baseras på de tekniker som kursen bygger på men får givetvis även innehålla tekniker från andra områden
- Ska vara nätverksbaserat
- Ska vara bra både till både form och funktion
- Ska inte vara en av de tidigare uppgifterna på kursen men gärna en utökning av en eller flera

av dessa

Tips

Huvudsaken är att man gör något som är kul! Några exempel:

- XHTML/CSS- eller SMIL-tutorial för något eller några av kursens ämnen eller något ämne inom HTTP-klientsidan som inte tas upp på kursen
- MPEG-4 Systems
- DHTML-program där JavaScript anropar ett Java-äpple som anropar ett HTTP-serversides program som anropar ett annat ställe för data, exempelvis värdet på någon aktie
- DHTML-grafikdemo med rörlig grafik gjord av bilder uppbyggda av 1*1 pixel stora bilder
- DHTML-spel
- Egengjord Java-mediaspelare för Ogg Vorbis och Ogg Theora eller några andra format, se »[SUN: Java Media Framework API \(JMF\)](#)«, eventuellt med funktioner för RTSP/RTP
- Egengjord minimal audio och/eller video standard och tillhörande strömmande mediaspelare och mediaserver

Hjälp

På denna sida finns alla hjälpresurser på kursen.

Handledning

Kör man fast så får man hjälp via news och e-post. Notera dock att det inte är någon handledning på:

- Kunskap som ingår i förkunskapskraven (kompileringsfel med mera)
- Gesällprovet

News är tänkt som en plattform för diskussion mellan studenterna men handledarna kommer att besvara de frågor som fortfarande är obesvarade senare under de dagar som det är handledning. E-post går direkt till handledarna och de kommer att försöka svara snabbt men eftersom de ibland får väldigt många frågor kan svaren dröja. Notera att handledarna enbart arbetar inom angivna handledningstider vilket medför att man kanske inte hinner få svar samma dag på en sent ställd fråga.

News

News bör användas för frågor som kan intressera andra:

- Läs först igenom de tidigare frågorna och svaren
- Hittar man inte en lösning så:
 - Ange ett väl beskrivande ämne för brevet
 - Var mycket tydlig vid beskrivningen av problemet
 - Sänd inte in mycket kod eller fullständiga program

News-server:

- Adress: `atlas.dsv.su.se`
- Port: 1199 (notera att detta inte är default numret och därför måste sättas manuellt)

News-grupper:

- `public.ip3.a.1` för frågor om uppgift 1

- public.ip3.a.2 för frågor om uppgift 2
- ... och så vidare ...
- public.ip3.misc för övriga frågor

Se även [news](#) för mer detaljerad information om hur man konfigurerar en news-klient.

E-post

E-post bör användas för specifika frågor utan intresse för andra:

- Sänd inte e-post direkt till lärarna utan använd länken nedan
- Om man ska sända med kod:
 - Sänd endast med de delar som är relevanta för frågan
 - Koden ska vara välkommenterad och välstrukturerad
- Använd "svara till alla" på eventuella följdfrågor

E-post-server:

- Ingående e-brev (POP3):
 - Adress: mail.dsv.su.se
 - Port: 110
- Utgående e-brev (SMTP):
 - Adress: mail.dsv.su.se
 - Port: 25

Notera att om man sitter uppkopplad utanför DSV's domän (exempelvis via Telia Bredband) så ska man använda leverantörers SMTP-server för utgående brev.

E-post länk:

- [Klicka här för handledning via e-post](#)

Se även [e-post](#) för mer detaljerad information om hur man konfigurerar en e-post-klient.

FAQ

Här följer de frågor och svar som brukar vara vanligast:

HTTP-klientsidan

Fråga/svar 1

W3C's CSS-validator skiljer inte på CSS 2 och CSS Mobile Profile 1.0? Vad gör man då?

W3C's validator för CSS Mobile Profile 1.0 är bristfällig. Den som har möjlighet kan istället testa »[TopStyle Pro](#)« som är en mycket bra editor för CSS.

Fråga/svar 2

Hur ska de inledande raderna i ett XHTML 1.1 dokument se ut?

Så här ser alla XHTML 1.1 dokument ut i början:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
```

```

<head>
  <meta http-equiv="Content-Type" content="application/xhtml+xml; charset=iso-8859-1"/>
  <title>
    Min titel
  </title>
  <link type="text/css" href="min_css_fil.css" rel="stylesheet"/>
</head>

```

Man kan givetvis byta ut värdet i title-taggen, länka till ett annat CSS eller inte till något CSS alls.

För XHTML Basic 1.0 så ska man använda en liknande konstruktion: »[W3C: XHTML Basic](#)«.

Fråga/svar 3

W3C's XHTML-validator krånglar när man använder svenska tecken, vad ska jag göra?

Kryssa i Encoding: iso-8859-1.

Fråga/svar 4

Hur kör man Java-äpplen med Firefox och XHTML?

Man gör enligt följande (här är X.class startklassen och X.jar jar-filen):

```

<object classid="java:X.class" width="640" height="480">
  <param name="type" value="application/x-java-applet;version=1.5"/>
  <param name="code" value="X.class"/>
  <param name="archive" value="X.jar"/>
  <param name="codebase" value="."/>
</object>

```

Som en () kan nämnas att man med IE gör på ett annat sätt (som vi inte använder på kursen):

```

<object classid="clsid:8AD9C840-044E-11D1-B3E9-00805F499D93" codebase="http://java.sun.com/update/1.5."
  <param name="code" value="X.class"/>
  <param name="archive" value="X.jar"/>
  <param name="type" value="application/x-java-applet;version=1.5"/>
  <param name="scriptable" value="false"/>
</object>

```

Fråga/svar 6

Hur använder man dynamiska sidor i IE som inte stöder DOM 2 fullt ut?

Man gör ett JavaScript, exempelvis [patch.js](#), som laddas före andra JavaScript och definierar i detta JavaScript meta-metoder som för de delar som inte stöds av IE och använder sedan dessa metoder i de efterföljande JavaScripten.

Fråga/svar 7

Kan man dölja XHTML, CSS och JavaScript för användaren?

Man får då använda browserspecifika lösningar. I exempelvis IE så kan man kryptera JavaScript med programmet [screnc.exe](#). Detta krypterar enbart JavaScript men om man även vill dölja XHTML-dokumentet och CSS-filen så kan man generera hela XHTML-dokumentet med JavaScript-metoden document.write. Detta kan vara omständigt att göra manuellt så man kan exempelvis göra ett minimalt Perl-program som gör det:

```

#!/usr/local/bin/perl

open(F1, "<x.html");
open(F2, ">x.js");

```

```
while($row = <F1>) {
    chomp $row;
    print F2 'document.write(\'\' . $row . '\');' . "\n";
}

close F1;
close F2;

system("screnc x.js x_enc.js");
```

Detta förutsätter att man kör Win32 med programmet [screnc.exe](#) i samma katalog.

Fråga/svar 8

Kan man göra ett JavaScript som automatiskt visar alla länkar som går utanför webbplatsen med en annan stilsättning och även öppnar dessa länkar i ett nytt fönster trots att man använder XHTML 1.1?

Det går, se sidan »[Pierre A. I. Wijkman](#)« som har den funktionen med:

- [patch.js](#)
- [window_opener.js](#)

Fråga/svar 9

Kan man göra ramsidor med CSS som även fungerar i IE?

Det går, se sidan »[OpenLab](#)«.

Fråga/svar 10

Hur kompilerar man FFMPEG för Win32?

Man gör enligt följande:

1. Skapa ett admin-konto som inte har några mellanslag i namnet
2. Logga in på det kontot
3. Installera »[MinGW](#)«
4. Installera »[MSYS](#)« (svara J på frågan om post installation)
5. Installera »[zlib](#)«:
 - Lägg filen under: <sökväg>\msys\1.0\home\<användarnamn>\
 - Starta MSYS-miljön
 - Packa upp: tar xvzf zlib-1.2.1.tar.gz
 - Gå till katalogen: cd zlib-1.2.1
 - Konfigurera: ./configure
 - Kompilera och bygg: make
 - Installera zlib-bibloteken: cp libz.a /mingw/lib
 - Installera zlib-headers: cp zlib.h zconf.h /mingw/include
6. Installera »[FFMPEG](#)«:
 - Lägg filen under <sökväg>\msys\1.0\home\<användarnamn>\
 - Starta MSYS-miljön
 - Packa upp: tar xvzf ffmpeg-0.4.9-pre1.tar.gz
 - Gå till katalogen: cd ffmpeg-0.4.9-pre1
 - Konfigurera: ./configure
 - Kompilera och bygg: make

Notera att man bör enabla flera olika extra tillbehör till FFMPEG för att kunna konvertera mellan många olika format och det måste man göra med configure, ett urval är:

```
./configure --enable-mp3lame --enable-vorbis --enable-ming32 --enable-gpl
```

Se vidare ./configure --help. Vissa av dessa tillbehör kräver separata nerladdningar och installationer innan man installerar FFMPEG.

E-post

På denna sida beskrivs hur man konfigurerar e-post-klienten. Följande är ett exempel på hur man konfigurerar e-post i Outlook Express, förfarandet är ungefär detsamma i andra moderna e-post/news-klienter:

- Klicka på Verktyg/Konton
- Klicka på Add/Mail
- Skriv in namn och e-post-adress
- Skriv in adressen för inkommande och utgående e-post-servrar
- Skriv in användarnamn och lösenord
- Klicka på Avsluta

Vill man använda exempelvis hotmail istället så kan man göra följande för att få all e-post som kommer till DSV vidareänd till hotmail-kontot:

- telnet triton.dsv.su.se
- emacs .forward
- Skriv in hotmail-adressen, spara och avsluta emacs
- Logga ut

De som använder ett e-postkonto på DSV kan även använda »[DSV: Webmail](#)«.

News

På denna sida beskrivs hur man konfigurerar en news-klient. Följande är ett exempel på hur man konfigurerar news i Outlook Express, förfarandet är ungefär detsamma i andra moderna e-post/news-klienter:

- Klicka på Verktyg/Konton
- Klicka på Lägg till/News
- Skriv in namn och e-post-adress
- Ange adressen för news-servern
- Klicka på Avsluta
- Klicka på mappen Alla och markera aktuell news-server
- Klicka på Egenskaper
- Klicka på Avancerat
- Ange port nummer till news-servern
- Klicka på ok och sen Stäng
- Svara Ja på frågan Vill du ladda ner nyhets-grupper från news-kontot
- Dubbelklicka på alla grupper som är av intresse
- Klicka på ok

Default så visar Outlook Express bara de senaste meddelandena. Vill man ändra på det så:

- Klicka på Verktyg/Alternativ
- Klicka på mappen Underhåll
- Ändra vid Ta bort news meddelanden efter 5 dagar (exempelvis genom att ta bort markeringen helt)

Anledningen till att vi använder news istället för FirstClass är att news är en öppen mycket vanlig standard på Internet och att det även ingår som några förslag på gesällprov.

Redovisning i detalj

Denna beskrivning utgår från att man använder DSV's datorer men man kan använda andra ställen än DSV för redovisningen. Notera dock att andra ställen kan använda en annan huvudkatalog för webpublicering och andra rättigheter på kataloger och filer.

Steg för steg

1. Starta ett [filöverföringsprogram](#) och koppla dig till ditt DSV/Unix/Linux-konto
2. Skapa katalogstrukturen nedan och ladda upp filerna i denna struktur
3. Sätt rättigheterna på kataloger och filer så att allt blir läsbart från en browser
4. Testa att allt kan nås från en browser och att all källkod ser bra ut från en browser
5. Ange på redovisningssidan:
 - o URL'en till denna katalogstruktur: `http://www.dsv.su.se/~x/ip3/`, där x är användarnamnet
 - o Att allt är klart för redovisning

Katalogstruktur

Man måste ha **exakt följande katalogstruktur** men använder man en annan plats än DSV så måste inte www katalogen ha det namnet.

```
www
  ip3
    ip3.zip (ip3 zippad, ej innehållande ip3.zip givetvis)
    1.1
      källkodsfiler för uppgift 1.1
    1.2
      källkodsfiler för uppgift 1.2
    1.3
      källkodsfiler för uppgift 1.3
    1.4
      källkodsfiler för uppgift 1.4

    ... osv ...

  7
    källkodsfiler för gesällprovet
    description
      beskrivning i text och bild av gesällprovet
```

Observera:

- o Katalogen www måste ha stora/versala bokstäver
- o Skapa inte kataloger för frivilliga uppgifter som inte är gjorda
- o Lägg ingen fil med namnet `index.html` eller `index.htm` direkt i uppgiftskatalogerna eftersom man då inte kan lista innehållet i dessa kataloger från Internet

Beskrivning av gesällprovet

Gesällprovet ska beskrivas generellt med både text och bild (på programmet under körning) på något av följande sätt:

- o HTML-dokument med bilderna inkluderade
- o PDF-dokument med bilderna inkluderade
- o TXT-dokument med bilderna separata

Ett HTML-dokument kan göras antingen för hand i en vanlig text-editor eller med ett högnivåverktyg.

För att skapa PDF-filer i Win32 kan man exempelvis:

1. Lägg till en skrivare, exempelvis HP Color LaserJet PS
2. Högerklicka på skrivaren och välj Egenskaper, klicka på Portar och kryssa i FILE
3. Skriv ut Word-dokumentet till denna skrivare varefter en fil <namn>.prn skapas
4. Gå till »PS2PDF.com« och konvertera filen till PDF

En bild (använd inte BMP-format) av ett program under körning fås enligt:

- Win32: Tryck på <CTRL><ALT><PRINT SCR> samtidigt, vilket medför att bilden på programmet läggs i urklippsbufferten som sedan lätt kan klistras in i ett HTML-dokument i en HTML-editor
- Linux/Unix: Skriv i kommandofönstret `import dump.jpeg` och klicka sen på det fönster som man vill ha en bild på (ändrar man till en annan extension så får man bilden som en annan bildtyp)

All eventuell java-kod i gesällprovet ska beskrivas med verktyget javadoc.

Efter erhållet betyg

När man fått G/VG eller 3/4/5 på ett arbete så bör man ändra rättigheterna på källkodsfilerna så att ingen annan kan läsa dem.

Nyheter

På denna sida visas alla uppdateringar.

Tid	Rum	Uppdatering
2005-01-17 06:00	Allt	Allt klart utom uppgift 6 och exemplen
2005-01-18 23:10	Uppgifter	Uppgift 6 online
2005-02-03 23:00	Uppgifter	Exemplen online (har lagts till gradvist sedan kursen startades)
2005-02-14 00:30	Schema	Tider för deadlines + tentor
2005-03-09 21:30	Schema	Extra deadline + tenta

Internetprogrammering IV

På denna sida finns hela kursen utom utom OH-bilderna och exemplen.

Intro

Välkommen Pierre! På denna sida beskrivs kursen generellt.

Kurskoder

Kursen är på 5 poäng och går för följande universitet/högskolor:

- SU med kod IP4
- Nätuniversitetet med kod IB9050

Kursen ges även i en utökad fördjupande form, kallad IP4+, på 5 + 2 poäng för följande universitet/högskolor:

- SU med kod IP4+

Notera att man **inte senare kan komplettera** från IP4 till IP4+ utan att man måste välja IP4+ direkt från start.

Innehåll

Denna kurs lär ut Internetprogrammering fördjupat mot XML-baserade distribuerade tekniker för både stationära och mobila enheter:

- Extensible Markup Language (XML):
 - Intro:
 - Kärnkoncept
 - Namnrymder
 - Entiteter
 - Dokumentmodeller:
 - Document Type Definition (DTD)
 - XML Schema
 - Processande:
 - Händelse-baserat med Simple API for XML (SAX)
 - Träd-baserat med Document Object Model (DOM)
 - Högnivå-baserat med Java API for XML Processing (JAXP) och JDOM
- Distribuerade processer:
 - Remote Method Invocation (RMI):
 - Klientsidan
 - Serversidan
 - Dynamisk laddning av klasser
 - Callbacks
 - Säker kommunikation
 - XML Remote Procedure Call (XML-RPC) och Java API for XML-based RPC (JAX-RPC):
 - Klientsidan
 - Serversidan
 - Säker kommunikation
 - Web Services (WS) med Simple Object Access Protocol (SOAP) och Web Service Definition Language (WSDL):
 - Klientsidan
 - Serversidan
 - Säker kommunikation
- Mobila enheter med Java Micro Edition (J2ME) med Connected Limited Device Configuration (CLDC) 1.1 och Mobile Information Device Profile (MIDP) 2.0:
 - Intro:
 - Grafiska gränssnitt och händelsehantering
 - Lagring med Record Management System (RMS)
 - Nätverkskopplingar:
 - Stream sockets på klientsidan
 - Stream sockets på serversidan
 - Datagram sockets
 - XML-baserade distribuerade processer:
 - XML
 - XML-RPC
 - Web Services med SOAP och WSDL
 - Extensioner:
 - Wireless Messaging API (WMA) 2.0
 - Mobile Media API (MMAPI) 1.1
 - Mobile 3D Graphics API för J2ME
 - J2ME Web Services Specification
 - Java APIs för Bluetooth
 - PDA Optional Packages för J2ME

Språk: Java, XML.

Undervisningsform

Pedagogiken för denna kurs är:

- (1) Frihet, (2) tävling och (3) samarbete --> glädje och motivation --> ett bra resultat!

Frihet, tävling och samarbete gör att man går från passivt mottagande till aktivt sökande. Genom frihet tillåts man själv bli nyfiken och ställa frågor innan man får svar. Genom tävling motiveras man att göra sitt bästa. Detta leder fram till glädje och motivation och därmed till ett bra resultat.

Observera! Detta är en distanskurs och kräver förmåga till självständigt arbete. Kursen har dock, till skillnad mot vanliga distanskurser, kontinuerligt och direkt stöd via handledning på distans. Kursen är krävande och man bör inte läsa andra kurser samtidigt. Kursen fokuserar starkt på praktiskt arbete och examinationen bygger på detta arbete. Digitalslöjd!

Examination

Examinationen på kursen består av:

- Praktiska uppgifter, 4 poäng (R/G/VG) (R/3/4/5):
 - Flera bestämda/styrda
 - Ett fritt valt gesällprov
- Tentamen, 1 poäng (U/G) (U/3)

För fördjupningskursen gäller utöver ovanstående även:

- Extra praktiska uppgifter, 2 poäng (R/G) (R/3):
 - Flera bestämda/styrda

Totalbetyget på båda kurserna bestäms av de praktiska uppgifterna med tyngdpunkt på det fritt valda gesällprovet.

Notera att man måste lämna in lösningar på de praktiska uppgifterna som stämmer med **nuvarande läsårs praktiska uppgifter**. Kurserna utvecklas dock bara i små steg med gradvisa förfiningar så det innebär inte att man måste göra om allt om man inte har fått betyg på de praktiska uppgifterna vid den sista deadline i Augusti.

Litteratur

Litteraturen är:

- »[Java & XML \(andra utgåvan\)](#)« skriven av McLaughlin och utgiven av »[O'Reilly](#)«

För de som går IP4+ behövs även:

- »[Wireless Java](#)« skriven av Mahmoud och utgiven av »[O'Reilly](#)«

Billigaste boken hittas med hjälp av »[Bokfynd.nu](#)«. Vi rekommenderar »[Amazon](#)« och »[WHSmith](#)« i England som brukar ha bra priser och snabba leveranser. Köp inte böcker utanför EU eftersom det tar lång tid (2-3 veckor) att få böckerna och sen får man en ganska stor extra avgift från tullen.

4 steg för att klara denna kurs

Gör följande för att klara denna kurs:

1. Registrera dig hos »[DSV: Studievägledningen](#)« som har e-post svl@dsv.su.se
2. Köp och läs kurslitteraturen
3. Lös och redovisa uppgifterna
4. Anmäl dig till tentamen (minst en vecka innan den går) och klara denna

Lärare

Vi önskar alla en givande och rolig kurs!

- »[Pierre A. I. Wijkman](#)«, ansvarig
- Handledarna

Lycka till!

Schema

På denna sida finns alla tider och platser för kursen. Alla tider är preliminära och kan ändras.

Föreläsningar

Detta är en distanskurs och vi ger inte en hel föreläsningsserie. Den inledande föreläsningen fungerar enbart som en kick-off eller frösättning. Notera att innehåll markerat som reserv innebär att ingen föreläsning ges men att detta kan ändras vid behov.

Tid	Rum	Föreläsare	Innehåll
2005-03-29 kl 13-17	Sal A, Forum, Kista	Pierre A. I. Wijkman	Intro, XML

Handledning

All handledning sker via e-post och news (se sidan Hjälp).

Samtliga studenter har handledning:

- Kursens 1:a - 5:e vecka: samtliga vardagar kl 10-17

Studenter som läser +-kursen (7 poäng) har även handledning på +-delen:

- Kursens 6:e - 7:e vecka: samtliga vardagar kl 10-17

Studenter från Nätuniversitetet som läser på halvfart har även handledning:

- Kursens 6:a - 10:e vecka: samtliga vardagar kl 10-17

För de som arbetar på DSV finns info om prio-tider för labbsalarna under »[DSV: Scheman](#)«.

Examination

Notera följande angående tid för rättning:

- Det brukar ta upp till **3 veckor tills rättningen är klar**
- Ansvarig kan **inte rätta vissa personers uppgifter före andras**, exempelvis de som har bråttom på grund av CSN

- När man fått betyg på kursen så dröjer det **ytterligare tid** tills allt är infört i Daisy och Ladok

Var god **e-posta inte ansvarig om detta** för att försöka påskynda processen, det är väldigt många studenter på samtliga kurser.

Man får välja tid för tenta och deadline oberoende av om man går på SU eller KTH och så vidare. Typen nedan är enbart en rekommendation.

Deadlines för uppgifterna

Det finns flera deadlines för inlämning av uppgifterna och man väljer själv en som passar. Man redovisar samtliga uppgifter direkt eftersom det av praktiska skäl tyvärr inte går att delrätta och ge återkoppling efter hand som deluppgifterna blir klara. De som går en + kurs kan dock lämna in ordinarie uppgifter och + uppgifter separat vid olika deadlines. Varje deadline är normalt daterad till dagen innan tentamen.

Tid	Rum	Typ
2005-05-03 kl 24	Forum, Kista	Deadline 1
2005-05-17 kl 24	Forum, Kista	Deadline IP4+
2005-08-12 kl 24	Forum, Kista	Deadline 2/Sommartentamen

Observera:

- Redovisade uppgifter **rättas endast i samband med dessa deadlines** och **uppskov ges aldrig**
- Får man **rest** så **rättas den igen först vid nästa deadline**
- Betyget påverkas inte beroende på vilken deadline man väljer
- Man måste inte välja en deadline som ligger i anslutningen till en tenta
- Handledningen slutar ofta innan deadline 1 (det finns tyvärr inga resurser för en längre handledningsperiod)

Tentamen

Inga hjälpmedel är tillåtna på tentamen. Man bör ha gjort klart de obligatoriska uppgifterna för 5 poängskursen (exklusive gesällprovet) innan tentamen eftersom tentamen kommer att fråga om dessa (detta är dock inget strikt krav för att få gå upp på tentamen). Vissa frågor kan vara mer generella och vissa kan vara mer tekniska och fråga om hur viss kodning (med avseende på uppgifterna) görs.

Tid	Rum	Typ
2005-05-04 kl 16-20	Forum, Kista	Tentamen
2005-08-13 kl 10-14	Forum, Kista	Omtentamen/Sommartentamen

Observera:

- Man måste anmäla sig i Daisy **senast en vecka innan tentamen går**, se »[DSV: Att tentera vid DSV](#)«
- Man måste inte välja en tenta som ligger i anslutningen till en deadline
- Sal meddelas i Daisy eller på telefonsvarare 08 674 70 04 senast en dag innan tentamen, se även »[DSV: Scheman](#)«

Var god **e-posta inte ansvarige** (han har inga befogenheter här) **om detta**, utan kontakta istället »[DSV: Studentexpeditionen](#)«.

Kursmaterial

På denna sida finns information om arbetssätt och vilka mjukvaror man behöver. Här finns även OH-bilder och tidigare tentor.

Arbetssätt och mjukvaror

På denna kurs kan man arbeta på följande sätt:

- På DSV
- På distans online kopplad via telnet eller SSH till sitt DSV/Unix-konto:
 - [Terminalprogram](#)
- På distans offline med det egna systemet konfigurerat för allt som behövs:
 - [Editor](#)
 - [Java](#)
 - »[Java Micro Edition Wireless Toolkit](#)«
 - [Filöverföringsprogram](#)

Det senare är det klart roligaste och mest lärorika sättet eftersom man då måste installera allt som behövs på den egna datorn. Notera att ingen handledning ges på detta eftersom det finns alltför många olika varianter av konfigurationer.

OH-bilder

OH-bilderna finns enbart här:

- [00_intro.pdf](#)
- [01_content.pdf](#)
- [02_xml.pdf](#)

Tidigare tentor

Tidigare tentor finns enbart här:

- [Tidigare tentor](#)

Notera att de är baserade på de uppgifter som var obligatoriska vid tidpunkten för tentan och som möjligtvis inte är det nu.

Terminalprogram

På denna sida beskrivs hur man använder terminalprogram.

Terminalserver

Om man behöver logga in och köra kommandon på en avlägsen dator så använder man ett terminalprogram/klient. På DSV används följande dator som terminalserver:

- `triton.dsv.su.se`

Denna dator kör Linux och har både telnet- och SSH-server igång.

Om man inte släpps in kan det bero på att DSV's brandvägg inte släpper in just det IP-nummer som man använder. Då kan man kontakta [Christer Jansson](#) som kan registrera IP-nummret i brandväggen så att man kommer förbi den.

Telnet

I Win32 och Linux så kan man köra ett Terminal-program direkt i ett kommando-fönster:

- o telnet triton.dsv.su.se

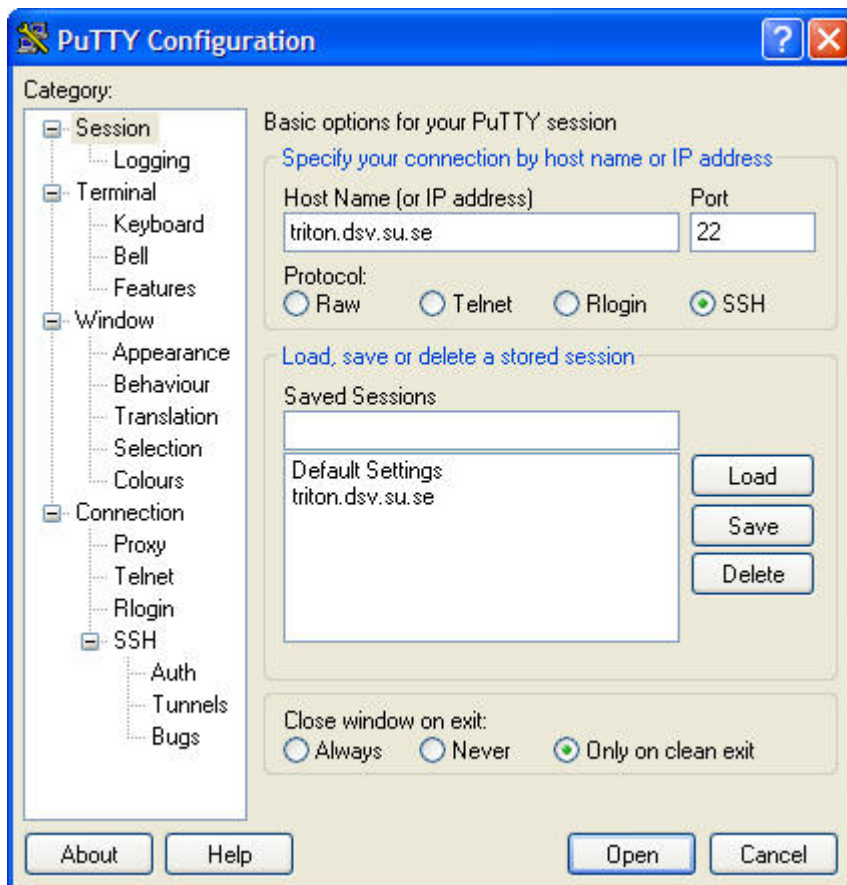
Man anger sedan användarnamn och lösenord.

SSH

När man använder telnet så sänds all information okrypterad över Internet och det kan vara farligt. Vill man undvika det kan man istället använda SSH och följande program rekommenderas:

- o Win32: »[Putty](#)«
- o Linux: ssh

Så här ser det ut när man startar Putty:



Editor

På denna sida beskrivs det verktyg man använder för att skapa kod med.

Intro

Man kan använda allt från en enkel editor till en avancerad grafisk utvecklingsmiljö. Valet är fritt! En avancerad grafisk utvecklingsmiljö kan vara något svårare att komma igång med men är

överlägsen när man väl lärt sig den. En grafisk utvecklingsmiljö kan generera vissa delar av koden i ett program automatiskt, exempelvis den kod som tar han om det grafiska användargränssnittet, vilket snabbar upp konstruktionsarbetet betydligt. En stor nackdel med detta är att koden tenderar att bli omfattande och främmande för en nybörjare och det blir mycket svårt att anknyta sin egen kod till den kod som verktyget har skapat. Det bästa är att börja med en enkel editor och sedan gå över till en grafisk utvecklingsmiljö.

Observera att koden ska vara läsbar direkt i en vanlig browser för enklare rättning. Man bör därför inte använda en editor som använder kontrolltecken och tabbar för att indentera koden.

Enkla

- Notepad (Win 32)
- »[Pico](#)« (Unix)
- »[Emacs](#)« (Win32 och Unix)

Använder man Notepad så måste man spara filerna genom att skiva citationstecken runt filnamnet "x.java" när man namnger och sparar filen, annars sparas filen som x.java.txt.

Använder man emacs för Java bör man inte använda JDE-tillägget eftersom det lätt blir fel med classpath med mera. Kompilera, kör, sätt classpath med mera i ett kommandofönster istället:

- Skriv koden i en editor och spara filen (här x.java)
- Starta ett kommandofönster och gå till katalogen där koden ligger sparad
- Kompilera med `javac x.java`
- Kör med `java x`

Tips Win32:

- Har man Windows95/98/ME så bör man börja med att skriva doskey i kommandofönstret så att man kan använda "uppåtpil" och "nedåtpil" istället för att skriva om allt man skrivit tidigare
- Man kan lägga en genväg till DOS-promten i varje katalog och konfigurera (höger-klicka och välj Egenskaper) så att den startar i samma katalog:
 - Windows 98: Välj Program/Arbetsmapp och skriv en .
 - Windows 2000/XP: Välj Shortcut/Start in och gör detta text-fält tomt
- När det är långa krångliga kommandon så kan man i DOS använda BAT-filer genom att skriva kommandot i en fil med namnet run.bat så kan man sedan köra kommandot med run

Vill man strukturera upp sin XHTML-kod så rekommenderas:

- »[W3C: Tidy](#)«
- »[SourceForge: Tidy](#)«
- »[Charlie's Tidy Add-Ons](#)«

Mer komplexa

- »[SciTe](#)«, se även »[SciTe Bundled](#)«-versionen
- »[JEdit](#)« (för Java-kod)
- »[HTML-kit](#)« (för XML- och XHTML-kod)

Konfiguration av SciTE:

- Ladda den fil som bestämmer SciTE's funktion: Options / Load Global Options File
- Sen kan man sätta lite olika saker, exempelvis:
 - Använd 3 mellanslag vid intendering: `indent.size=3`
 - Använd inte tabbar vid intendering: `use.tabs=0`

- Spara sen filen

Sessionshantering med SciTE:

- Om man har många filer öppna och vill fortsätta med dessa senare så sparar man sessionen med: `File / Save Session ...`
- När man vill öppna alla filer igen sen så går man till: `File / Load Session ...`

Kompletta utvecklingsmiljöer

- »[Eclipse](#)« (för Java-kod)

Filöverföringsprogram

På denna sida beskrivs hur man använder filöverföringsprogram.

Intro

För att kunna sända filer mellan olika datorer använder man filöverföringsprogram och vanliga standarder för det är FTP, SFTP och FTPS.

Ett vanligt fall är när man har HTML-filer som man vill sända till DSV för att synliggöra dem på Internet. Man använder då sitt DSV/Unix/Linux-konto och lägger allt under `www`-katalogen. När filerna är överförda måste man även sätta rättigheterna på filerna och katalogerna:

- Kataloger: `chmod 755`
- De flesta filtyperna: `chmod 744 *`
- CGI-filer: `chmod 711 *.cgi`

Notera att `www`-katalogen inte ska vara med i URL'ens sökväg när man sedan ska hämta filer i en browser:

- En fil `z.html` på DSV/Unix/Linux-kontot under katalogen `www/x/y` når man i en browser med `http://dsv.su.se/~<användarnamn>/x/y/z.html`

Man kan även nå sina filer på DSV/Win32 men det är endast DSV/Unix/Linux som har en tillförlitlig HTTP-server.

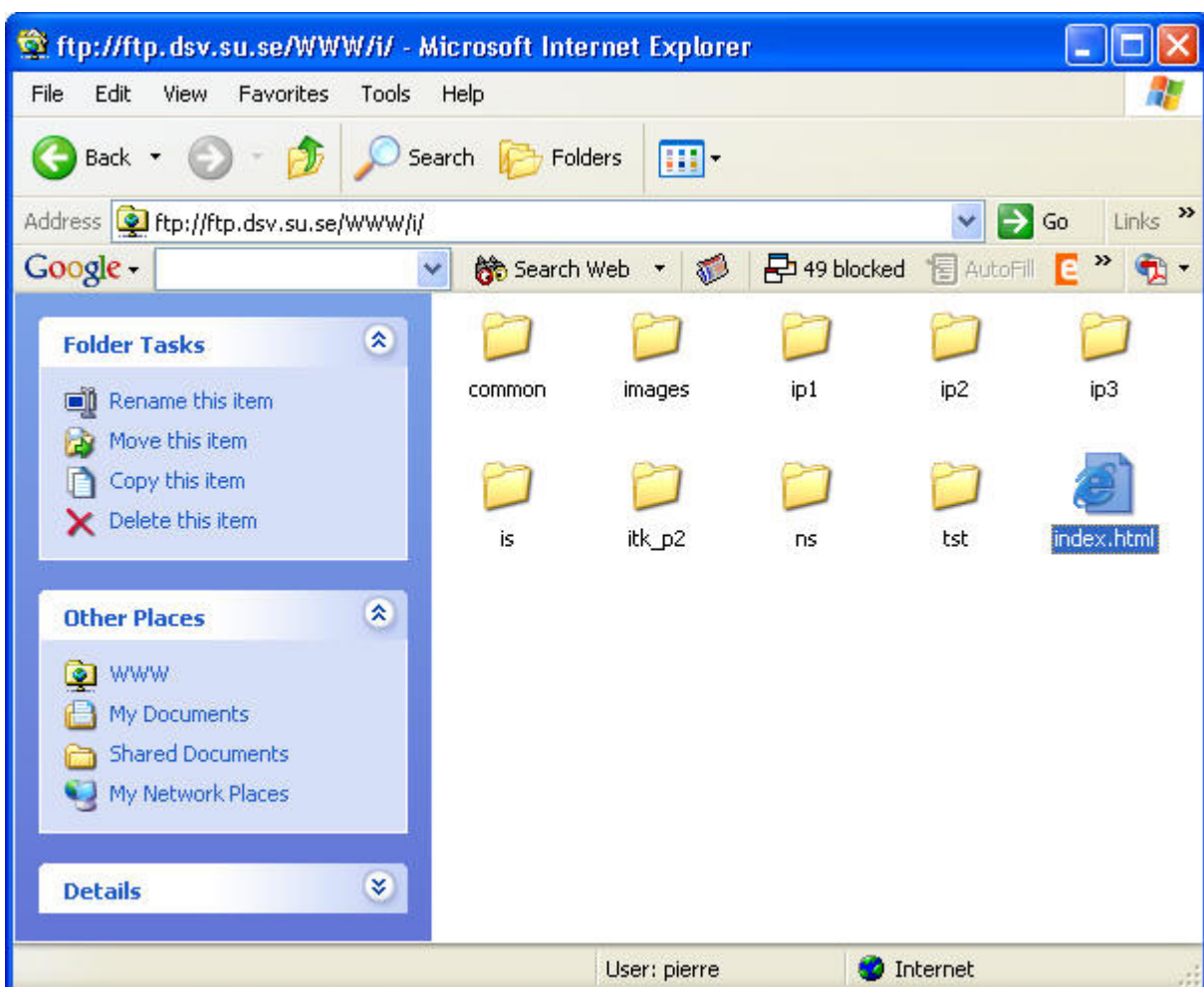
FTP

I Win32 och Linux så kan man köra en FTP-klient direkt i ett kommando-fönster:

```
Command Prompt - ftp ftp.dsv.su.se
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Pierre A. I. Wijkman\Desktop>ftp ftp.dsv.su.se
Connected to ftp.dsv.su.se.
220 mimas FTP server (Version 6.00+heimdall-0.4e+krb4-1.1) ready.
User (ftp.dsv.su.se:(none)):
```

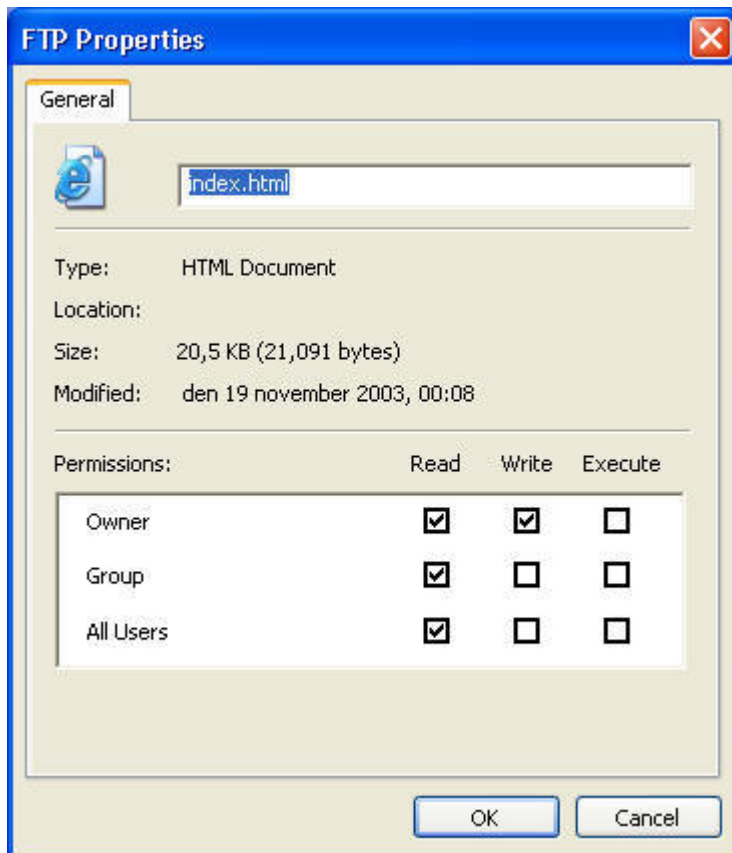
Det är dock ofta lättare att använda en GUI-baserad FTP-klient. Det finns många olika program och »Shareware.com« är ett ställe som man kan leta på. Man även använda en vanlig browser, exempelvis Internet Explorer:



Man använder sen drag-and-drop. Man kan även sätta fil- och katalogrättigheter från detta program:

- Högerklicka på fil eller katalog
- Välj Egenskaper
- Sätt rättigheter

Så här kan det se ut:



Notera dock att Internet Explorer inte stöder ASCII-mode.

Man ansluter sig till FTP-servern på följande dator:

- DSV/Unix/Linux: `ftp://ftp.dsv.su.se`

Använder man Internet Explorer så kan man skapa en genväg enligt:

- `ftp://<användarnamn><lösenord>@ftp.dsv.su.se`

Notera att man bör sända:

- Rena text-filer (som CGI-program med mera) i ASCII-mode
- Binära filer (som bilder, kompillerade program med mera) i BINARY-mode

Använder man en brandvägg så bör man se till att initieringen av dataöverföringskanalen sker på klient-sidan och det gör man genom att använda `passive mode` som kan ställas in i de flesta FTP-klienter.

SFTP och FTPS

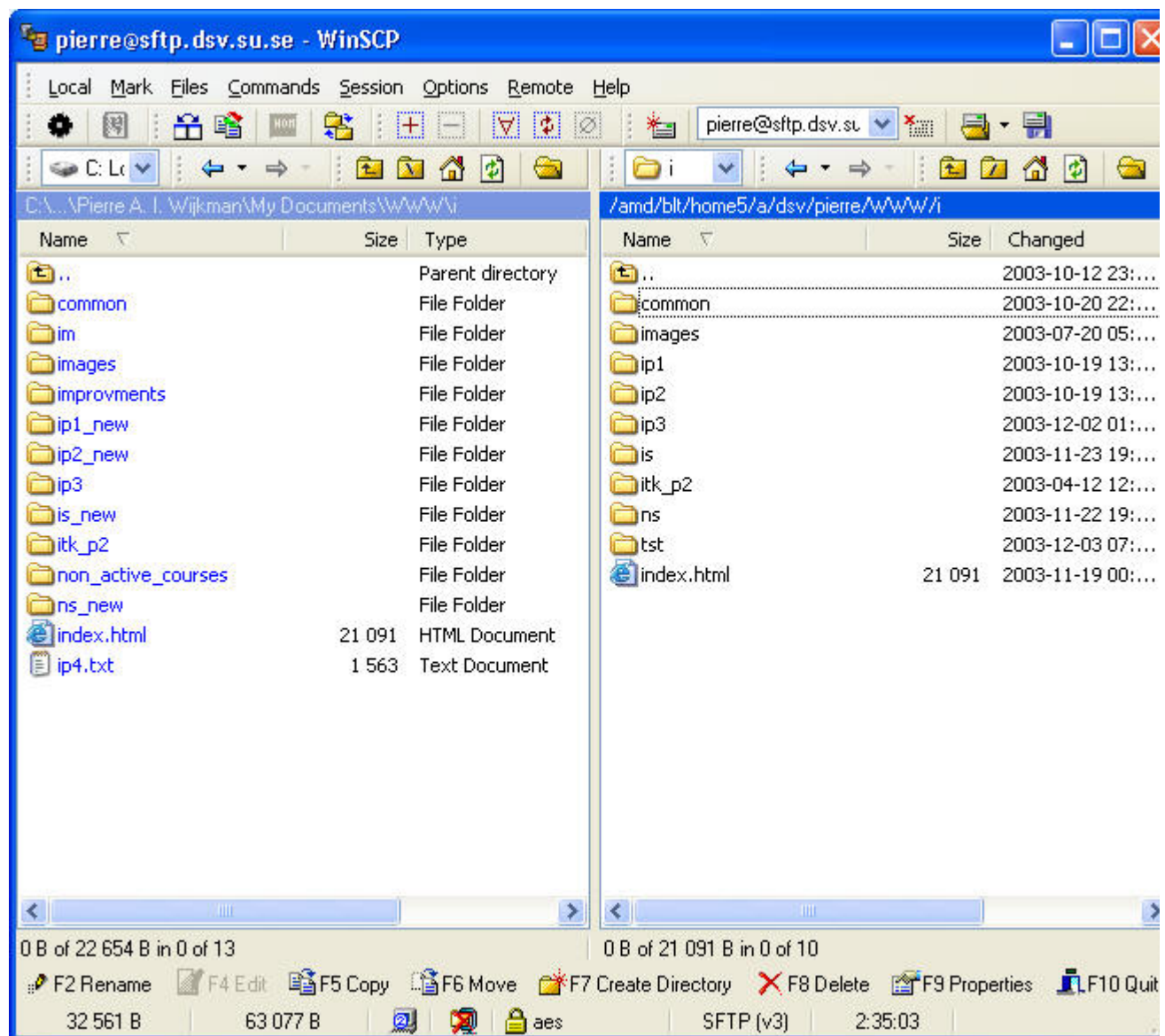
När man använder FTP så sänds all information, även lösenordet, okrypterat över Internet och det kan vara farligt. Man bör istället använda någon av följande tekniker:

- SFTP som använder sig av tunnling via SSH och först upprättar en SSH-anslutning som sedan används för att ansluta till en server via FTP
- FTPS som ansluter som vanlig FTP enligt RFC959 för att sedan begära en säker anslutning via SSL/TLS enligt RFC2228

DSV stöder enbart SFTP så man måste använda ett sådant program och bra program för det är:

- Unix/Linux: sftp som ingår i OpenSSH
- Win32: »[WinSCP](#)«

Exempelvis så ser WinSCP ut så här:



Man ansluter till SFTP-servern på följande datorer:

- DSV/Unix/Linux: sftp.dsv.su.se
- DSV/Win32: wftp.dsv.su.se

Detta säkrare sätt rekommenderas starkt.

Tidigare tentor

Tentamen

Internetprogrammering IV

Skriv tydligt. Kod ska skrivas på ett mycket välstrukturerat sätt. Använd gärna figurer för att förtydliga. Totalt bör hela svaret omfatta minst 2 sidor.

Beskriv RMI generellt och visa med Java-kod hur en minimal RMI-klient kan se ut.

Inga hjälpmedel. Pierre A. I. Wijkman ▫ 2005-05-04

Externa källor

På denna sida finns länkar till externa källor. Tipsa gärna om fler bra länkar.

Java

- » [SUN: Java](#)«
- » [SUN: Core Java J2SE 5.0](#)«
- » [SUN: JDK 5.0 Documentation](#)«
- » [SUN: Technologies Home](#)«
- » [SUN: Java Community Process Program](#)«
- » [SUN: SunSource.Net](#)«
- » [JavaWorld](#)«
- » [JavaPro](#)«
- » [Java Almanac: The Java Developers Almanac 1.4](#)«
- » [O'Reilly: ONJava.com](#)«
- » [Jakarta: The Apache Jakarta Project](#)«
- » [IBM: Java Technology](#)«
- » [Java & Internet Glossary](#)«
- » [Java Programming Resources](#)«
- » [JALOPY: Java Source Code Formatter Beautifier Pretty Printer](#)«

Plug-in och Web Start

- » [SUN: Java Plugin Guide](#)«
- » [SUN: Java Web Start Technology](#)«
- » [SUN: Deploying Software with JNLP and Java Web Start](#)«
- » [SUN: Java Web Start Developer's Guide](#)«

API Specifikationer

- » [SUN: API Specifications](#)«
- » [SUN: JavaTM 2 Platform, Standard Edition 5.0 API Specification](#)«

Lektioner

- » [SUN: Tutorials](#)«
- » [SUN: The Java Tutorial](#)«
 - » [SUN: Writing Applets](#)«
 - » [SUN: Threads: Doing Two or More Tasks At Once](#)«
 - » [SUN: JAR Files](#)«
 - » [SUN: Regular Expressions](#)«
- » [SUN: Regular Expressions and the Java Programming Language](#)«

- » [AppletWeb](#)«

XML

- » [W3C: Document Object Model](#)«
- » [Apache: XML](#)«
- » [SUN: XML](#)«
- » [SUN: Java Technology and XML-Part One](#)«
- » [JDOM.org: JDOM](#)«
- » [JDOM.org: JDOM v1.0 API Specification](#)«
- » [JavaWorld: Easy Java/XML integration with JDOM: Part 1](#)«
- » [JavaWorld: Easy Java/XML integration with JDOM: Part 12](#)«
- » [O'Reilly: XML.com](#)«
- » [O'Reilly: XML.com: The XMLPULL API](#)«
- » [PerfectXML](#)«
- » [Startkabel: XML](#)«
- » [Processing XML with Java](#)«

Distribuerade processer

RMI

- » [SUN: Java Remote Method Invocation](#)«
- » [SUN: RMI](#)«
- » [SUN: Dynamic code downloading using RMI](#)«
- » [The Coffee Break: Introduction to Java RMI](#)«
- » [JavaWorld: Revolutionary RMI: Dynamic class loading and behavior objects](#)«
- » [Ryerson University: RMI Callbacks](#)«

XML-RPC

- » [Apache: XML-RPC](#)«
- » [Apache: xmlrpc 2.0-a1-dev API](#)«
- » [XML-RPC.com](#)«
- » [SUN: Java API for XML-Based RPC](#)«

SOAP/Web Services

- » [W3C: XML Protocol Working Group](#)«
- » [W3C: Web Services Activity](#)«
- » [OASIS: Uddi](#)«
- » [SUN: Java Technology and Web Services](#)«
- » [SUN: The Java Web Services Tutorial 1.0](#)«
- » [SUN: Working with XML: The Java/XML Tutorial](#)«
- » [Apache: WebServices - SOAP](#)«
- » [Apache: WebServices - Axis](#)«
- » [O'Reilly: WebServices.xml.com](#)«
- » [O'Reilly: Web Services](#)«
- » [O'Reilly: Top Ten FAQs for Web Services](#)«
- » [WebServices.Org](#)«
- » [IBM: SOA and Web services](#)«
- » [Google: Google Web APIs](#)«

JXTA

- » [JXTA: XML-RPC](#)«

Övrigt

- » [Berkeley: Berkeley Open Infrastructure for Network Computing](#)«

JME

- » [SUN: Java 2 Platform, Micro Edition](#)«
- » [SUN: The J2ME Wireless Toolkit 2.1](#)«
- » [SUN: Programming Wireless Devices Samples](#)«
- » [kXML Project](#)«
- » [kXML-RPC Project](#)«
- » [kSOAP Project](#)«
- » [J2ME Polish](#)«
- » [Symbian](#)«

Övrigt

- » [The Internet Engineering Task Force](#)«
- » [W3C](#)«
- » [DSV: Datormiljö](#)«
- » [KTHB Forumbiblioteket](#)«
- » [WebReference.com](#)«
- » [Webmonkey](#)«
- » [W3Schools](#)«
- » [Dev Shed Open Source Web Development](#)«
- » [CMD.exe](#)«

Uppgifter

På denna sida finns uppgifterna till kursen. Läs först noga alla [krav](#) och sedan alla [tips](#). Siffror inom () är kapitelhänvisningar.

XML

Intro

- 1.1.1 [Kärnkoncept](#) obligatorisk (XML: 1-2)
- 1.1.2 [Namnrymder](#) obligatorisk (XML: 1-2)
- 1.1.3 [Entiteter](#) obligatorisk

Dokumentmodeller

- 1.2.1 [DTD](#) obligatorisk (XML: 1-2)
- 1.2.2 [Schema](#) obligatorisk (XML: 1-2)

Processande

- 1.3.1 [Händelse-baserat med SAX](#) obligatorisk (XML: 3-4)
- 1.3.2 [Träd-baserat med DOM](#) obligatorisk (XML: 5-6)
- 1.3.3 [Högnivå-baserat med JAXP och JDOM](#) obligatorisk (XML: 7-9)

Distribuerade processer

RMI

- 2.1.1 [Klientsidan](#) obligatorisk
- 2.1.2 [Serversidan](#) obligatorisk
- 2.1.3 [Dynamisk laddning av klasser](#) frivillig
- 2.1.4 [Callbacks](#) frivillig
- 2.1.5 [Säker kommunikation](#) frivillig

XML-RPC och JAX-RPC

- 2.2.1 [Klientsidan](#) obligatorisk (XML: 11)
- 2.2.2 [Serversidan](#) obligatorisk (XML: 11)
- 2.2.3 [Säker kommunikation](#) frivillig

Web Services med SOAP och WSDL

- 2.3.1 [Klientsidan](#) frivillig (XML: 12-14)
- 2.3.2 [Serversidan](#) frivillig (XML: 12-14)
- 2.3.3 [Säker kommunikation](#) frivillig

Mobila enheter med J2ME

Intro

- 3.1.1 [Grafiska gränssnitt och händelsehantering](#) frivillig IP4, obligatorisk IP4+ (J2ME: 1-6)
- 3.1.2 [Lagring med Record Management System](#) frivillig IP4, obligatorisk IP4+ (J2ME: 8)

Nätverkskopplingar

- 3.2.1.1 [Stream sockets på klientsidan](#) frivillig IP4, obligatorisk IP4+ (J2ME: 7)
- 3.2.1.2 [Stream sockets på serversidan](#) frivillig
- 3.2.2 [Datagram sockets](#) frivillig

XML-baserade distribuerade processer

- 3.3.1 [XML](#) frivillig
- 3.3.2 [XML-RPC](#) frivillig
- 3.3.3 [Web Services med SOAP och WSDL](#) frivillig

Extensioner

- 3.4 [Extensioner](#) frivillig

Gesällprov

- 4 [Gesällprov](#) obligatorisk

Krav

På denna sida finns de krav som gäller för uppgifterna.

Samarbete och betyg

- Alla uppgifter utom gesällprovet ska göras i grupp om 1-2 personer, gesällprovet ska göras ensam
- Vissa av uppgifterna är frivilliga men desto fler av dessa som man gör desto större blir sannolikheten för att få betyget VG/5
- Man kan givetvis ändå få betyget VG/5 om man gör ett väldigt bra gesällprov
- Bästa gesällprov till deadline 1 vinner ett pris

Kodens utseende

En vanlig orsak till betyget R är att nedanstående krav inte är uppfyllda.

All kod

- All kod ska vara välstrukturerad
- Man får själv bestämma stil på koden men man måste vara konsekvent
- Använd inte en editor som använder kontrolltecken och tabbar för indentering, koden ska kunna läsas och se bra ut i en vanlig enkel editor som exempelvis Notepad och därmed även i en browser
- Designen på det (eventuella) grafiska användargränssnittet är fri men försök göra det så snyggt som möjligt (man bör alltså inte göra programmen identiska med de ofta lite tråkiga exempelprogram som ges för varje uppgift)

Java-kod

- All Java-kod ska vara välkommenterad med en kommentar per klass och metod, se vidare »[SUN: Code Conventions for the Java Programming Language](#)«
- Eventuella äpplen som ska redovisas kan vara körbara antingen via appletviewer eller via en plug-in i en vanlig browser
- Använd Swing-GUI-komponenter, inte de nu omoderna AWT-GUI-komponenterna (givetvis är det ok att använda layoutmanagers, händelseklasser med mera från AWT)

Tips

På denna sida finns tips för uppgifterna.

Ordning och organisation

- Använd en katalog för varje huvuduppgift
- Under hela konstruktionsprocessen:
 - Välstrukturerad kod
 - Bra namn på klasser, variabler, konstanter

Evolutionär utveckling

Många moderna programmeringsspråk består av:

- En liten språkkärna
- Ett mycket stort bibliotek av färdigkonstruerade klasser

Den självklara fördelen är att man slipper konstruera en stor mängd saker som annars skulle ta mycket lång tid att göra men nackdelen är att det kan vara svårt att veta och hitta vad som finns

samt att man inte får samma grundläggande förståelse.

Matematik är ingenting man förstår, matematik är något som man vänjer sig vid

von Neumann

Till att börja med bör man skaffa sig en övergripande förståelse för det område man ska ge sig in på. Detta görs bäst med hjälp av böcker, tidningar och/eller de tutorials och artiklar som finns på Internet. När man har skaffat sig en denna övergripande förståelse så bör man så fort som möjligt testa några av de exempel man stött på. Det är ofta först när man praktiskt tillämpar ett område som man lär sig, eller vänjer sig vid, området och därmed får en förankring eller förståelse av beteendet hos objekten man laborerar med.

Man bör inte skriva av koden "rakt av" utan man bör istället:

- Konstruera en minimal körbar kärna (med sin egen stil på koden)

Dett gör man för varje nytt område man ger sig in på. Man kan sedan:

- Förändra denna minimala körbara kärna mot ett önskbart mål med **små förändringar** av koden

Detta är mycket viktigt! Man går mot ett önskat mål med:

- Små förändringar av koden
- Kompilering
- Testkörning

Man släpper aldrig kontakten med det fungerande programmet.

We are like sailors who on the open sea must reconstruct their ship but are never able to start afresh from the bottom. Where a beam is taken away a new one must at once be put there, and for this the rest of the ship is used as support. In this way, by using the old beams and driftwood the ship can be shaped entirely anew, but only by gradual reconstruction.

Otto Neurath

Detta är en mycket viktig tumregel. Försöker man förändra koden i stora steg så blir det fel och dessa fel är ofta svåra att lokalisera och rätta till. Ofta samverkar flera fel och detta gör det givetvis ännu svårare.

Ofta är stora program konstuerade av många lager/nivåer där varje nivå består av program konstruerade enligt ovanstående enkla principer. Man använder sedan program på en nivå som enheter för konstruktion av program på nästa nivå, och även denna process följer de ovan anvisade principerna.

Övrigt

- Detta är inte övningsuppgifter och man måste läsa kursboken och viss information som finns på Internet för att klara uppgifterna
- Alla program är testade i DSVs miljö för både Win32 och (det överlägsna :-) Unix/Linux
- Alla program går att utveckla lokalt ej uppkopplad (på Win32 och Unix/Linux) men man måste ha TCP/IP installerat på sin dator (vilket man har om man exempelvis surfar på Internet)
- Man kan använda IP-adressen 127.0.0.1 eller localhost som motsvarar den egna maskinen

XML: Intro: Kärnkoncept

Allmänt

Uppgiften introducerar till och går igenom det mesta inom XML.

Uppgift

Gör en XML-fil som:

- Beskriver ditt CV
- Är relativt komplicerad med:
 - Element som innehåller element (minst 2 nivåer)
 - Attribut

Kontrollera att XML-filen är välutformad.

Exempel

Inget exempel ges.

Tips

Man kan kontrollera att en XML-fil är välutformad genom att visa den med exempelvis Firefox. Är det fel i XML-filen meddelar Firefox detta.

XML: Intro: Namnrymder

Allmänt

Uppgiften introducerar till namnrymder.

Uppgift

Bygg ut uppgiften [XML: Intro: Kärnkoncept](#) så att den använder minst två namnrymder.

Kontrollera att XML-filen är välutformad.

Exempel

Inget exempel ges.

Tips

Inga tips ges.

XML: Intro: Entiteter

Allmänt

Uppgiften introducerar till entiteter.

Uppgift

Bygg ut uppgiften [XML: Intro: Kärnkoncept](#) så att den använder entiteter:

- Interna entiteter
- Externa entiteter (genom att först dela upp XML-dokumentet i delar och sedan bygga ihop dessa delar från ett annat XML-dokument)

Kontrollera att XML-filen är välutformad. Frivillig överkurs är att även titta på XInclude.

Exempel

Inget exempel ges.

Tips

Firefox har en bugg för externa entiteter »[Bugzilla Bug 69799: External entities are not included in XML document](#)« och man bör här använda IE eller någon annan browser istället.

XML: Dokumentmodeller: DTD

Allmänt

Uppgiften introducerar till dokumentmodeller med DTD.

Uppgift

Skapa en dokumentmodell med en DTD som definerar XML-dokumentet i uppgiften [XML: Intro: Kärnkoncept](#) så strikt det går. Kopiera sedan XML-dokumentet och lägg till en pekare som visar att det tillhör denna DTD.

Kontrollera att XML-filen är valid.

Exempel

Inget exempel ges.

Tips

Inga tips ges.

XML: Dokumentmodeller: Schema

Allmänt

Uppgiften introducerar till dokumentmodeller med schema.

Uppgift

Skapa en dokumentmodell med ett schema som definerar XML-dokumentet i uppgiften [XML: Intro: Kärnkoncept](#) så strikt det går. Kopiera sedan XML-dokumentet och lägg till en pekare som visar att det tillhör detta schema.

Kontrollera att XML-filen är valid.

Exempel

Inget exempel ges.

Tips

Inga tips ges.

XML: Processande: Händelse-baserat med SAX

Allmänt

Uppgiften introducerar till hur Java kan accessa XML-dokument händelse-baserat med SAX.

Uppgift

Skapa ett fristående program som vid kommando-prompten startas med:

```
java SAXChecker <xml-document> <validate>
```

som kan avgöra om filen <xml-document> är välutformad och/eller valid (beroende på om <validate> är angiven som true eller false). Använd en SAX-parser och XML-dokumenterna från uppgifterna [XML: Intro: Kärnkoncept](#) och [XML: Dokumentmodeller: DTD](#).

Exempel

Hämta filen:

- [SAXChecker.class](#)

Kör programmet med:

○ `java SAXChecker <xml-document> <validate>`

Tips

För att få en parser att validera så kan man exempelvis göra:

```
XMLReader parser = XMLReaderFactory.createXMLReader();  
parser.setFeature("http://xml.org/sax/features/validation", true);
```

XML: Processande: Träd-baserat med DOM

Allmänt

Uppgiften introducerar till hur Java kan accessa XML-dokument träd-baserat med DOM.

Uppgift

Skapa ett fristående program som vid kommando-prompten startas med:

```
java DOMEcho <xml-document>
```

som listar innehållet i filen <xml-document>. Använd en DOM-parser och XML-dokumentet från uppgiften [XML: Intro: Kärnkoncept](#).

Exempel

Ett exempel kommer troligtvis senare.

Tips

Inga tips ges.

XML: Processande: Högnivå-baserat med JAXP och JDOM

Allmänt

Uppgiften introducerar till hur Java kan accessa XML-dokument högnivå-baserat med JAXP och JDOM.

Uppgift

Skapa ett fristående program som implementerar en chat-klient som använder ett XML-baserat protokoll enligt DTD-filen [message.dtd](#) för meddelandena. Programmet ska göra följande:

- Vid sändning av meddelanden:
 - Använd klassen Document för att skapa ett XML-dokument som pekar på den publika DTD-filen http://atlas.dsv.su.se/~pierre/i/05_ass/ip4/a_1_3_3/message.dtd för meddelandet
 - Använd klassen XMLOutputter för att få XML-dokumentet formaterat som en String utan radbrytningar för sändning till chat-servern
- Vid mottagning av meddelanden:
 - Använd klassen SAXBuilder för att kontrollera validitet och skapa ett Document-objekt
 - Använd detta Document-objekt för att accessa de specifika delarna i XML-dokumentet som man sedan ska visa användaren: <namn> (<epost>): <meddelande>

På detta sätt får vi en strikt kontroll av att de meddelanden som sänds och tas emot följer vårt protokoll och att andra program enkelt kan koppla ihop sig med detta system. Gör även:

- En snygg utskrift i kommandoprompten av det XML-dokument som sänds och tas emot
- En felkontroll så att användaren informeras om de meddelanden som inte följer protokollet med utskriften KUNDE INTE PARSA ETT MEDDELANDE: <meddelande>

Notera att beskrivningen i denna uppgift är baserad på extensionen JDOM men att man istället kan använda det i Java inbyggda JAXP om man vill.

Exempel

Ett enkelt exempel som använder extensionen JDOM kan köras enligt följande:

Hämta filerna:

- [jdom.jar](#)
- [ChatClient.class](#)
- [Chatter.class](#)
- [XMLChatClient.class](#)
- [XMLChatClient\\$L1.class](#)
- [XMLChatClient\\$L2.class](#)

Kör programmet med:

- Unix/Linux: `java -classpath jdom.jar:. XMLChatClient`
- Win32: `java -classpath jdom.jar;. XMLChatClient`

Programmet startar default mot chat-servern på atlas.dsv.su.se port 9494 och om man vill använda en egen chat-server lokalt så kan man starta programmet enligt uppgiften [Lågnivåkopplingar: Stream sockets: Klient-sidan](#) från kursen Internetprogrammering I.

Tips

Om man har gjort chat-klienten i uppgiften [Lågnivåkopplingar: Stream sockets: Klient-sidan](#) från kursen Internetprogrammering I så kan man utgå från den lösningen men för de som inte har gjort denna uppgift så finns klassen [ChatClient.class](#) som är en hjälpklass som enbart hjälper till med:

- Anknytning med metoden `public void connect(String host, int port)`
- Sändning med metoden `public void send(String message)`
- Mottagning genom att den egna klassen implementerar interfacet [Chatter.java](#)

När man skapar en instans av denna klass så måste man sända med en referens till den egna klassen:

```
ChatClient chatClient = new ChatClient(this);
```

Man måste vara duktig på paketet `java.io` för att lösa denna uppgift och man kan exempelvis använda följande konstruktion för att bygga ett XML-dokument direkt från ett `String`-objekt:

```
Document doc = builder.build(new StringReader(in.readLine()));
```

Eftersom man måste sända hela XML-dokumentet på en rad så behöver man konvertera ett `Document`-objekt till ren text utan radbrytningar och då är metoden `setLineSeparator` bra:

```
Format format = Format.getCompactFormat();  
format.setLineSeparator("");  
XMLOutputter xmlOutputter = new XMLOutputter(format);
```

För att visa saker snyggt i kommandofönstret behöver man dock använda ett annat format.

Distribuerade processer: RMI: Klientsidan

Allmänt

Uppgiften belyser distribuerade processer med RMI där man delar upp ett program i flera delar där delarna körs på olika maskiner. Med RMI kan man i sina program använda objekt som körs på andra maskiner på ett mycket naturligt sätt, trots att objekten egentligen "lever" på olika maskiner.

Uppgift

På servern `atlas.dsv.su.se` körs ett RMI-serversidesprogram (med tillhörande stubbe [Server_Stub.class](#)) som man kan nå via RMI-registret på defaultporten 1099 anknäring server och som kan anropas via de metoder som programmet implementerar i interfacet [RemoteServer.java](#). Detta program simulerar ett antal bollar som studsar runt inom ett koordinatsystem som har:

- Origo längst upp till vänster
- Positiva x-värden åt höger
- Positiva y-värden neråt

Varje boll beskrivs med:

- Position x som ligger inom intervallet 0-500
- Position y som ligger inom intervallet 30-500
- Radie r som ligger inom intervallet 0-70, där bollar med radie 0 tas bort

Metoderna som kan anropas har följande funktion:

- `addBall` lägger till en ny boll
- `pauseBalls` pausar eller startar bollarna
- `getBalls` returnerar alla bollar i en vektor där:
 - Element 1 innehåller första bollens x -värde
 - Element 2 innehåller första bollens y -värde
 - Element 3 innehåller första bollens radie
 - Element 4 innehåller andra bollens x -värde, och så vidare

Gör ett RMI-klientsidesprogram som använder dessa metoder för att grafiskt åskådliggöra de processer som kan sättas igång och som körs på RMI-serversidesprogrammet.

Frivillig överkurs är att istället implementera Mandelbrots mängd på liknande sätt fast då även göra serversidesprogrammet samtidigt.

Exempel

Hämta filerna:

- [Client.class](#)
- [Client\\$L.class](#)
- [RemoteServer.class](#)
- [Server_Stub.class](#)

Kör igång RMI-klientprogrammet med:

- `java Client atlas.dsv.su.se`

Tryck på +-knappen för att skapa nya bollar och på p-knappen för att pausa och sätta igång bollarna. Testa att starta flera program för att se att bollarna är synkroniserade på de olika programmen, gärna på flera olika maskiner. Man kan även testa att köra RMI-klientprogrammet som ett äpple:

- `appletviewer http://atlas.dsv.su.se/~pierre/i/05_ass/ip4/a_2_1/space/client/client.html`

Ett annat exempel är ett RMI-klientprogram för Mandelbrots mängd. Hämta filerna:

- [Client.class](#)
- [Client\\$ThePanel.class](#)
- [Client\\$MouseListener1.class](#)
- [Client\\$MouseMotionListener1.class](#)
- [RemoteServer.class](#)
- [Server_Stub.class](#)

Kör igång RMI-klientprogrammet med:

- `java Client atlas.dsv.su.se`

Zooma i bilden genom att trycka ner vänster musknapp och dra ut en fyrkant. Man kan även testa att köra RMI-klientprogrammet som ett äpple:

- `appletviewer http://atlas.dsv.su.se/~pierre/i/05_ass/ip4/a_2_1/mandelbrot/client/client.html`

Tips

Var väldigt noga och tänk minimalistiskt, det blir lätt fel annars.

Man kan givetvis vända på det här med att köra krävande processer på serversidan. Man kan exempelvis göra en distribuerad beräkning genom att konstruera ett äpple som beräknar en del av en krävande process och där serversidan enbart organiserar de olika processerna. Har man många besökare på sin webbplats så kan man på så sätt få tillgång till enormt mycket datakraft, men man bör givetvis informera besökarna om att man använder deras datorer (speciellt om man använder dolda äpplen) och även göra äpplet så att det inte förbrukar alla resurser på besökarnas datorer.

Slutligen några tips om [Mandelbrots mängd](#) för de som vill implementera uppgiften på detta sätt.

Distribuerade processer: RMI:

Serversidan

Allmänt

Uppgiften belyser distribuerade processer med RMI på serversidan.

Uppgift

Gör RMI-serversidesprogrammet som passar till uppgiften [Distribuerade processer: RMI: Klient sidan](#).

Man behöver bara testa sitt RMI-klientprogram och RMI-serverprogram lokalt på den egna maskinen men det är extra kul om man kör på flera olika maskiner.

De som valt att implementera Mandelbrots mängd har redan gjort denna uppgift och kan gå vidare till nästa uppgift.

Exempel

Hämta filerna:

- [RemoteServer.class](#)
- [Server.class](#)
- [Server\\$Ball.class](#)

Kör igång RMI-serverprogrammet med:

- `java Server`

Ett annat exempel är ett RMI-serverprogram för Mandelbrots mängd. Hämta filerna:

- [RemoteServer.class](#)
- [Server.class](#)

Kör igång RMI-serverprogrammet med:

- `java Server`

Tips

Istället för att starta ett `rmiregistry` manuellt, innan man startar sitt RMI-serverprogram, så kan man starta ett `rmiregistry` direkt i Java-koden med:

```
LocateRegistry.createRegistry(1099);
```

Om man redan har gjort det i en annan process så kan man istället hitta detta `rmiregistry` med:

```
Registry registry = LocateRegistry.getRegistry(1099);
```

Man kan även ha flera `rmiregistry` igång på samma maskin om man väljer olika portnummer för dessa.

Distribuerade processer: RMI: Dynamisk laddning av klasser

Allmänt

Uppgiften belyser hur man använder dynamisk laddning av klasser med RMI.

Uppgift

Gör om RMI-klientsidesprogrammet i uppgiften [Distribuerade processer: RMI: Klientsidan](#) så att det kan hämta stubben [Server_Stub.class](#) dynamiskt vid körning.

Frivillig utökning är att bygga om båda uppgifterna [Distribuerade processer: RMI: Klientsidan](#) och [Distribuerade processer: RMI: Serversidan](#) så att de även båda använder och hämtar andra klasser dynamisk.

Exempel

Hämta filerna:

- [Client.class](#)
- [Client\\$L.class](#)
- [RemoteServer.class](#)
- [policy](#)

Kör igång RMI-klientprogrammet med:

- ```
java -Djava.security.policy=policy -
Djava.rmi.server.codebase=http://atlas.dsv.su.se/~pierre/i/05_ass/ip4/a_2_1/space/server/
Client atlas.dsv.su.se
```

## Tips

RMI-klientprogrammet måste använda en `RMISecurityManager` och startas med en codebase som pekar på det ställe där stubben [Server\\_Stub.class](#) finns.

Vill man få detta att fungera med äpplen så måste man, som tidigare, lägga äpplet på samma ställe som [Server\\_Stub.class](#) ligger på.

# Distribuerade processer: RMI: Callbacks

## Allmänt

Uppgiften belyser hur man använder callbacks med RMI.

## Uppgift

Gör om RMI-programmen i uppgifterna [Distribuerade processer: RMI: Klientsidan](#) och [Distribuerade processer: RMI: Serversidan](#) så att de använder callbacks. När en RMI-klient ansluter eller avslutar sin uppkoppling till RMI-servern så ska RMI-servern informera alla anslutna klienter om vilka klienter som är anslutna och RMI-klienten ska skriva ut denna information i kommandofönstret.

## Exempel

Hämta filerna:

- [RemoteClient.class](#)
- [Client.class](#)
- [ClientGUI.class](#)
- [ClientGUI\\$L.class](#)
- [RemoteServer.class](#)
- [Server\\_Stub.class](#)

Kör igång RMI-klientprogrammet med:

- `java Client atlas.dsv.su.se`

I detta program kan man även trycka på e-knappen för att avsluta programmet. Eftersom detta program enbart fungerar på datorer med direkt anslutning till Internet (det fungerar inte bakom routrar som ger klienterna egna interna IP-adresser) så kan man även testa det lokalt på den egna datorn om man först startar upp RMI-servern. Hämta filerna:

- [RemoteServer.class](#)
- [Server.class](#)
- [Server\\$Ball.class](#)
- [Server\\$ClientSender.class](#)
- [RemoteClient.class](#)
- [Client\\_Stub.class](#)

Kör igång RMI-serverprogrammet med:

- `java Server run_local`

Kör igång RMI-klientprogrammet med:

- `java Client localhost`

Troligtvis kommer ytterligare en version som fungerar bättre under kursen.

## Tips

Man bör börja med ett minimalt exempel innan man bygger hela denna uppgift. Exempelvis så kan man börja med att låta RMI-servern anropa en metod hos RMI-klienten med en `String` som argument som sen RMI-klienten skriver ut i kommandofönstret.

I exemplet ovan har följande två interface använts:

- [RemoteServer.java](#)
- [RemoteClient.java](#)

# Distribuerade processer: RMI:

# Säker kommunikation

## Allmänt

Uppgiften belyser distribuerade processer med RMI med säker kommunikation.

## Uppgift

Gör om uppgifterna [Distribuerade processer: RMI: Klientsidan](#) och [Distribuerade processer: RMI: Serversidan](#) så att de använder säker kommunikation via SSL.

## Exempel

Inget exempel ges.

## Tips

Inget tips ges.

# Distribuerade processer: XML-RPC och JAX-RPC: Klientsidan

## Allmänt

Vi ska nu titta på hur man hos en klient, via XML, kan anropa metoder som utförs på en annan maskin. Detta är likt RMI (som dock inte använder XML) men skillnaden är att:

- Språk:
  - RMI måste köras med Java på både klient och server
  - XML-RPC och JAX-RPC kan använda olika språk och språket hos klienten behöver inte vara samma som språket hos servern
- Möjligheter:
  - RMI är kraftfullt eftersom man kan hantera egendefinierade godtyckliga (serialiserbara) objekt och kommunikationen kan vara tvåvägs
  - XML-RPC och JAX-RPC kan enbart hantera en mindre uppsättning fördefinierade typer (som ofta räcker långt) och kommunikationen går enbart från klienten till servern (men givetvis kan klienten också ha en XML-RPC-server igång och på så sätt kan man få tvåvägskommunikation)
- Utvecklarvänlighet:
  - RMI kan vara krångligt (interface, stub, RMI-registry med mera)
  - XML-RPC och JAX-RPC är lätt

Uppgiften belyser distribuerade processer med XML-RPC och JAX-RPC på klientsidan.

## Uppgift

På servern `atlas.dsv.su.se` port 7777 körs ett XML-RPC-serversidesprogram som man kan nå via anknötning `prime_handler` och som kan anropas via metoden:

```
public String getPrime(String password)
```

Detta program försöker hela tiden hitta större och större primtal och lagrar detta primtal på en fil för att kunna fortsätta vid det vid en eventuell omstart. Metoden `getPrime` returnerar således detta senast funna primtal men returnerar samtidigt även den tid då detta hittades. För att kunna köra metoden måste lösenordet `2isAnOddPrime` sändas med som argument.

Gör ett XML-RPC-klientsidesprogram som använder denna metod för att få senast funna primtal och den tid då det hittades.

Notera att vi i denna uppgift använder Apache XML-RPC men att man istället kan använda JAX-RPC om man vill.

## Exempel

Hämta filerna:

- [xmlrpc-1.2-b1.jar](#)
- [Client.class](#)

Kör igång XML-RPC-klientprogrammet med:

- Unix/Linux: `java -classpath xmlrpc-1.2-b1.jar:. Client atlas.dsv.su.se 7777 2isAnOddPrime`
- Win32: `java -classpath xmlrpc-1.2-b1.jar;. Client atlas.dsv.su.se 7777 2isAnOddPrime`

## Tips

Inga tips ges.

# Distribuerade processer: XML-RPC och JAX-RPC: Serversidan

## Allmänt

Uppgiften belyser distribuerade processer med XML-RPC och JAX-RPC på serversidan.

## Uppgift

Gör XML-RPC-serversidesprogrammet som passar till uppgiften [Distribuerade processer: XML-RPC och JAX-RPC: Klientsidan](#).

Man behöver bara testa sitt XML-RPC-klientprogram och XML-RPC-serverprogram lokalt på den egna maskinen men det är extra kul om man kör på två olika maskiner.

## Exempel

Hämta filerna:

- [xmlrpc-1.2-b1.jar](#)
- [Server.class](#)
- [PrimeHandler.class](#)

Kör igång XML-RPC-klientprogrammet med:

- Unix/Linux: `java -classpath xmlrpc-1.2-b1.jar:. Server <port>`
- Win32: `java -classpath xmlrpc-1.2-b1.jar;. Server <port>`

## Tips

Inga tips ges.

# Distribuerade processer: XML-RPC och JAX-RPC: Säker kommunikation

## Allmänt

Uppgiften belyser distribuerade processer med XML-RPC och JAX-RPC med säker kommunikation.

## Uppgift

Gör om uppgifterna [Distribuerade processer: XML-RPC och JAX-RPC: Klientsidan](#) och [Distribuerade processer: XML-RPC och JAX-RPC: Serversidan](#) så att de använder säker kommunikation via SSL.

## Exempel

Inget exempel ges.

## Tips

Inget tips ges.

# Distribuerade processer: Web Services med SOAP och WSDL: Klientsidan

## Allmänt

Uppgiften belyser distribuerade processer med Web Services på klientsidan. Lite förenklat kan man säga att Web Services med SOAP och WSDL både har RMI's kraftfullhet och XML-RPC's enkelhet.

## Uppgift

På servern `atlas.dsv.su.se` körs en Web Service (serversidesprogram) som beskrivs i en »[WSDL-fil](#)« och nås via:

- `http://atlas.dsv.su.se:8080/axis/services/PrimeService`

Detta program har samma funktion som XML-RPC-serversidesprogrammet i uppgiften [Distribuerade processer: XML-RPC och JAX-RPC: Klientsidan](#). Gör ett Web Service-klientsidesprogram som använder denna Web Service för att få senast funna primtal och den tid då det hittades.

Notera att vi i denna uppgift använder Apache Axis men att man istället kan använda Java Web Services Developer Pack (Java WSDP) om man vill. Notera dock att Java WSDP kommer att ingå i nästa version av J2SE och att det troligtvis inte är riktigt stabilt ännu.

## Exempel

Hämta filerna:

- [axis.jar](#)
- [axis-ant.jar](#)
- [commons-discovery.jar](#)
- [commons-logging.jar](#)
- [jaxrpc.jar](#)
- [log4j-1.2.8.jar](#)
- [saaj.jar](#)
- [wsdl4j.jar](#)
- [activation.jar](#)
- [mail.jar](#)
- [Client.class](#)

Kör igång Web Service-klientprogrammet med:

- Unix/Linux: `java -classpath axis.jar:axis-ant.jar:commons-discovery.jar:commons-logging.jar:jaxrpc.jar:log4j-1.2.8.jar:saaj.jar:wsdl4j.jar:activation.jar:mail.jar:. Client http://atlas.dsv.su.se:8080/axis/services/PrimeService 2isAnOddPrime`
- Win32: `java -classpath axis.jar;axis-ant.jar;commons-discovery.jar;commons-logging.jar;jaxrpc.jar;log4j-1.2.8.jar;saaj.jar;wsdl4j.jar;activation.jar;mail.jar;. Client http://atlas.dsv.su.se:8080/axis/services/PrimeService 2isAnOddPrime`

## Tips

Man kan börja med att titta på »[Google: Google Web APIs](#)« som är mycket lättanvänt men specialdesignat för just Google. Ett enkelt exempel på detta, hämta filerna:

- [googleapi.jar](#)
- [GoogleWS.class](#)

Kör igång Web Service-klientprogrammet med:

- Unix/Linux: `java -classpath googleapi.jar:. GoogleWS <sökord>`
- Win32: `java -classpath googleapi.jar;. GoogleWS <sökord>`

Kanske kan man använda detta API senare för sitt gesällprov.

# Distribuerade processer: Web Services med SOAP och WSDL: Serversidan

## Allmänt

Uppgiften belyser distribuerade processer med Web Services på serversidan.

## Uppgift

Gör den Web Service (serversidesprogrammet) som passar till uppgiften [Distribuerade processer: Web Services med SOAP och WSDL: Klient sidan](#).

Man behöver bara testa sitt Web Service-klientprogram och Web Service-serverprogram lokalt på den egna maskinen men det är extra kul om man kör på två olika maskiner. Frivillig överkurs är att även titta på UDDI.

## Exempel

Inget exempel ges.

## Tips

Man behöver [Tomcat](#) och »[Apache: Axis](#)«.

# Distribuerade processer: Web Services med SOAP och WSDL: Säker kommunikation

## Allmänt

Uppgiften belyser distribuerade processer med Web Services med säker kommunikation.

## Uppgift

Gör om uppgifterna [Distribuerade processer: Web Services med SOAP och WSDL: Klient sidan](#) och [Distribuerade processer: Web Services med SOAP och WSDL: Serversidan](#) så att de använder säker kommunikation.

## Exempel

Inget exempel ges.

## Tips

Inget tips ges.

# Mobila enheter med J2ME: Intro:

# Grafiska gränssnitt och händelsehantering

## Allmänt

Vi ska nu titta på ett område som kommer bli mycket stort och är under snabb utveckling, programmering för mobila enheter med Java Micro Edition (J2ME). Denna version av Java används för att kunna köra Java-program på enheter med:

- Begränsat minne
- Begränsad processorkraft
- Begränsade nätverksmöjligheter

Det finns i två huvudvarianter, kallade konfigurationer:

- Connected Limited Device Configuration (CLDC 1.0 och CLDC 1.1) är till för enklare maskiner, med ungefär 512 KByte minne, som mobiltelefoner och mer begränsade PDA'er
- Connected Device Configuration (CDC) är till för något kraftfullare maskiner, med ungefär 2 MByte minne, som kraftfulla mobiltelefoner, kraftfulla PDA'er, web-telefoner och set-top-boxar

Inom varje konfiguration kan man bygga på ett antal extrapaket, kallade profiler:

- CLDC Profiler:
  - Mobile Information Device Profile (MIDP 1.0 och MIDP 2.0) lägger till nätverkshantering, GUI och lokal lagringsmöjlighet
  - PDA Profile (PDAP) är lik MIDP men har bättre GUI-hantering och kan nå egenskaper i operativsystemet bättre
- CLD Profiler:
  - Foundation Profile (FP) bygger ut CDC så att man får nästan hela J2SE 1.3
    - Personal Basis Profile (PBP) lägger till GUI-funktioner i Foundation Profile så att man kan använda maskiner med begränsade möjligheter till GUI-hantering
      - Personal Profile (PP)
    - RMI Profile lägger till (klient-sides) RMI till Foundation Profile
    - Game Profile lägger till spel-specifika funktioner till CLD

J2ME består alltså av många olika varianter för många olika typer av maskiner. Flera av dessa profiler är under utveckling men några är klara. En vanlig kombination som är klar och används i många mobiler och PDA'er är:

- Konfiguration: CLDC
- Profil: MIDP

Program utvecklade under CLDC/MIDP kallas för MIDlets. Till detta finns en mängd paket (optional packages) som man kan använda, se vidare uppgiften [Mobila enheter med J2ME: Extensioner](#).

Innan denna standardisering i konfigurationer och profiler tog fart fanns det några andra versioner av Java för små maskiner:

- JavaCard är till för smarta kort och är den minsta miljön för Java (16 KByte ROM 512 Byte RAM)
- EmbeddedJava är baserat på JDK 1.1 och är till för inbyggda system (512 KByte ROM och 512 KByte RAM) och med denna variant får man ta bort godtyckligt många funktioner som inte behövs i en viss maskin
- PersonalJava är till för kraftfullare maskiner (2 MByte ROM och 1MByte RAM) och är kärnan i J2SE 1.1.8 och finns i många PDA'er och en del avancerade mobiler (exempelvis

SonyEricsson P800 och Nokia Communicator)

Dessa har ersatts av någon kombination av konfiguration/profil-paket enligt ovan.

Vi ska i denna och två efterföljande uppgifter stegvis bygga en chat-klient för mobila enheter och börjar med att titta på hur man konstruerar ett grafiska gränssnitt i J2ME.

## Uppgift

Bygg ett grafiskt gränssnitt där användaren ska kunna:

- Ange host i ett text-fält
- Ange port i ett text-fält
- Göra kommandot connect

Kommandot connect behöver i denna uppgift enbart visa nästa grafiska gränssnitt där användaren ska kunna:

- Se den senaste konversationen
- Ange meddelandet som ska sändas i ett text-fält
- Göra kommandot send för att sända meddelanden

Kommandot send behöver i denna uppgift enbart lägga till meddelandet från text-fältet till listan över den senaste konversationen. Ingen uppkoppling ska göras.

## Exempel

Några bilder av ett enkelt exempel:

- [Bild 1](#)
- [Bild 2](#)

Se även den färdiga chat-klienten i uppgiften [Mobila enheter med J2ME: Nätverkskopplingar: Stream sockets på klientsidan](#).

## Tips

Om man ska utveckla för CLDC/MIDP så behöver man använda JME Wireless Toolkit (som kräver vanliga J2SE). Vill man rikta in sig på någon speciell maskin så finns det ofta "device emulators" att hämta för dessa på Internet.

Kodningen skiljer sig ganska mycket mot kodning under J2SE, speciellt för:

- GUI-kod, se paketet javax.microedition.lcdui
- Nätverks-kod, se paketet javax.microedition.io
- Lagrings-kod, se paketet javax.microedition.rms

Man måste alltså först sätta sig in i dessa saker.

Om man utvecklar för CLDC/MIDP och har fått något att fungera i JME Wireless Toolkit så betyder inte det att det fungerar direkt i alla maskiner som stöder CLDC/MIDP utan man måste testa och eventuellt specialanpassa koden först. Det beror på att olika maskiner inte har samma implementation av CLDC/MIDP som SUN har i sitt JME Wireless Toolkit, exempelvis så stöder vissa maskiner mer än vad som krävs i CLDC/MIDP, SonyEricssons (gamla) P800 stöder exempelvis nätverks-funktioner med sockets trots att standarden enbart kräver HTTP-uppkopplingar.

Om man vill kan man även titta på kraftfullare PersonalJava som stöds i en del mobiler och PDA'er, exempelvis SonyEricsson (gamla) P800. För just denna mobil/PDA så måste man hämta Symbian OS 7 UIQ SDK for Sony Ericsson P800 på 500 MByte och del andra saker. Med Symbians SDK får programmen samma look-and-feel som de övriga programmen i mobilen (UIQ) och man kan nå många saker i operativsystemet. PersonalJava kommer dock att ersättas av CLD/PBP och CLD/PP.

# Mobila enheter med J2ME: Intro: Lagring med Record Management System

## Allmänt

Uppgiften belyser hur man lagrar information i J2ME med Record Management System.

## Uppgift

Bygg ut uppgiften [Mobila enheter med J2ME: Intro: Grafiska gränssnitt och händelsehantering](#) så att programmet sparar det som anges i text-fälten:

- ☐ Host
- ☐ Port

Nästa gång programmet startas så ska dessa värden hämtas från lagringsplatsen och visas i text-fälten som defaultvärden.

## Exempel

Se den färdiga chat-klienten i uppgiften [Mobila enheter med J2ME: Nätverkskopplingar: Stream sockets på klientsidan](#).

## Tips

Inget tips ges.

# Mobila enheter med J2ME: Nätverkskopplingar: Stream sockets på klientsidan

## Allmänt

Uppgiften belyser hur man använder nätverkskopplingar med stream sockets på klientsidan i J2ME.

## Uppgift

Bygg ut uppgiften [Mobila enheter med J2ME: Intro: Lagring med Record Management System](#) så

att:

- Kommandot connect även gör så att en uppkoppling sker till en chat-server
- Kommandot send gör så att meddelandet i text-fältet sänds
- Meddelanden kan tas emot och visas

## Exempel

Ett enkelt exempel kan köras enligt följande:

Hämta filerna:

- [MANIFEST.MF](#)
- [chat\\_client.jar](#)
- [chat\\_client.jad](#)

Kör programmet med:

- Dubbelklicka på filen chat\_client.jad

Man kan även köra programmet på en modern mobil enhet och då installera det på ett av två sätt:

- Via den egna datorn:
  - Hämta filerna ovan till datorn
  - Anslut den mobila enheten till datorn via WLAN, bluetooth, infrarött eller USB
  - Installera filerna ovan med det installationsprogram (som körs på datorn) som följde med vid köpet av den mobila enheten
- Direkt till den mobila enheten:
  - Anslut den mobila enheten till Internet via WLAN, 3G eller GPRS
  - Starta en browser på den mobila enheten och gå till URL  
»[http://atlas.dsv.su.se/~pierre/i/05\\_ass/ip4/a\\_3\\_2\\_1\\_1/bin/chat.jad](http://atlas.dsv.su.se/~pierre/i/05_ass/ip4/a_3_2_1_1/bin/chat.jad)«
  - Godkänn installationen via Over-The-Air Provisioning (OTA)

## Tips

Se uppgiften [Lågnivåkopplingar: Stream sockets: Klient-sidan](#) på kursen Internetprogrammering I för tips hur man ansluter sin chat-klient till en chat-server. Man kan även använda chat-servern i uppgiften [Mobila enheter med J2ME: Nätverkskopplingar: Stream sockets på serversidan](#). Om det inte finns någon att prata med så är ett tips att använda:

- Host: atlas.dsv.su.se
- Port: 4456

Det är en robot som pratar Engelska.

# Mobila enheter med J2ME: Nätverkskopplingar: Stream sockets på serversidan

Allmänt

Uppgiften belyser hur man använder nätverkskopplingar med stream sockets på serversidan i J2ME.

## Uppgift

Gör ett J2ME-program som har samma funktion som programmet i uppgiften [Lågnivåkopplingar: Stream sockets: Server-sidan](#) på kursen Internetprogrammering I.

## Exempel

Ett enkelt exempel kan köras enligt följande:

Hämta filerna:

- [MANIFEST.MF](#)
- [chat\\_server.jar](#)
- [chat\\_server.jad](#)

Kör programmet med:

- Dubbelklicka på filen chat\_server.jad

Man kan även köra programmet på en modern mobil enhet enligt uppgiften [Mobila enheter med J2ME: Nätverkskopplingar: Stream sockets på klientsidan](#).

## Tips

Inga tips ges.

# Mobila enheter med J2ME: Nätverkskopplingar: Datagram sockets

## Allmänt

Uppgiften belyser hur man använder nätverkskopplingar med datagram sockets i J2ME.

## Uppgift

Gör ett program med ett grafiskt användargränssnitt som implementerar mottagardelen i ett enkelt distribuerat ritprogram (whiteboard), se uppgiften [Lågnivåkopplingar: Datagram sockets: Unicast](#) på kursen Internetprogrammering I. Allt som en klient från en stationär enhet sänder ska alltså ritas upp på den mobila enheten. Frivillig utökning är att även implementera en ritfunktion på den mobila enheten.

## Exempel

Ett enkelt exempel kan köras enligt följande:

Hämta och exempelprogrammet från uppgiften [Lågnivåkopplingar: Datagram sockets: Unicast](#) på kursen Internetprogrammering I.

- Kör igång programmet med: `java -jar Draw.jar 2001 localhost 2000`

Hämta filerna:

- [MANIFEST.MF](#)
- [datagram\\_client.jar](#)
- [datagram\\_client.jad](#)

Kör programmet med:

- Dubbelklicka på filen `datagram_client.jad`

Rita sen i programmet som kör på den stationära enheten.

Man kan även köra programmet på en modern mobil enhet enligt uppgiften [Mobila enheter med J2ME: Nätverkskopplingar: Stream sockets på klientsidan](#), notera att programmet kör på port 2000.

## Tips

Programmet på den stationära enheten sänder varje ritad punkt som en `String` på formen:

- x-koordinat + mellanslag + y-koordinat: `x y`

I J2ME finns inte stöd för `split` på strängar så man får implementera det själv, exempelvis:

```
char[] charArray = message.toCharArray();
boolean flag = false;
String s1 = "", s2 = "";
for(int i = 0; i < charArray.length; i++) {
 if(charArray[i] == ' ') {
 flag = true;
 } else {
 if(!flag) {
 s1 += charArray[i];
 } else {
 s2 += charArray[i];
 }
 }
}
```

# Mobila enheter med J2ME: XML-baserade distribuerade processer: XML

## Allmänt

Uppgiften belyser hur man använder XML i J2ME.

## Uppgift

Gör ett J2ME-program som har samma funktion som programmet i uppgiften [XML: Processande:](#)

[Högnivå-baserat med JAXP och JDOM.](#)

## Exempel

Inget exempel ges.

## Tips

Använd ett tillbehör för XML, exempelvis »[kXML Project](#)«.

Tillbehör ska läggas under lib-katalogen under det aktuella projektet i Java Micro Edition Wireless Toolkit.

# Mobila enheter med J2ME: XML-baserade distribuerade processer: XML-RPC

## Allmänt

Uppgiften belyser hur man använder XML-RPC i J2ME.

## Uppgift

Gör ett J2ME-program som har samma funktion som programmet i uppgiften [Distribuerade processer: XML-RPC och JAX-RPC: Klient sidan](#).

## Exempel

Ett enkelt exempel kan köras enligt följande:

Hämta filerna:

- [MANIFEST.MF](#)
- [xmlrpc.jar](#)
- [xmlrpc.jad](#)

Kör programmet med:

- Dubbelklicka på filen xmlrpc.jad

Man kan även köra programmet på en modern mobil enhet enligt uppgiften [Mobila enheter med J2ME: Nätverkskopplingar: Stream sockets på klient sidan](#).

## Tips

Använd ett tillbehör för XML, exempelvis »[kXML-RPC Project](#)«.

Tillbehör ska läggas under lib-katalogen under det aktuella projektet i Java Micro Edition Wireless

Toolkit.

# Mobila enheter med J2ME: XML-baserade distribuerade processer: Web Services med SOAP och WSDL

## Allmänt

Uppgiften belyser hur man använder Web Services med SOAP och WSDL i J2ME.

## Uppgift

Gör ett J2ME-program som har samma funktion som programmet i uppgiften [Distribuerade processer: Web Services med SOAP och WSDL: Klientsidan](#).

## Exempel

Ett enkelt exempel kan köras enligt följande:

Hämta filerna:

- [MANIFEST.MF](#)
- [soap.jar](#)
- [soap.jad](#)

Kör programmet med:

- Dubbelklicka på filen soap.jad

Man kan även köra programmet på en modern mobil enhet enligt uppgiften [Mobila enheter med J2ME: Nätverkskopplingar: Stream sockets på klientsidan](#).

## Tips

Använd ett tillbehör för XML, exempelvis »[kSOAP Project](#)«.

Tillbehör ska läggas under lib-katalogen under det aktuella projektet i Java Micro Edition Wireless Toolkit.

# Mobila enheter med J2ME: Extensioner

## Allmänt

Uppgiften belyser hur man använder extensioner i J2ME.

# Uppgift

Gör ett J2ME-program som använder en eller flera extensioner. Man kan exempelvis använda de extensioner som finns med i JME Wireless Toolkit:

- Wireless Messaging API (WMA)
- Mobile Media API (MMAPI)
- Mobile 3D Graphics API for J2ME
- J2ME Web Services Specification
- Java APIs for Bluetooth
- PDA Optional Packages for the J2ME Platform

Det finns även andra paket, exempelvis:

För CLDC-konfigurationen:

- Java USB API
- Information Module Profile (IMP)
- Java Technology for the Wireless Industry (JTWI)
- Location API for J2ME
- SIP API for J2ME
- Security and Trust Services API for J2ME (SATSA)
- J2ME Web Services APIs (WSA)

För CDC-konfigurationen:

- J2ME RMI Optional Package (RMI OP)
- JDBC Optional Package for CDC/Foundation Profile API

För mer info, se »[Introduction to Mobility Java Technology](#)«.

## Exempel

Inget exempel ges.

## Tips

Olika mobila enheter stöder olika varianter av CLDC och MIDP och har olika stöd för extensioner. Trenden är dock att senare modeller stöder mer, se exempelvis »[SonyEricsson K750i](#)« som har ett mycket bra stöd (både CLDC 1.1, MIDP 2.0 och de flesta extensioner som ingår i JME Wireless Toolkit. Ett bra ställe för att ta reda på vad som stöds i olika modeller är »[J2ME Polish](#)«.

Notera att alla API-beskrivningar inte finns med i JME Wireless Toolkit utan att man får ladda ner dem separat.

Det finns många exempel (med källkod) på program som använder extensioner i JME Wireless Toolkit.

Man kan bygga enormt många olika saker, ett tips är exempelvis Push-to-talk eller ett system där en användare från en HTML-sida kan kontakta en server på en mobil enhet (via ett HTTP-serversides-program) med text eller ljud (med Java Sound i ett äpple). Fältet är fritt!

# Gesällprov

# Allmänt

Gesällprovet är en uppgift för den egna kreativiteten och är ett fritt valt (men obligatoriskt) arbete där du får chansen att visa vad du går för konstruktionsmässigt (istället för minnesmässigt som på en tenta) i "lugn och ro".

## Uppgift

Följande krav måste uppfyllas:

- Ska baseras på de tekniker som kursen bygger på men får givetvis även innehålla tekniker från andra områden
- Ska vara nätverksbaserat
- Ska vara bra både till både form och funktion
- Ska inte vara en av de tidigare uppgifterna på kursen men gärna en utökning av en eller flera av dessa

## Tips

Huvudsaken är att man gör något som är kul! Några exempel:

- Implementera något som använder exempelvis:
  - JXTA
  - CORBA
  - Voyager
  - Jini
  - DeJay
- Implementera ett eget system för distribuerade processer
- Implementera något i J2ME som använder RMI, XML-RPC eller Web Services

## Hjälp

På denna sida finns alla hjälpresurser på kursen.

## Handledning

Kör man fast så får man hjälp via news och e-post. Notera dock att det inte är någon handledning på:

- Kunskap som ingår i förkunskapskraven (kompileringsfel med mera)
- Gesällprovet

News är tänkt som en plattform för diskussion mellan studenterna men handledarna kommer att besvara de frågor som fortfarande är obesvarade senare under de dagar som det är handledning. E-post går direkt till handledarna och de kommer att försöka svara snabbt men eftersom de ibland får väldigt många frågor kan svaren dröja. Notera att handledarna enbart arbetar inom angivna handledningstider vilket medför att man kanske inte hinner få svar samma dag på en sent ställd fråga.

## News

News bör användas för frågor som kan intressera andra:

- Läs först igenom de tidigare frågorna och svaren
- Hittar man inte en lösning så:
  - Ange ett väl beskrivande ämne för brevet
  - Var mycket tydlig vid beskrivningen av problemet
  - Sänd inte in mycket kod eller fullständiga program

News-server:

- Adress: atlas.dsv.su.se
- Port: 1199 (notera att detta inte är default numret och därför måste sättas manuellt)

News-grupper:

- public.ip4.a.1 för frågor om uppgift 1
- public.ip4.a.2 för frågor om uppgift 2
- ... och så vidare ...
- public.ip4.misc för övriga frågor

Se även [news](#) för mer detaljerad information om hur man konfigurerar en news-klient.

## E-post

E-post bör användas för specifika frågor utan intresse för andra:

- Sänd inte e-post direkt till lärarna utan använd länken nedan
- Om man ska sända med kod:
  - Sänd endast med de delar som är relevanta för frågan
  - Koden ska vara välkommenterad och välstrukturerad
- Använd "svara till alla" på eventuella följdfrågor

E-post-server:

- Ingående e-brev (POP3):
  - Adress: mail.dsv.su.se
  - Port: 110
- Utgående e-brev (SMTP):
  - Adress: mail.dsv.su.se
  - Port: 25

Notera att om man sitter uppkopplad utanför DSV's domän (exempelvis via Telia Bredband) så ska man använda leverantörers SMTP-server för utgående brev.

E-post länk:

- [Klicka här för handledning via e-post](#)

Se även [e-post](#) för mer detaljerad information om hur man konfigurerar en e-post-klient.

## FAQ

Här följer de frågor och svar som brukar vara vanligast:

### XML

#### Fråga/svar 1

Vad är XML och varför är det bra?

XML är ett standard-format för data och det finns många fördelar med detta:

- En maskin-människa fördel är att man kan skilja på data och presentation, samma data kan presenteras på olika sätt (fokus på kursen Internetprogrammering III)
- En maskin-maskin fördel är att allt fler program kommunicerar i olika skikt och har man en standard för datat som sänds mellan dessa program så får man (1) en kontroll av de data som sänds mellan program uppfyller en viss standard, (2) ett enkelt sätt att laborera med data i programmen och (3) en möjlighet att lätt byta ut program (fokus på kursen Internetprogrammering IV)

Data givet i XML-form parsas och man kan göra detta på två sätt:

1. Man kontrollerar att data givet i ett XML-dokument följer den XML-standard som finns och om ett XML-dokument uppfyller detta krav så kallas det för välutformat
2. Man kontrollerar (1) och att det följer vissa direktiv vi själva (eller någon annan) specificerat i en så kallad DTD eller XML-Schema och ett XML-dokument som uppfyller detta strängare krav kallas valid

Man parsar även för att kunna plocka isär och komma åt data i ett XML-dokument. Tekniskt sett så kan man göra parsningen på två sätt:

- SAX-parsning är snabb och men har inte ett fullständig representation av ett XML-dokument i minnet
- DOM-parsning är långsammare men har en fullständig representation av ett XML-dokument i minnet

De API som man använder för dessa två sätt att parse är anpassade för att fungera med flera olika språk och är ofta ganska svåra och icke-intuitiva att använda som Java-programmerare. Därför finns det andra API som exempelvis JDOM som är ett API som ligger ovanpå SAX och DOM och som möjliggör en mycket enklare hantering av XML med Java eftersom de integrerar mycket väl med Java och de designmönster som Java-programmerare är vana att använda.

XML är ett område som växer enormt fort och Java och XML är ett perfekt par för plattformsoberoende processande.

## E-post

På denna sida beskrivs hur man konfigurerar e-post-klienten. Följande är ett exempel på hur man konfigurerar e-post i Outlook Express, förfarandet är ungefär detsamma i andra moderna e-post/news-klienter:

- Klicka på Verktyg/Konton
- Klicka på Add/Mail
- Skriv in namn och e-post-adress
- Skriv in adressen för inkommande och utgående e-post-servrar
- Skriv in användarnamn och lösenord
- Klicka på Avsluta

Vill man använda exempelvis hotmail istället så kan man göra följande för att få all e-post som kommer till DSV vidareasänd till hotmail-kontot:

- telnet triton.dsv.su.se
- emacs .forward
- Skriv in hotmail-adressen, spara och avsluta emacs
- Logga ut

De som använder ett e-postkonto på DSV kan även använda »[DSV: Webmail](#)«.

# News

På denna sida beskrivs hur man konfigurerar en news-klient. Följande är ett exempel på hur man konfigurerar news i Outlook Express, förfarandet är ungefär detsamma i andra moderna e-post/news-klienter:

- Klicka på Verktyg/Konton
- Klicka på Lägg till/News
- Skriv in namn och e-post-adress
- Ange adressen för news-servern
- Klicka på Avsluta
- Klicka på mappen Alla och markera aktuell news-server
- Klicka på Egenskaper
- Klicka på Avancerat
- Ange port nummer till news-servern
- Klicka på ok och sen Stäng
- Svara Ja på frågan Vill du ladda ner nyhets-grupper från news-kontot
- Dubbelklicka på alla grupper som är av intresse
- Klicka på ok

Default så visar Outlook Express bara de senaste meddelandena. Vill man ändra på det så:

- Klicka på Verktyg/Alternativ
- Klicka på mappen Underhåll
- Ändra vid Ta bort news meddelanden efter 5 dagar (exempelvis genom att ta bort markeringen helt)

Anledningen till att vi använder news istället för FirstClass är att news är en öppen mycket vanlig standard på Internet och att det även ingår som några förslag på gesällprov.

# Redovisning i detalj

Denna beskrivning utgår från att man använder DSV's datorer men man kan använda andra ställen än DSV för redovisningen. Notera dock att andra ställen kan använda en annan huvudkatalog för webpublicering och andra rättigheter på kataloger och filer.

## Steg för steg

1. Starta ett [filöverföringsprogram](#) och koppla dig till ditt DSV/Unix/Linux-konto
2. Skapa katalogstrukturen nedan och ladda upp filerna i denna struktur
3. Sätt rättigheterna på kataloger och filer så att allt blir läsbart från en browser
4. Testa att allt kan nås från en browser och att all källkod ser bra ut från en browser
5. Ange på redovisningssidan:
  - URL'en till denna katalogstruktur: `http://www.dsv.su.se/~x/ip4/`, där x är användarnamnet
  - Att allt är klart för redovisning

## Katalogstruktur

Man måste ha **exakt följande katalogstruktur** men använder man en annan plats än DSV så måste inte ww katalogen ha det namnet.

```

www
ip4
 ip4.zip (ip4 zippad, ej innehållande ip4.zip givetvis)
 1.1.1
 källkodsfiler för uppgift 1.1.1
 1.1.2
 källkodsfiler för uppgift 1.1.2
 1.1.3
 källkodsfiler för uppgift 1.1.3
 1.2.1
 källkodsfiler för uppgift 1.2.1

... osv ...

4
 källkodsfiler för gesällprovet
 description
 beskrivning i text och bild av gesällprovet

```

Observera:

- Katalogen `www` måste ha stora/versala bokstäver
- Skapa inte kataloger för frivilliga uppgifter som inte är gjorda
- Lägg ingen fil med namnet `index.html` eller `index.htm` direkt i uppgiftskatalogerna eftersom man då inte kan lista innehållet i dessa kataloger från Internet

## Beskrivning av gesällprovet

Gesällprovet ska beskrivas generellt med både text och bild (på programmet under körning) på något av följande sätt:

- HTML-dokument med bilderna inkluderade
- PDF-dokument med bilderna inkluderade
- TXT-dokument med bilderna separata

Ett HTML-dokument kan göras antingen för hand i en vanlig text-editor eller med ett högnivåverktyg.

För att skapa PDF-filer i Win32 kan man exempelvis:

1. Lägg till en skrivare, exempelvis HP Color LaserJet PS
2. Högerklicka på skrivaren och välj Egenskaper, klicka på Portar och kryssa i FILE
3. Skriv ut Word-dokumentet till denna skrivare varefter en fil `<namn>.prn` skapas
4. Gå till »[PS2PDF.com](http://PS2PDF.com)« och konvertera filen till PDF

En bild (använd inte BMP-format) av ett program under körning fås enligt:

- Win32: Tryck på `<CTRL><ALT><PRINT SCRN>` samtidigt, vilket medför att bilden på programmet läggs i urklippsbufferten som sedan lätt kan klistras in i ett HTML-dokument i en HTML-editor
- Linux/Unix: Skriv i kommandofönstret `import dump.jpeg` och klicka sen på det fönster som man vill ha en bild på (ändrar man till en annan extension så får man bilden som en annan bildtyp)

All eventuell java-kod i gesällprovet ska beskrivas med verktyget `javadoc`.

## Efter erhållet betyg

När man fått G/VG eller 3/4/5 på ett arbete så bör man ändra rättigheterna på källkodsfilerna så att ingen annan kan läsa dem.

# Nyheter

På denna sida visas alla uppdateringar.

| Tid                 | Rum       | Uppdatering                                                                      |
|---------------------|-----------|----------------------------------------------------------------------------------|
| 2005-03-29<br>10:10 | Allt      | Allt klart utom vissa uppgifter och exempel                                      |
| 2005-03-29<br>20:05 | Feedback  | Momentrådet uppdaterat                                                           |
| 2005-04-26<br>13:30 | Uppgifter | Uppgifterna och exemplen online (har lagts till gradvist sedan kursen startades) |
| 2005-04-27<br>05:55 | Hjälp     | FAQ uppdaterad                                                                   |

# Internetsäkerhet

På denna sida finns hela kursen utom utom OH-bilderna, exemplen och tentorna.

# Intro

Välkommen Pierre! På denna sida beskrivs kursen generellt.

# Kurskoder

Kursen är på 4 poäng och går för följande universitet/högskolor:

- SU med kod INS
- KTH-FoV med kod 2I4105

# Innehåll

Denna kurs lär ut Internetprogrammering fördjupat mot kunskaper om säkerhet i distribuerade informationssystem där systemen kommunicerar över osäkra informationskanaler:

- Säkerhets policy
- Autentisering och nyckeldistribution i publika nät (asymmetriska metoder med publika och privata nycklar, certifikat, signering, verifiering, Public-Key Infrastructure (PKI) och Trusted Third Party (TTP) med mera):
  - Program (kod signering och verifikation)
  - Meddelanden och godtyckliga filer
  - Nätverk
- Konfidentialitet (symmetriska metoder med hemliga nycklar, kryptering, dekryptering med mera)
- Integritet (signerade integritetskoder (MIC/MAC))
- Oavvislighet (asymmetriska metoder + certifikat auktoritet (CA))
- Protokoll och specifikationer för säker kommunikation:
  - WWW och Web Services säkerhet (SSL/TLS/SET/3dSecure)
  - E-post säkerhet (PGP, S/MIME)
  - Virtuella Privata Nätverk (VPN) teknik (IPSec)
- Portskanning, trojanska hästar och virus
- Brandväggar

Kursen går igenom detta både teoretiskt och praktiskt och är uppdelad i två delar. Den teoretiska delen ligger till grund för genomförande av den praktiska delen.

Språk: Java.

## Undervisningsform

Pedagogiken för denna kurs är:

- (1) Frihet, (2) tävling och (3) samarbete --> glädje och motivation --> ett bra resultat!

Frihet, tävling och samarbete gör att man går från passivt mottagande till aktivt sökande. Genom frihet tillåts man själv bli nyfiken och ställa frågor innan man får svar. Genom tävling motiveras man att göra sitt bästa. Detta leder fram till glädje och motivation och därmed till ett bra resultat.

Observera! Detta är en distanskurs och kräver förmåga till självständigt arbete. Kursen har dock, till skillnad mot vanliga distanskurser, kontinuerligt och direkt stöd via handledning på distans. Kursen är krävande och man bör inte läsa andra kurser samtidigt.

## Examination

Examinationen på kursen består av:

- En tentamen på den teoretiska delen, 2 poäng (U/G/VG)
- Ett antal praktiska uppgifter, 2 poäng (R/G/VG) (R/3/4/5):
  - Ett antal bestämda programmeringsuppgifter
  - Ett fritt valt gesällprov

Notera att man måste lämna in lösningar på de praktiska uppgifterna som stämmer med **nuvarande läsårs praktiska uppgifter**. Kurserna utvecklas dock bara i små steg med gradvisa förfiningar så det innebär inte att man måste göra om allt om man inte har fått betyg på de praktiska uppgifterna vid den sista deadline i Augusti.

## Litteratur

Litteraturen är:

- Teoretisk del: »[Network Security Essentials \(andra utgåvan\)](#)« skriven av Stallings och utgiven av »[Prentice Hall](#)«
- Praktisk del: »[Java Security \(andra utgåvan\)](#)« skriven av Oaks och utgiven av »[O'Reilly](#)«

Billigaste boken hittas med hjälp av »[Bokfynd.nu](#)«. Vi rekommenderar »[Amazon](#)« och »[WHSmith](#)« i England som brukar ha bra priser och snabba leveranser. Köp inte böcker utanför EU eftersom det tar lång tid (2-3 veckor) att få böckerna och sen får man en ganska stor extra avgift från tullen.

## 4 steg för att klara denna kurs

Gör följande för att klara denna kurs:

1. Registrera dig hos »[DSV: Studievägledningen](#)« som har e-post [svl@dsv.su.se](mailto:svl@dsv.su.se)
2. Köp och läs kurslitteraturen
3. Lös och redovisa uppgifterna
4. Anmäl dig till tentamen (minst en vecka innan den går) och klara denna

# Lärare

Vi önskar alla en givande och rolig kurs!

- »[Ulrika Norman](#)«, ansvarig för teoretisk del
- »[Pierre A. I. Wijkman](#)«, ansvarig för praktisk del
- Handledarna

Lycka till!

# Schema

På denna sida finns alla tider och platser för kursen. Alla tider är preliminära och kan ändras.

## Föreläsningar

| Tid                 | Rum                 | Föreläsare    | Innehåll                              |
|---------------------|---------------------|---------------|---------------------------------------|
| 2004-11-23 kl 13-15 | F3, Electrum, Kista | Ulrika Norman | Intro till kommunikationssäkerhet     |
| 2004-11-25 kl 13-15 | Sal B, Forum, Kista | Ulrika Norman | Nyckeldistribution, certifikat        |
| 2004-11-30 kl 13-15 | Sal B, Forum, Kista | Ulrika Norman | Public key infrastructure             |
| 2004-12-02 kl 13-15 | Sal B, Forum, Kista | Ulrika Norman | Säkra tillämpningar, e-post, e-handel |

## Handledning

All handledning sker via e-post och news (se sidan Hjälp).

Samtliga studenter har handledning:

- Kursens 1:a - 4:e vecka: samtliga vardagar kl 10-17

För de som arbetar på DSV finns info om prio-tider för labbsalarna under »[DSV: Scheman](#)«.

## Examination

Notera följande angående tid för rättning:

- Det brukar ta upp till **3 veckor tills rättningen är klar**
- Ansvarig kan **inte rätta vissa personers uppgifter före andras**, exempelvis de som har bråttom på grund av CSN
- När man fått betyg på kursen så dröjer det **ytterligare tid** tills allt är infört i Daisy och Ladok

Var god **e-posta inte ansvarig om detta** för att försöka påskynda processen, det är väldigt många studenter på samtliga kurser.

Man får välja tid för tenta och deadline oberoende av om man går på SU eller KTH och så vidare. Typen nedan är enbart en rekommendation.

## Deadlines för uppgifterna

Det finns flera deadlines för inlämning av uppgifterna och man väljer själv en som passar. Man redovisar samtliga uppgifter direkt eftersom det av praktiska skäl tyvärr inte går att delrätta och ge

återkoppling efter hand som deluppgifterna blir klara. De som går en + kurs kan dock lämna in ordinarie uppgifter och + uppgifter separat vid olika deadlines. Varje deadline är normalt daterad till dagen innan tentamen.

| Tid              | Rum      | Typ            |
|------------------|----------|----------------|
| 2004-12-19 kl 24 | Internet | Deadline 1     |
| 2005-02-18 kl 24 | Internet | Deadline 2     |
| 2005-08-31 kl 24 | Internet | Sommardeadline |

Observera:

- Redovisade uppgifter **rättas endast i samband med dessa deadlines och uppskov ges aldrig**
- Får man **rest** så **rättas den igen först vid nästa deadline**
- Betyget påverkas inte beroende på vilken deadline man väljer
- Man måste inte välja en deadline som ligger i anslutningen till en tenta
- Handledningen slutar ofta innan deadline 1 (det finns tyvärr inga resurser för en längre handledningsperiod)

## Tentamen

Inga hjälpmedel är tillåtna på tentamen. Man bör ha gjort klart de obligatoriska uppgifterna för 5 poängskursen (exklusive gesällprovet) innan tentamen eftersom tentamen kommer att fråga om dessa (detta är dock inget strikt krav för att få gå upp på tentamen). Vissa frågor kan vara mer generella och vissa kan vara mer tekniska och fråga om hur viss kodning (med avseende på uppgifterna) görs.

| Tid                 | Rum          | Typ            |
|---------------------|--------------|----------------|
| 2004-12-20 kl 09-13 | Forum, Kista | Tentamen       |
| 2005-02-19 kl 10-14 | Forum, Kista | Omtentamen     |
| 2005-08-23 kl 09-13 | Forum, Kista | Sommartentamen |

Observera:

- Man måste anmäla sig i Daisy **senast en vecka innan tentamen går**, se »[DSV: Att tentera vid DSV](#)«
- Man måste inte välja en tenta som ligger i anslutningen till en deadline
- Sal meddelas i Daisy eller på telefonsvarare 08 674 70 04 senast en dag innan tentamen, se även »[DSV: Scheman](#)«

Var god **e-posta inte ansvarige** (han har inga befogenheter här) **om detta**, utan kontakta istället »[DSV: Studentexpeditionen](#)«.

# Kursmaterial

På denna sida finns information om arbetssätt och vilka mjukvaror man behöver. Här finns även OH-bilder och tidigare tentor.

## Arbetssätt och mjukvaror

På denna kurs kan man arbeta på följande sätt:

- På DSV
- På distans online kopplad via telnet eller SSH till sitt DSV/Unix-konto:

- [Terminalprogram](#)
- På distans offline med det egna systemet konfigurerat för allt som behövs:
  - [Editor](#)
  - [Java](#)
  - Policyfiler för obegränsad kryptering/dekryptering läggs under katalogen  
 <java\_home>/jre1.5.0/lib/security (på Win32 ofta C:\Program Files\Java\jre1.5.0\lib\security):
    - [local\\_policy.jar](#)
    - [US\\_export\\_policy.jar](#)
  - [Filöverföringsprogram](#)

Det senare är det klart roligaste och mest lärorika sättet eftersom man då måste installera allt som behövs på den egna datorn. Notera att ingen handledning ges på detta eftersom det finns alltför många olika varianter av konfigurationer.

## OH-bilder

OH-bilderna finns enbart här:

- [f1\\_theory.pdf](#)
- [f2\\_theory.pdf](#)
- [f3\\_theory.pdf](#)
- [f4\\_theory.pdf](#)
- [kerberos\\_intro.pdf](#)

## Tidigare tentor

Tidigare tentor finns enbart här:

- [2003-12-15.pdf](#)
- [2004-02-14.pdf](#)

# Terminalprogram

På denna sida beskrivs hur man använder terminalprogram.

## Terminalserver

Om man behöver logga in och köra kommandon på en avlägsen dator så använder man ett terminalprogram/klient. På DSV används följande dator som terminalserver:

- `triton.dsv.su.se`

Denna dator kör Linux och har både telnet- och SSH-server igång.

Om man inte släpps in kan det bero på att DSV's brandvägg inte släpper in just det IP-nummer som man använder. Då kan man kontakta [Christer Jansson](#) som kan registrera IP-nummret i brandväggen så att man kommer förbi den.

## Telnet

I Win32 och Linux så kan man köra ett Terminal-program direkt i ett kommando-fönster:

- telnet triton.dsv.su.se

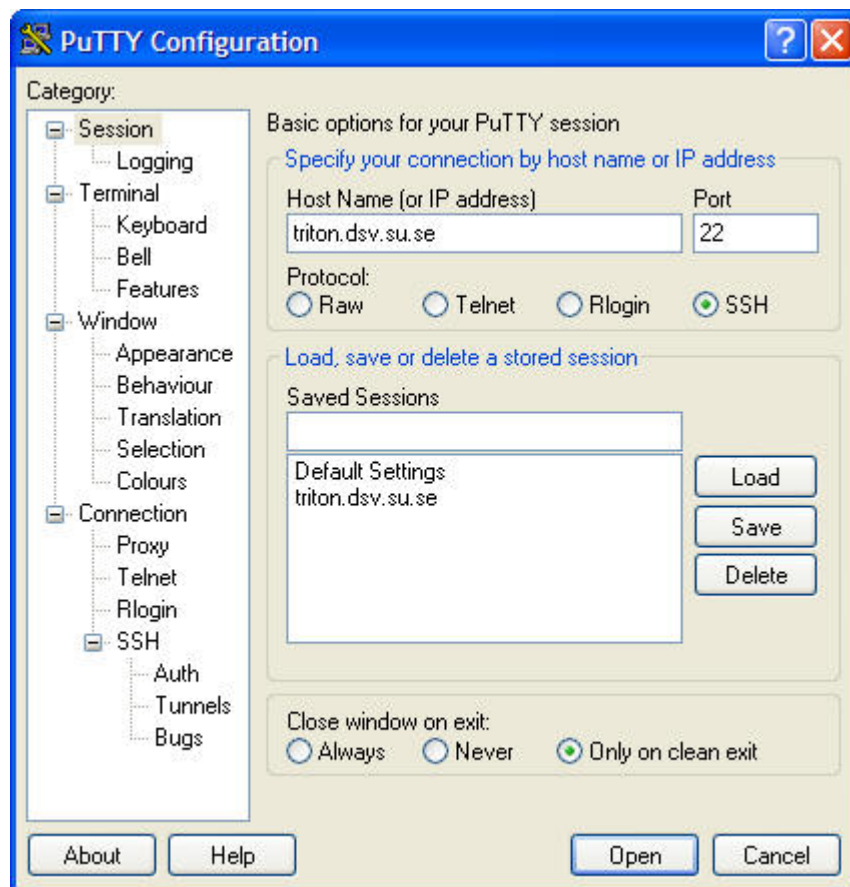
Man anger sedan användarnamn och lösenord.

## SSH

När man använder telnet så sänds all information okrypterad över Internet och det kan vara farligt. Vill man undvika det kan man istället använda SSH och följande program rekommenderas:

- Win32: »[Putty](#)«
- Linux: ssh

Så här ser det ut när man startar Putty:



## Editor

På denna sida beskrivs det verktyg man använder för att skapa kod med.

## Intro

Man kan använda allt från en enkel editor till en avancerad grafisk utvecklingsmiljö. Valet är fritt! En avancerad grafisk utvecklingsmiljö kan vara något svårare att komma igång med men är överlägsen när man väl lärt sig den. En grafisk utvecklingsmiljö kan generera vissa delar av koden i ett program automatiskt, exempelvis den kod som tar han om det grafiska användargränssnittet, vilket snabbar upp konstruktionsarbetet betydligt. En stor nackdel med detta är att koden tenderar att bli omfattande och främmande för en nybörjare och det blir mycket svårt att anknyta sin egen kod till den kod som verktyget har skapat. Det bästa är att börja med en enkel editor och sedan gå

över till en grafisk utvecklingsmiljö.

Observera att koden ska vara läsbar direkt i en vanlig browser för enklare rättning. Man bör därför inte använda en editor som använder kontrolltecken och tabbar för att indentera koden.

## Enkla

- Notepad (Win 32)
- »[Pico](#)« (Unix)
- »[Emacs](#)« (Win32 och Unix)

Använder man Notepad så måste man spara filerna genom att skiva citationstecken runt filnamnet "X.java" när man namnger och sparar filen, annars sparas filen som X.java.txt.

Använder man emacs för Java bör man inte använda JDE-tillägget eftersom det lätt blir fel med classpath med mera. Kompilera, kör, sätt classpath med mera i ett kommandofönster istället:

- Skriv koden i en editor och spara filen (här X.java)
- Starta ett kommandofönster och gå till katalogen där koden ligger sparad
- Kompilera med `javac X.java`
- Kör med `java X`

Tips Win32:

- Har man Windows95/98/ME så bör man börja med att skriva doskey i kommandofönstret så att man kan använda "uppåtpil" och "nedåtpil" istället för att skriva om allt man skrivit tidigare
- Man kan lägga en genväg till DOS-promten i varje katalog och konfigurera (höger-klicka och välj Egenskaper) så att den startar i samma katalog:
  - Windows 98: Välj Program/Arbetsmapp och skriv en .
  - Windows 2000/XP: Välj Shortcut/Start in och gör detta text-fält tomt
- När det är långa krångliga kommandon så kan man i DOS använda BAT-filer genom att skriva kommandot i en fil med namnet run.bat så kan man sedan köra kommandot med run

Vill man strukturera upp sin XHTML-kod så rekommenderas:

- »[W3C: Tidy](#)«
- »[SourceForge: Tidy](#)«
- »[Charlie's Tidy Add-Ons](#)«

## Mer komplexa

- »[SciTe](#)«, se även »[SciTe Bundled](#)«-versionen
- »[JEdit](#)« (för Java-kod)
- »[HTML-kit](#)« (för XML- och XHTML-kod)

Konfiguration av SciTE:

- Ladda den fil som bestämmer SciTE's funktion: Options / Load Global Options File
- Sen kan man sätta lite olika saker, exempelvis:
  - Använd 3 mellanslag vid intendering: `indent.size=3`
  - Använd inte tabbar vid intendering: `use.tabs=0`
- Spara sen filen

Sessionshantering med SciTE:

- Om man har många filer öppna och vill fortsätta med dessa senare så sparar man sessionen

- med: File / Save Session ...
- När man vill öppna alla filer igen sen så går man till: File / Load Session ...

## Kompletta utvecklingsmiljöer

- »[Eclipse](#)« (för Java-kod)

# Java

På denna sida beskrivs installation av Java och Java API specifikationen samt konfiguration av Java.

## Installation av Java

Java finns i tre huvudversioner:

- Java Micro Edition (J2ME) som är gjord för mobila/trådlösa enheter
- Java Standard Edition (J2SE) som har fokus på klientsidan
- Java Enterprise Edition (J2EE) som har fokus på serversidan

Vi ska i fortsättningen titta på J2SE som finns i sex huvudversioner: 1.0.x, 1.1.x, 1.2.x, 1.3.x, 1.4.x och 1.5.x (kallad 5.x). Det är stor skillnad mellan dessa versioner och på denna kurs används den senaste versionen. Den finns för nerladdning för olika operativsystem:

Version 1.4.2:

- DOS/Windows95/98/ME: »[J2SE 1.4.2](#)«
- Windows NT/2000/XP: »[J2SE 1.4.2](#)«
- Unix/Linux: »[J2SE 1.4.2](#)«
- MacOS: »[J2SE 1.4.2](#)«

Version 1.5.0:

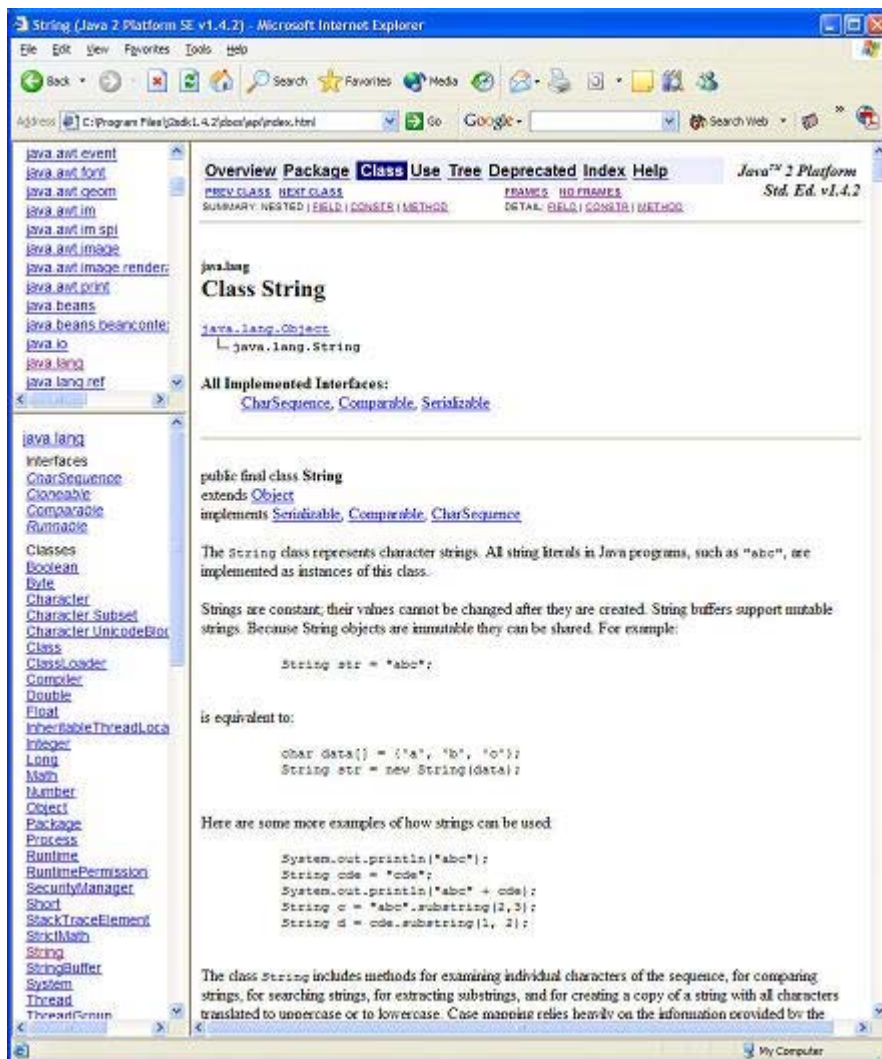
- DOS/Windows95/98/ME: »[J2SE 1.5.0](#)«
- Windows NT/2000/XP: »[J2SE 1.5.0](#)«
- Unix/Linux: »[J2SE 1.5.0](#)«
- MacOS: <finns ej än>

I texten nedan används version 1.4.2 i sökvägar med mera och man får byta denna text så att den passar den version som man själv använder. Efter nerladdning installerar man Java genom att starta installationsprogrammet.

## Installation av Java API specifikationen

Java API specifikationen är en datorbaserad uppslagsbok som behövs eftersom Javas bibliotek av klasser är enormt stort och växer snabbt hela tiden för varje ny version av Java. Om man ska slå upp specifika saker som klass-namn, variabel-namn, metod-namn med mera så är Java API specifikationen till mycket stor hjälp. Notera att man bara får dåligt självförtroende om man börjar med Java API specifikationen utan att ha en grundläggande förståelse och utan att man har testat några programexempel tidigare.

Java API specifikationen kan läsas online men även laddas ner från samma ställe som Java hämtas från om man vill läsa den offline. Uppslagsboken körs i en webb-läsare och ser ut så här:



Notera att det finns en stor mängd tillbehör till Java som inte ingår i någon av huvudversionerna vars API-beskrivningar måste hämtas ner separat.

## Konfiguration av Java

### Enkel körning

För att kunna nå alla program som ingår i J2SE på ett enkelt sätt så sätter man omgivningsvariabeln path till den katalog där programmen ligger. Då kan man exempelvis köra java- och javac-programmen från alla kataloger utan att behöva skriva hela sökvägen till dessa program eftersom systemet nu hittar dessa program automatiskt.

DOS/Windows95/98/ME:

- Antag att vi har installerat java till c:\Program Files\j2sdk1.4.2\_04
- Öppna filen c:\autoexec.bat från ett DOS-fönster genom att skriva edit autoexec.bat
- Skriv följande sist i denna fil:
  - Nya versioner av operativen: set path=%path%;C:\Program Files\j2sdk1.4.2\bin
  - Gamla versioner av operativen: set path=%path%;C:\Progra-1\j2sdk1.4.2\bin
- Spara, avsluta och starta om datorn

Windows NT/2000/XP:

- Antag att vi har installerat java till c:\Program Files\j2sdk1.4.2\_04

- Klicka på Start/Settings/Control Panel/System/Advanced/Environment Variables...
- Under System variables, välj Variable: Path och klicka på Edit
- Lägg till följande sist i Variable Value: ;C:\Program Files\j2sdk1.4.2\_04\bin
- Spara, avsluta och starta om datorn

Unix/Linux:

- Eftersom du är Unix/Linux-användare så ser du detta som en utmaning och vill inte fuska och ha någon instruktionen för detta :-).

MacOS:

- Eftersom du är MacOS-användare så fungerar allt automatiskt och du behöver inte någon instruktionen för detta :-).

## Tillbehör

Det finns ett stort antal tillbehör till de tre huvudversionerna (J2ME, J2SE, J2EE). Dessa tillbehör kallas för standard extensioner eller extra paket och är utbyggnader som finns inom många olika områden, exempelvis säkerhet och media. För att använda ett tillbehör så går man till SUN's eller någon annan leverantörs sida på Internet och hämtar en eller flera filer.

Det finns två huvudtyper av tillbehör:

- Plattformsberoende gjorda enbart med Java
- Plattformsberoende gjorda med Java och native kod

Plattformsberoende tillbehör används för saker där effektiviteten är viktig eller om tillbehöret måste nå plattformsspecifik hårdvara som i exempelvis vid användande av 3D-grafik och/eller andra media-tekniker

Det finns tre huvudsätt att köra Java-program:

- Fristående program
- Äpplen
- Web Start

Varje sätt använder tillbehör på sitt eget speciella sätt och sammanfattningsvis kan man använda tillbehör på följande sätt:

- Fristående program:
  - Plattformsberoende tillbehör:
    - Tillbehöret läggs i speciella kataloger
    - Tillbehöret ligger på ett godtyckligt ställe och man sätter classpath dit
    - Tillbehöret ligger på samma ställe som huvudprogrammet
  - Plattformsberoende tillbehör:
    - Tillbehöret läggs i speciella kataloger
    - Tillbehöret ligger på samma ställe som huvudprogrammet
- Äpplen:
  - Plattformsberoende tillbehör:
    - Tillbehöret läggs i speciella kataloger
    - Tillbehöret ligger på samma ställe som huvudprogrammet
  - Plattformsberoende tillbehör:
    - Tillbehöret läggs i speciella kataloger
- Web Start:
  - Plattformsberoende tillbehör:
    - Tillbehöret läggs i speciella kataloger
    - Tillbehöret ligger på samma ställe som huvudprogrammet
  - Plattformsberoende tillbehör:

- Tillbehöret läggs i speciella kataloger

Vi ska inte gå in på alla detaljer om detta men titta lite generellt på de tre olika huvudsätten att använda tillbehör, se »[Java extensioner med nativelib-filer](#)« för mer detaljerad information.

## Tillbehöret läggs i speciella kataloger

JAR- och eventuella DLL- (eller motsvarande) filer placeras enligt:

- JAR-filer läggs här:
  - För att kunna köra program (JRE): <sökväg>\Java\j2re1.4.2\_04\lib\ext
  - För att kunna utveckla program (kompilera mm) (JDK): <sökväg>\j2sdk1.4.2\_04\lib
- DLL-filer (eller motsvarande) läggs här:
  - För att kunna köra program (JRE): <sökväg>\Java\j2re1.4.2\_04\bin

Notera att sökvägarna ovan kan vara olika på olika maskiner.

När JAR- och DLL- (eller motsvarande) filer ligger på dessa ställen behöver man inte sätta classpath till dessa filer för att kunna köra och utveckla program som är beroende av tillbehöret.

## Tillbehöret ligger på ett godtyckligt ställe och man sätter classpath dit

Om filerna ligger på andra ställen än beskrivet ovan så kan man sätta omgivningsvariabeln classpath till antingen:

- en katalogstruktur med CLASS-filer, eller
- en eller flera JAR-filer

Genom att sätta classpath så berättar man för programmen javac, java med flera var detta tillbehör finns. Notera att man även kan sätta classpath direkt när man använder javac eller java.

DOS/Windows95/98/ME:

- I filen C:\autoexec.bat skriver man på sista raden set classpath=%classpath%;c\x\y

Windows NT/2000/XP:

- Gå till Kontrollpanelen/System/Avancerat/Omgivningsvariabler och sätt eller uppdatera classpath.

Unix/Linux:

- Man sätter classpath olika beroende på vilket skal man använder:
  - C-skäl: setenv classpath <sökväg>
  - Bourne-skäl: classpath=<sökväg> och sen: export classpath
  - Korn-skäl: export classpath=<sökväg>
  - Bash-skäl: export classpath=<sökväg>

Detta kan man lägga i den fil som kör automatiskt (som har olika namn i olika skal) när man loggar in.

MacOS:

- Se Unix/Linux

## Tillbehöret ligger på samma ställe som huvudprogrammet

Man kan slutligen lägga tillbehöret på samma ställe som huvudprogrammet. Är tillbehöret packat i en JAR-fil så packar man upp det i denna katalog. Mer tydligt, antag att man har sina Java-program här:

- C:\x\y

Då ska man lägga tillbehörskatalogen z här:

- C:\x\y\z

En nackdel med detta sätt är att man måste ha flera kopior av tillbehöret om man ska använda det på flera ställen.

## Filöverföringsprogram

På denna sida beskrivs hur man använder filöverföringsprogram.

### Intro

För att kunna sända filer mellan olika datorer använder man filöverföringsprogram och vanliga standarder för det är FTP, SFTP och FTPS.

Ett vanligt fall är när man har HTML-filer som man vill sända till DSV för att synliggöra dem på Internet. Man använder då sitt DSV/Unix/Linux-konto och lägger allt under `www`-katalogen. När filerna är överförda måste man även sätta rättigheterna på filerna och katalogerna:

- Kataloger: `chmod 755`
- De flesta filtyperna: `chmod 744 *`
- CGI-filer: `chmod 711 *.cgi`

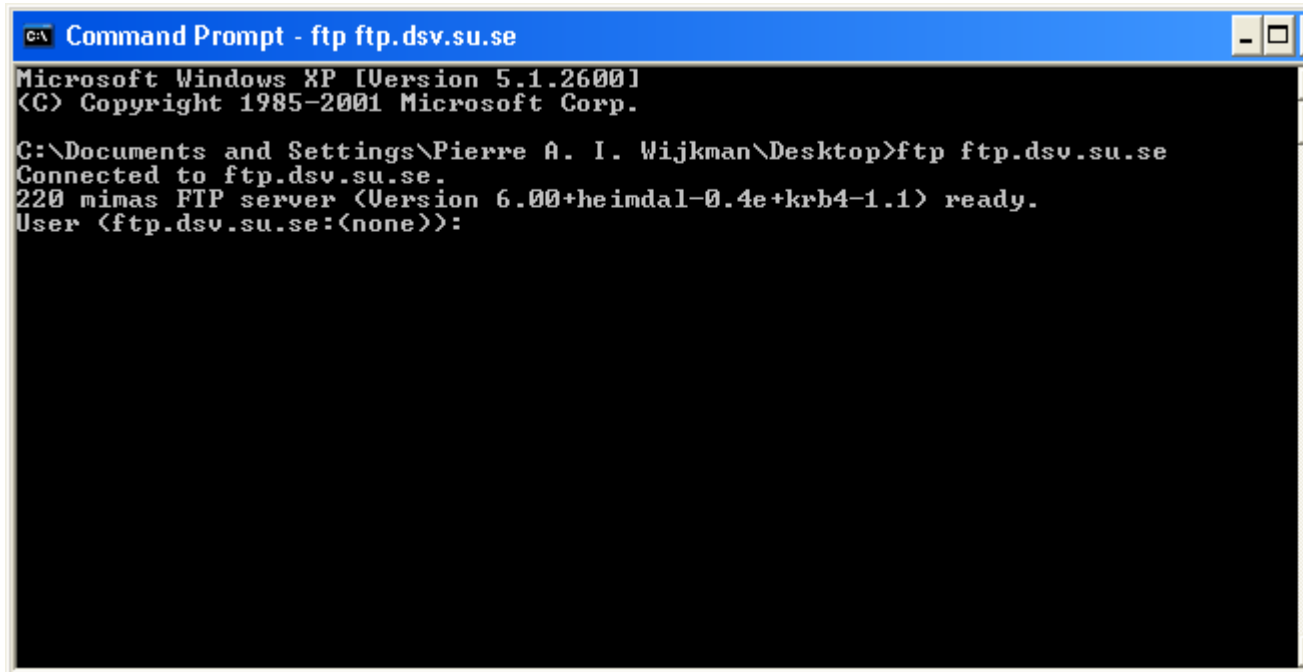
Notera att `www`-katalogen inte ska vara med i URL'ens sökväg när man sedan ska hämta filer i en browser:

- En fil `z.html` på DSV/Unix/Linux-kontot under katalogen `www/x/y` når man i en browser med `http://dsv.su.se/~<användarnamn>/x/y/z.html`

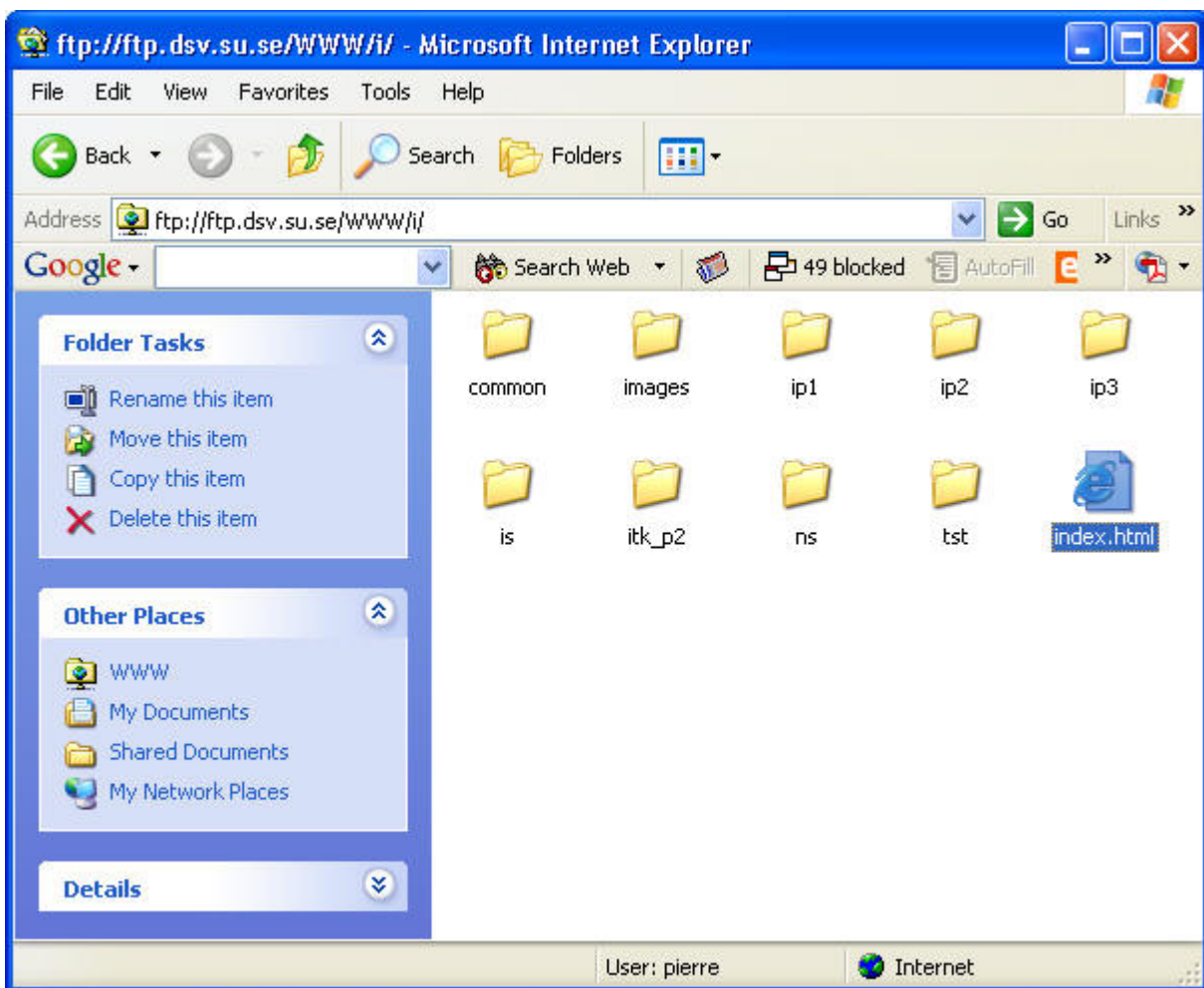
Man kan även nå sina filer på DSV/Win32 men det är endast DSV/Unix/Linux som har en tillförlitlig HTTP-server.

### FTP

I Win32 och Linux så kan man köra en FTP-klient direkt i ett kommando-fönster:



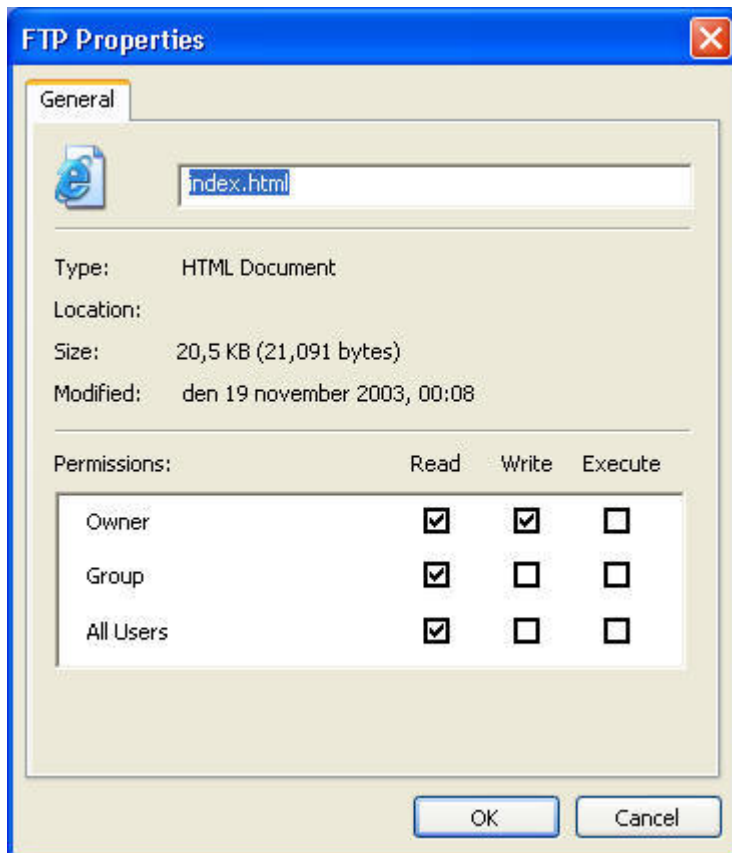
Det är dock ofta lättare att använda en GUI-baserad FTP-klient. Det finns många olika program och »[Shareware.com](http://Shareware.com)« är ett ställe som man kan leta på. Man även använda en vanlig browser, exempelvis Internet Explorer:



Man använder sen drag-and-drop. Man kan även sätta fil- och katalogrättigheter från detta program:

- Högerklicka på fil eller katalog
- Välj Egenskaper
- Sätt rättigheter

Så här kan det se ut:



Notera dock att Internet Explorer inte stöder ASCII-mode.

Man ansluter sig till FTP-servern på följande dator:

- DSV/Unix/Linux: `ftp://ftp.dsv.su.se`

Använder man Internet Explorer så kan man skapa en genväg enligt:

- `ftp://<användarnamn><lösenord>@ftp.dsv.su.se`

Notera att man bör sända:

- Rena text-filer (som CGI-program med mera) i ASCII-mode
- Binära filer (som bilder, kompillerade program med mera) i BINARY-mode

Använder man en brandvägg så bör man se till att initieringen av dataöverföringskanalen sker på klient-sidan och det gör man genom att använda `passive mode` som kan ställas in i de flesta FTP-klienter.

## SFTP och FTPS

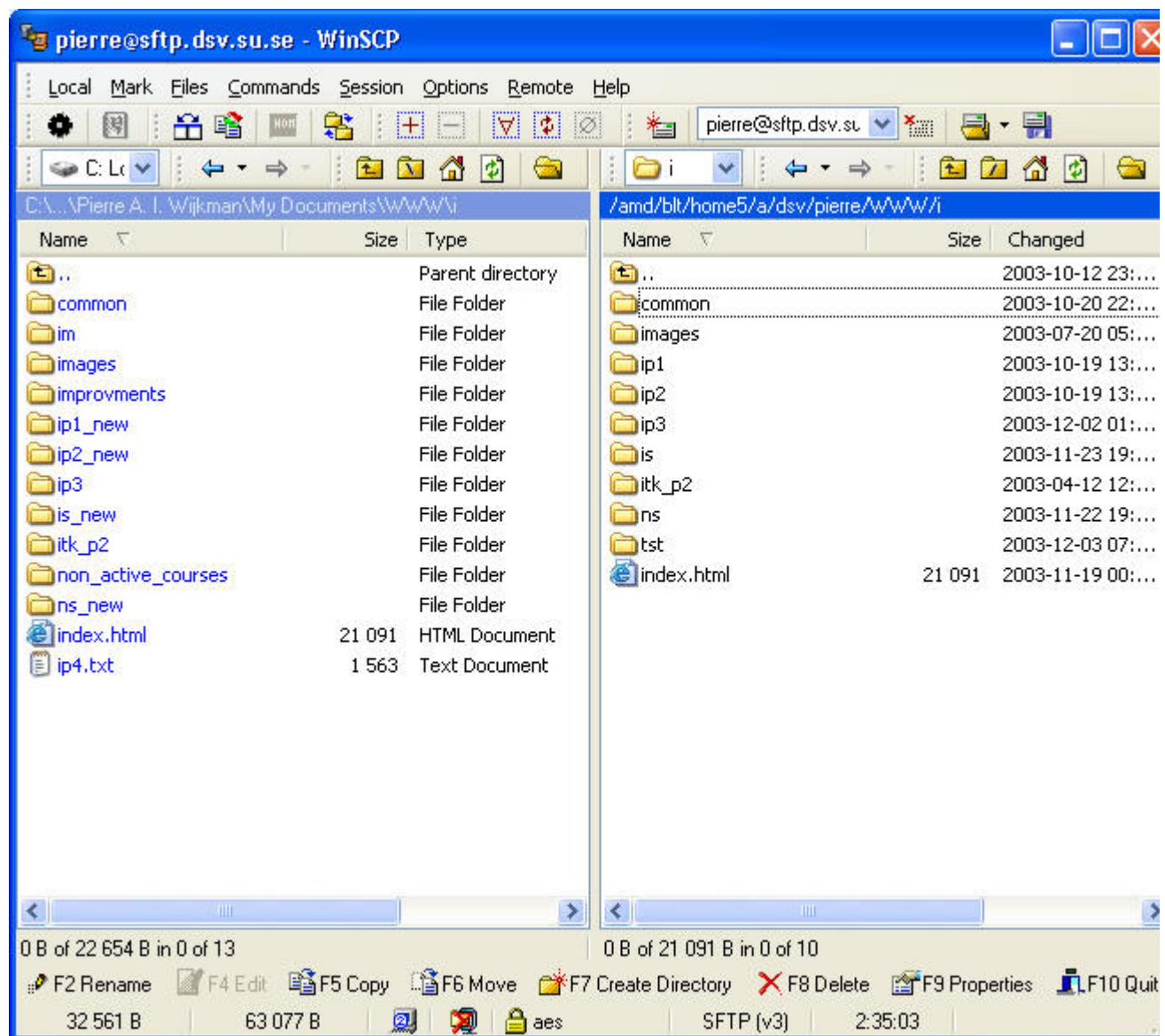
När man använder FTP så sänds all information, även lösenordet, okrypterat över Internet och det kan vara farligt. Man bör istället använda någon av följande tekniker:

- SFTP som använder sig av tunnling via SSH och först upprättar en SSH-anslutning som sedan används för att ansluta till en server via FTP
- FTPS som ansluter som vanlig FTP enligt RFC959 för att sedan begära en säker anslutning via SSL/TLS enligt RFC2228

DSV stöder enbart SFTP så man måste använda ett sådant program och bra program för det är:

- Unix/Linux: sftp som ingår i OpenSSH
- Win32: »[WinSCP](#)«

Exempelvis så ser WinSCP ut så här:



Man ansluter till SFTP-servern på följande datorer:

- DSV/Unix/Linux: sftp.dsv.su.se
- DSV/Win32: wftp.dsv.su.se

Detta säkrare sätt rekommenderas starkt.

## Externa källor

På denna sida finns länkar till externa källor. Tipsa gärna om fler bra länkar.

## Säkerhet

- »[SUN: Security](#)«
- »[SUN: Articles: Security](#)«
- »[SUN: Security Code Samples](#)«
- »[SUN: Java Secure Socket Extension Reference Guide](#)«
- »[JavaWorld: Java Tip 115: Secure JavaMail with JSSE](#)«
- »[New Order: Computer Security and Networking Portal](#)«
- [How We Cracked the Code Book Ciphers](#)

## Lektioner

- »[SUN: Trail: Security in Java 2 SDK 1.2](#)«
- »[SUN: JavaTM Cryptography Extension \(JCE\) Reference Guide](#)«
- »[SUN: JavaTM Secure Socket Extension \(JSSE\) Reference Guide](#)«

## Providers

- »[JSMIME](#)«
- »[Wedgetail: JCSI](#)«
- »[IAIK](#)«
- »[Cryptix](#)«

# Uppgifter

På denna sida finns uppgifterna till kursen. Läs först noga alla [krav](#) och sedan alla [tips](#). Siffror inom () är kapitelhänvisningar.

## Säkra program:

- 1.1 [Policy](#) obligatorisk (1-2)
- 1.2.1 [Signering/verifiering med certifikat: Äpplen](#) obligatorisk (7, 12)
- 1.2.2 [Signering/verifiering med certifikat: Web Start](#) frivillig (7, 12)

## Kryptering/dekryptering:

- 2 [Kryptering/dekryptering](#) obligatorisk (7, 9, 10, 13)

## Signering/verifiering:

- 3.1 [Signering/verifiering: Nyckelpar](#) obligatorisk (7, 9-12)
- 3.2 [Signering/verifiering: Certifikat](#) obligatorisk (7, 9-12)

## Sammanstatta tekniker:

- 4.1.1 [SSL: Högnivåkopplingar](#) obligatorisk (14)
- 4.1.2 [SSL: Lågnivåkopplingar](#) frivillig (14)
- 4.2.1 [S-MIME: Kryptering och/eller signering](#) frivillig
- 4.2.2 [S-MIME: Dekryptering och/eller verifiering](#) frivillig

## Anfall:

- 5.1 [Portscanning](#) frivillig
- 5.2 [Trojanska hästar](#) frivillig
- 5.3 [Virus och maskar](#) frivillig

## Gesällprov

- 6 [Gesällprov](#) obligatorisk

## Krav

På denna sida finns de krav som gäller för uppgifterna.

## Samarbete och betyg

- Alla uppgifter utom gesällprovet ska göras i grupp om 1-2 personer, gesällprovet ska göras ensam
- Vissa av uppgifterna är frivilliga men desto fler av dessa som man gör desto större blir sannolikheten för att få betyget VG/5
- Man kan givetvis ändå få betyget VG/5 om man gör ett väldigt bra gesällprov
- Bästa gesällprov till deadline 1 vinner ett pris

## Kodens utseende

En vanlig orsak till betyget R är att nedanstående krav inte är uppfyllda.

### All kod

- All kod ska vara välstrukturerad
- Man får själv bestämma stil på koden men man måste vara konsekvent
- Använd inte en editor som använder kontrolltecken och tabbar för indentering, koden ska kunna läsas och se bra ut i en vanlig enkel editor som exempelvis Notepad och därmed även i en browser
- Designen på det (eventuella) grafiska användargränssnittet är fri men försök göra det så snyggt som möjligt (man bör alltså inte göra programmen identiska med de ofta lite tråkiga exempelprogram som ges för varje uppgift)

### Java-kod

- All Java-kod ska vara välkommenterad med en kommentar per klass och metod, se vidare »[SUN: Code Conventions for the Java Programming Language](#)«
- Eventuella äpplen som ska redovisas kan vara körbara antingen via appletviewer eller via en plug-in i en vanlig browser
- Använd Swing-GUI-komponenter, inte de nu omoderna AWT-GUI-komponenterna (givetvis är det ok att använda layoutmanagers, händelseklasser med mera från AWT)

## Tips

På denna sida finns tips för uppgifterna.

# Ordning och organisation

- Använd en katalog för varje huvuduppgift
- Under hela konstruktionsprocessen:
  - Välstrukturerad kod
  - Bra namn på klasser, variabler, konstanter

## Evolutionär utveckling

Många moderna programmeringsspråk består av:

- En liten språkkärna
- Ett mycket stort bibliotek av färdigkonstruerade klasser

Den självklara fördelen är att man slipper konstruera en stor mängd saker som annars skulle ta mycket lång tid att göra men nackdelen är att det kan vara svårt att veta och hitta vad som finns samt att man inte får samma grundläggande förståelse.

**Matematik är ingenting man förstår, matematik är något som man vänjer sig vid**

von Neumann

Till att börja med bör man skaffa sig en övergripande förståelse för det område man ska ge sig in på. Detta görs bäst med hjälp av böcker, tidningar och/eller de tutorials och artiklar som finns på Internet. När man har skaffat sig en denna övergripande förståelse så bör man så fort som möjligt testa några av de exempel man stött på. Det är ofta först när man praktiskt tillämpar ett område som man lär sig, eller vänjer sig vid, området och därmed får en förankring eller förståelse av beteendet hos objekten man laborerar med.

Man bör inte skriva av koden "rakt av" utan man bör istället:

- Konstruera en minimal körbar kärna (med sin egen stil på koden)

Dett gör man för varje nytt område man ger sig in på. Man kan sedan:

- Förändra denna minimala körbara kärna mot ett önskbart mål med **små förändringar** av koden

Detta är mycket viktigt! Man går mot ett önskat mål med:

- Små förändringar av koden
- Kompilering
- Testkörning

Man släpper aldrig kontakten med det fungerande programmet.

**We are like sailors who on the open sea must reconstruct their ship but are never able to start afresh from the bottom. Where a beam is taken away a new one must at once be put there, and for this the rest of the ship is used as support. In this way, by using the old beams and driftwood the ship can be shaped entirely anew, but only by gradual reconstruction.**

Otto Neurath

Detta är en mycket viktig tumregel. Försöker man förändra koden i stora steg så blir det fel och dessa fel är ofta svåra att lokalisera och rätta till. Ofta samverkar flera fel och detta gör det givetvis

ännu svårare.

Ofta är stora program konstuerade av många lager/nivåer där varje nivå består av program konstruerade enligt ovanstående enkla principer. Man använder sedan program på en nivå som enheter för konstruktion av program på nästa nivå, och även denna process följer de ovan anvisade principerna.

## Övrigt

- Detta är inte övningsuppgifter och man måste läsa kursboken och viss information som finns på Internet för att klara uppgifterna
- Alla program är testade i DSVs miljö för både Win32 och (det överlägsna :- ) Unix/Linux
- Alla program går att utveckla lokalt ej uppkopplad (på Win32 och Unix/Linux) men man måste ha TCP/IP installerat på sin dator (vilket man har om man exempelvis surfar på Internet)
- Man kan använda IP-adressen 127.0.0.1 eller localhost som motsvarar den egna maskinen

# Säkra program: Policy

## Allmänt

Äpplen körs, till skillnad mot fristående applikationer, i en browser vilket av säkerhetsskäl medför att äpplen default är begränsade i vad de får göra, de körs i en så kallad sandlåda. Vi ska nu se hur man som användare själv kan bestämma vilka säkerhetsrestriktioner som ett äpple ska ha. Notera att fristående program har, tvärt emot äpplen, inga restriktioner alls default men att man kan använda exakt samma teknik på fristående program om man vill ge dem speciella restriktioner.

Uppgiften går ut på att skapa en speciell policy som låter ett äpplet göra vissa saker som det default inte tillåts göra i sandlådan.

## Uppgift

Skapa en policy-fil som ger ett program som startas med denna policy följande tillåtelse:

- Program som kommer från en viss plats, det vill säga en viss codeBase, får koppla upp sig till valfri adress på Internet

Kör programmet från uppgiften [Högnivåkopplingar: Tolkad information](#) (se nedan för koden) på kursen Internetprogrammering I med denna policy-fil och kontrollera att äpplet nu får koppla sig fritt var som helst på Internet.

## Exempel

Ett exempel kan köras enligt ett av följande:

- Starta äpplet lokalt via den egna datorn:
  - Hämta filen: [policy\\_1](#)
  - Hämta filen: [BrowserCool.jar](#)
  - Hämta filen: [browser\\_cool.html](#)
  - Kör med: `appletviewer -J-Djava.security.policy=policy_1 browser_cool.html`
- Starta äpplet via Internet:
  - Hämta filen: [policy\\_2](#)
  - Kör med: `appletviewer -J-Djava.security.policy=policy_2 http://atlas.dsv.su.se/~pierre/i/05_ass/is/a_1_1/browser_cool.html`

- Starta applet via Internet och hämta även policy-filen via Internet:
  - Kör med: `appletviewer -J-Djava.security.policy=http://atlas.dsv.su.se/~pierre/i/05_ass/is/a_1_1/policy_2 http://atlas.dsv.su.se/~pierre/i/05_ass/is/a_1_1/browser_cool.html`

Notera att policy-filerna är olika beroende på om man startar applet lokalt eller via Internet. Det sista sättet att starta applet är inte realistiskt om policy-filen inte är ens egen eftersom man då förlorar kontrollen på vad applet får göra.

## Tips

Kontrollera först att applet fungerar med `appletviewer` utan att använda policy-fil. Om Du tidigare gjort ett HTML-dokument som enbart fungerar för en vanlig browser så gör ett som fungerar för `appletviewer`.

Skapa policy-filen med det i JDK inbyggda verktyget `policytool`:

```
policytool
```

Detta startar ett program med grafiskt användargränssnitt. Om man redan har en policy-fil, säg `policy_1`, och vill editera denna så kan man göra det med kommandot:

```
policytool -file policy_1
```

Tänk på:

- CodeBase ska vara en URL:
  - Det ska vara / även under Win32
  - På DSV har det varit problem med ~ tecknet då man har fått " tecknet istället och en lösning är då att öppna policy-filen i en text-editor och ordna detta med den
- Permissions kan vara svår att fylla i så här kommer lite hjälp:
  - `SocketPermission = java.net.SocketPermission`
  - `TargetName = *`
  - `Actions = accept, connect, listen, resolve`

# Säkra program: Signering / verifiering med certifikat: Äpplen

## Allmänt

I den här uppgiften så bygger vi vidare på uppgiften [Säkra program: Policy](#) och vi ska nu avgöra om vi får ett program som kommer från både (1) rätt ställe och dessutom (2) rätt avsändare. Detta kan vi göra eftersom vi kommer att kräva att programmet är signerat av en viss specifik avsändare. Vi kommer då även in på hur man skapar och lagrar nycklar och certifikat med de verktyg för säkerhet som ingår i JDK: `keytool`, `jarsigner` och `policytool`.

## Uppgift

Arbeta i två kataloger, en tänkt att vara för den som äger koden, den andra tänkt för den som mottager koden:

- I ägar-katalogen: Skapa nycklar (en publik och en privat) med hjälp av keytool-verktyget och använd den privata nyckeln för att signera JAR-filen från uppgiften [Högnivåkopplingar: Tolkad information](#) från kursen Internetprogrammering I med jarsigner-verktyget. Skapa sedan ett certifikat för den publika nyckeln. Kopiera certifikatet till mottagar-katalogen.
- I mottagar-katalogen: Importera certifikatet (innehållande den publika nyckeln) med keytool-verktyget i mottagarens (eventuellt ej ännu skapade) keystore. Gör en policy-fil som specificerar (1) keystore och (2) vem som har signerat filen. Kör den signerade JAR-filen från ägar-katalogen med denna nya policy-fil och keystore och kontrollera att det signerade applet verifieras och nu får koppla sig fritt.

Lägg ägar-katalogen på Internet och anropa programmet lokalt från mottagar-katalogen.

## Exempel

Ett exempel kan köras enligt ett av följande sätt:

- Starta applet lokalt via den egna datorn:
  - Hämta filen: [keystore](#)
  - Hämta filen: [policy\\_1](#)
  - Hämta filen: [SignedBrowserCool.jar](#)
  - Hämta filen: [signed\\_browser\\_cool.html](#)
  - Kör med: `appletviewer -J-Djava.security.policy=policy_1 signed_browser_cool.html`
- Starta applet via Internet:
  - Hämta filen: [keystore](#)
  - Hämta filen: [policy\\_2](#)
  - Kör med: `appletviewer -J-Djava.security.policy=policy_2 http://atlas.dsv.su.se/~pierre/i/05_ass/is/a_1_2_1/sender/signed_browser_cool.html`
- Starta applet via Internet och hämta även policy- och keystore-filerna via Internet:
  - Kör med: `appletviewer -J-Djava.security.policy=http://atlas.dsv.su.se/~pierre/i/05_ass/is/a_1_2_1/sender/policy_3 http://atlas.dsv.su.se/~pierre/i/05_ass/is/a_1_2_1/sender/signed_browser_cool.html`

Notera att policy-filerna är olika beroende på om man startar applet lokalt eller via Internet. Det sista sättet att starta applet är inte realistiskt eftersom policy-filen (här [policy\\_3](#)) inte är ens egen och man då förlorar all kontroll på vad applet får göra.

## Tips

Kopiera policy-filen från uppgiften [Säkra program: Policy](#) och utgå från den.

Keystore i policy-filen ska vara en URL till aktuellt keystore.

Om man behöver testa så kan man verifiera en signerad fil med kommandot:

```
jarsigner -verify -verbose -keystore <keystore> <signed file>
```

Vill man även (frivilligt) titta på hur man gör i en vanlig browser med äpplen som ska kunna göra saker utanför den säkra sandlådan så gör man på ett liknande sätt, se exempelvis följande [exempel](#). Som man ser i exemplet så litar inte browsern på certifikatet eftersom det är självsignerat. Vill man använda denna teknik på riktigt så är det givetvis viktigt med tillförlitlighet och man ska då använda ett certifikat som man kan köpa från någon av de CA som browsern litar på default, exempelvis »[VeriSign: VeriSign Services for Developers](#)« eller »[Thawte: Code Signing Certificates](#)« (som kostar pengar).

## Säkra program:

# Signering / verifiering med certifikat: Web Start

## Allmänt

I den här uppgiften så ska vi titta på hur man genom signering och JNLP kan starta fristående program via en browser. Vi ska nu använda de i JDK inbyggda verktygen för säkerhet: `keytool` och `jarsigner`.

## Uppgift

Använd det fristående programmet från uppgiften [Högnivåkopplingar: Otolkad information](#) från kursen Internetprogrammering I och gör så att man kan starta det med Web Start.

## Exempel

Ett [enkelt exempel](#).

## Tips

Som man ser i exemplet så litar inte browsern på certifikatet eftersom det är självsignerat. Vill man använda denna teknik på riktigt så är det givetvis viktigt med tillförlitlighet och man ska då använda ett certifikat som man kan köpa från någon av de CA som browsern litar på default, exempelvis »[VeriSign: VeriSign Services for Developers](#)« eller »[Thawte: Code Signing Certificates](#)« (som kostar pengar).

Vi har nu tittat på hur användare kan lita på ett visst program. Om man är intresserad av det motsatta, hur ett program kan lita på användare, så kan man antingen programmera detta själv (ganska lätt) eller titta på »[SUN: Java Authentication and Authorization Service \(JAAS\)](#)« (ganska svårt).

# Kryptering / dekryptering

## Allmänt

Vi ska nu titta på kryptering/dekryptering med symmetriska algoritmer. På grund av att USA tidigare (innan version 1.4 av JDK) hade större restriktioner för export av stark kryptering så ingick inte detta i JDK utan man fick hämta paketet Java Cryptography Extension (JCE) separat från länder utanför USA. Detta är anledningen till att det nu finns två paket i JDK som har med säkerhet att göra:

- `java.security`
- `javax.crypto`

I den här uppgiften så ska vi kryptera och dekryptera en godtycklig fil med en symmetrisk algoritm. Detta ska vi göra utan att använda de verktyg för säkerhet som ingår i JDK, vi ska programmera utifrån de API'er som finns i JDK.

## Uppgift

Skapa följande 3 fristående program som vid kommando-prompten startas med:

- java KeyHandler <secretKey> som skapar en (hemlig) nyckel och sparar denna nyckel (okrypterad) i filen <secretKey>
- java EncryptHandler <data> <secretKey> <encryptedData> som krypterar filen <data> med (den hemliga) nyckeln i filen <privateKey> och sparar det som krypterats i filen <encryptedData>
- java DecryptHandler <encryptedData> <secretKey> <decryptetData> som dekrypterar filen <encryptedData> med (den hemliga) nyckeln i filen <secretKey> och sparar det som dekrypterats i filen <decryptetData>

Systemet ska göras utan användning av de verktyg för säkerhet som ingår i JDK.

Kryptering/dekrypterings-algoritmen som ska användas ska vara så stark som möjligt och välj mellan:

- Blowfish med 448 bits nyckel (om man använder en dator på DSV som inte har stöd för obegränsad kryptering/dekryptering så räcker det med en kortare nyckel, exempelvis en 32 bits nyckel)
- AES (kräver JDK 1.4.2 eller senare)

Man behöver inte använda klassen KeyStore utan kan spara nycklarna okrypterade direkt på fil.

## Exempel

Ett enkelt exempel kan köras enligt följande:

1. Hämta filen: [KeyHandler.class](#)
2. Kör programmet och gör nyckel: java KeyHandler <secretKey>
3. Hämta filen: [EncryptHandler.class](#)
4. Hämta filen: [data](#)
5. Kör programmet och kryptera: java EncryptHandler data <secretKey> <encryptedData>
6. Hämta filen: [DecryptHandler.class](#)
7. Kör programmet och dekryptera: java DecryptHandler <encryptedData> <secretKey> <decryptetData>

Detta exempel använder enbart en 32-bits nyckel så att man även kan testa på datorer utan stöd för obegränsad kryptering/dekryptering.

## Tips

Man bör importera följande för säkerhetsfunktionerna:

```
import javax.crypto.*;
import javax.crypto.spec.*;
import java.security.*;
```

Använd Objekt-strömmar vid fil-hanteringen av nycklar.

Om man valt Blowfish så skriv String algorithm = "Blowfish/ECB/PKCS5Padding" i både EncryptHandler och DecryptHandler, se »[SUN: JavaTM Cryptography Extension \(JCE\) Reference Guide: Appendix A: Standard Names](#)« för mer om detta.

# Signering / verifiering: Nyckelpar

## Allmänt

I den här uppgiften så ska vi signera och verifiera en godtycklig fil. Detta ska vi göra utan att använda de verktyg för säkerhet som ingår i JDK, vi ska alltså programmera utifrån de API'er som finns i JDK.

## Uppgift

Skapa följande 3 fristående program som vid kommando-prompten startas med:

- `java KeyHandler <privateKey> <publicKey>` som skapar ett privat/publikt nyckel-par och sparar dessa nycklar (okrypterade) till de två filerna `<privateKey>` och `<publicKey>`
- `java SignHandler <data> <privateKey> <signature>` som signerar filen `<data>` med (den privata) nyckeln `<privateKey>` och sparar signaturen i `<signature>`
- `java VerifyHandler <data> <publicKey> <signature>` som kontrollerar om (den publika) nyckeln i filen `<publicKey>` och signaturen i filen `<signature>` verifierar filen `<data>`

Vi använder alltså inte certifikat i denna uppgift utan laborerar direkt med nycklarna (okrypterade). Testa verifieringen både med en oförändrad och en förändrad data-fil. Systemet ska göras utan användning av de verktyg för säkerhet som ingår i JDK. Signerings/verifierings-algoritmerna som ska användas är:

- SHA-1 för message-digest
- DSA (med 1024 bits nycklar) för signaturen

Man behöver inte använda klassen `keystore` utan kan spara nycklarna okrypterade direkt på fil.

## Exempel

Ett enkelt exempel kan köras enligt följande:

1. Hämta filen: [KeyHandler.class](#)
2. Kör programmet och gör nycklar: `java KeyHandler <privateKey> <publicKey>`
3. Hämta filen: [SignHandler.class](#)
4. Hämta filen: [data](#)
5. Kör programmet och signera: `java SignHandler data <privateKey> <signature>`
6. Hämta filen: [VerifyHandler.class](#)
7. Kör programmet och verifiera: `java VerifyHandler data <publicKey> <signature>`

## Tips

Man bör importera följande för säkerhetsfunktionerna:

```
import java.security.*;
```

Använd objekt-strömmar vid fil-hanteringen av nycklar.

Använd de högnivå funktioner som finns tillgängliga, exempelvis i klassen `Signature`. Vill man hellre gå in på alla detaljer så bör man först bekanta sig med uppgiften [Kryptering/dekryptering](#) och sedan:

- Signering:
  - Skapa ett message-digest på filen
  - Använd den privata nyckeln för att signera detta message digest
  - Det som sen ska sändas är:
    - Data-filen
    - Den publika nyckeln (om inte mottagaren redan har den)
    - Signaturen

- Verifiering:
  - Mottagaren får:
    - Data-filen
    - Den publika nyckeln (om inte mottagaren redan har den)
    - Signaturen
  - Använd den publika nyckeln på signaturen för att få fram ett message digest 1
  - Skapa ett message-digest 2 på text-meddelandet
  - Kontrollera om message digest 1 = message digest 2, och om dessa är samma är meddelandet verifierat

# Signering / verifiering: Certifikat

## Allmänt

I den här uppgiften så ska vi, precis som i uppgiften [Signering/verifiering: Nyckelpar](#), signera och verifiera en godtycklig fil. Denna gång ska vi delvis använda de verktyg för säkerhet som finns i JDK och resten ska vi ska programmera utifrån de API'er som finns i JDK.

## Uppgift

Skapa först ett nyckelpar med verktyget keytool. Detta nyckelpar lagras då i en keystore.

Skapa sedan följande 2 fristående program som vid kommando-prompten startas med:

- `java SignHandler <keystore> <keystore_password> <alias> <key_password> <data> <certificate> <signature>` som signerar filen <data> med (den privata) nyckeln som finns i <keystore> under namnet <alias>, sparar ett certifikat (innehållande den publika nyckeln) i filen <certificate> och sparar signaturen i filen <signature> (sändaren kan sedan skicka filerna <data>, <signature> och <certificate> till mottagaren för verifiering)
- `java VerifyHandler <data> <certificate> <signature>` som kontrollerar om certifikatet (med den publika nyckeln) i filen <certificate> och signaturen i filen <signature> verifierar filen <data>

Testa verifieringen både med en oförändrad och en förändrad data-fil. Allt utom konstruktionen av nyckelparet ska göras utan användning av de verktyg för säkerhet som ingår i JDK. Algoritmerna som ska användas är samma som för uppgiften [Signering/verifiering: Nyckelpar](#).

## Exempel

Ett enkelt exempel kan köras enligt följande:

1. Hämta filen: [SignHandler.class](#)
2. Hämta filen: [keystore](#)
3. Hämta filen: [data](#)
4. Kör programmet och signera: `java SignHandler keystore abcdef pierre fedcba data <certificate> <signature>`
5. Hämta filen: [VerifyHandler.class](#)
6. Kör programmet och verifiera: `java VerifyHandler data <certificate> <signature>`

## Tips

Se tips för uppgiften [Signering/verifiering: Nyckelpar](#).

# Sammansatta tekniker: SSL: Högnivåkopplingar

## Allmänt

Man kan givetvis skapa system som kombinerar teknikerna från de två tidigare uppgifterna:

- Signering/verifiering
- Kryptering/dekryptering

Schneier (som gjort Blowfish) rekommenderar att man först signerar och sedan krypterar eftersom det är lättare att med lagens stöd hävda att någon signerat ett "riktigt" dokument. Vi ska nu bekanta oss med två av de standarder som finns för detta:

- Secure Socket Layer (SSL)
- Secure Multipurpose Internet Mail Extensions (S/MIME)

Vi börjar med SSL som är den standard som används vid säkra överföringar över webben av banker, e-affärer med mera. Uppgiften visar hur man använder SSL med högnivåkopplingar.

## Uppgift

Modifiera koden för browsern i uppgiften [Högnivåkopplingar: Otolkad information](#) från kursen Internetprogrammering I enligt följande:

- Browsern ska fungera med SSL och när man kopplar sig via HTTPS-protokollet så ska programmet fungera för både:
  - URL med namn, exempelvis `https://www.verisign.com`
  - URL med IP-nummer, exempelvis `https://65.205.249.60/`

Vill man (frivilligt) även använda autentisering av klient-sidan så får man lägga till ytterligare en rad kod samt göra certifikat. För att testa detta krävs dock att man besöker en webbplats som kräver att klienten ska autentisera sig.

## Exempel

Ett enkelt exempel kan köras enligt följande:

1. Hämta filen: [Browser.jar](#)
2. Kör programmet: `java -jar Browser.jar`

## Tips

Inga tips ges.

# Sammansatta tekniker: SSL: Lågnivåkopplingar

# Allmänt

Uppgiften visar hur man använder SSL med lågnivåkopplingar.

## Uppgift

Modifiera koden för chat-systemet i uppgifterna:

- [Lågnivåkopplingar: Stream sockets: Klient-sidan](#)
- [Lågnivåkopplingar: Stream sockets: Server-sidan](#)

från kursen Internetprogrammering I så att all kommunikation sker med SSL. Man behöver inte använda autentisering på klient-sidan.

## Exempel

Ett enkelt exempel kan köras enligt följande:

1. Hämta filen: [Server.jar](#)
2. Kör programmet med: `java -jar Server.jar`
3. Hämta filen: [Client.jar](#)
4. Kör programmet med: `java -jar Client.jar`

## Tips

På server-sidan anger man att klienten inte behöver autentiseras med:

```
serverSocket.setNeedClientAuth(false);
```

Lägg till följande cipherSuite `SSL_DH_anon_EXPORT_WITH_DES40_CBC_SHA` till både `SSLServerSocket` och `SSLSocket`.

# Sammanstatta tekniker: S/MIME: Signering och / eller kryptering

## Allmänt

Vi fortsätter med S/MIME som är den standard som används vid säkra överföringar av e-post över Internet av banker, e-affärer med mera. Uppgiften visar hur man S/MIME signerar och S/MIME krypterar e-post. Detta ska vi delvis göra med de verktyg för säkerhet som finns i JDK och resten ska vi programmera utifrån de API'er som finns i JDK och en extension från en annan leverantör som implementerar S/MIME.

## Uppgift

Modifiera koden för e-post-programmet i uppgiften [E-post: Sändning](#) från kursen Internetprogrammering I enligt följande:

- Skapa först ett nyckelpar med verktyget `keytool` och använd nyckelalgoritmen för RSA
- Skapa sedan ett fristående program som vid kommando-promten startas med `java`

SMIMESender <host> <from> <subject> <to> <message> som sänder ett meddelande till mail-servern <host> från <from> med ärende <subject> till <to> med meddelandet <message> som är S/MIME signerat och krypterat med nyckelparet

Allt utom konstruktionen av nyckelparet ska göras utan användning av de verktyg för säkerhet som ingår i JDK.

Om man vill så kan man även implementera så att man kan använda riktiga certifikat och hemliga nycklar från exempelvis »[Thawte](#)«. Det blir ju roligare eftersom programmet då fungerar med alla vanliga e-post-klienter som Mozilla Thunderbird, Eudora, Outlook, Outlook Express med flera. Notera att man då inte kan använda SUN's keystore-format JKS utan måste byta till PKCS12.

## Exempel

Inget exempel ges, den synliga funktionaliteten är mycket lik uppgiften [E-post: Sändning](#) på kursen Internetprogrammering I.

## Tips

Tyvärr så har inte SUN implementerat detta ännu men det finns andra providers som har gjort det, se »[SUN: Java Cryptography Extension \(JCE\) 1.2.2](#)«. Eftersom det ännu inte finns någon standardimplementation så får man vara beredd på att olika providers använder olika namn på klasser, interface med mera. Bra information har »[Wedgetail Communications: JCSI](#)«, exempelvis deras utmärkta exempel »[Wedgetail Communications: JCSI Examples](#)«. Väljer man denna provider så måste man använda följande JAR-filer:

- [activation.jar](#)
- [mail.jar](#)
- [jcsi\\_base.jar](#)
- [jcsi\\_jce.jar](#)
- [jcsi\\_provider.jar](#)
- [jcsi\\_smime.jar](#)

Dessa JAR-filer måste man sätta classpath till när man kompilerar och kör:

- Unix: `javac -classpath activation.jar:mail.jar:jcsi_base.jar:jcsi_jce.jar:jcsi_provider.jar:jcsi_smime.jar:. SMIMESender.java`
- Win32: `javac -classpath activation.jar;mail.jar;jcsi_base.jar;jcsi_jce.jar;jcsi_provider.jar;jcsi_smime.jar;. SMIMESender.java`

# Sammanfattning tekniker: S/MIME: Verifiering och / eller dekryptering

## Allmänt

I den här uppgiften så ska vi hämta ett e-post meddelande och S/MIME verifiera och S/MIME dekryptera detta. Denna gång ska vi delvis använda de verktyg för säkerhet som finns i JDK och resten ska vi programmera utifrån de API'er som finns i JDK och en extension från en annan leverantör som implementerar S/MIME.

## Uppgift

Modifiera koden för e-post-programmet i uppgiften [E-post: Mottagning](#) från kursen Internetprogrammering I enligt följande:

- Utgå från den keystore som skapades i uppgiften [Sammansatta tekniker: S/MIME: Signering och/eller kryptering](#)
- Skapa sedan ett fristående program som vid kommando-promten startas med java SMIMEReceiver <host> <användarnamn> <lösenord> som hämtar ett e-post meddelande från mail-servern <host> hos användare <användarnamn> med lösenord <lösenord> och S/MIME verifierar och dekrypterar detta med hjälp av nyckelparet i keystore

För att undvika komplexiteten med hanterandet av många olika certifikat och verifiering av dessa mot en riktig CA så tittar vi enbart på hur man verifierar och dekrypterar meddelanden sända till sig själv.

Allt ska göras utan användning av de verktyg för säkerhet som ingår i JDK.

## Exempel

Inget exempel ges, den synliga funktionaliteten är mycket lik uppgiften [E-post: Mottagning](#) på kursen Internetprogrammering I.

## Tips

Se tips för uppgiften [Sammansatta tekniker: S/MIME: Signering och/eller kryptering](#).

# Anfall: Portscanning

## Allmänt

Vi går nu in på några uppgifter som är av en annan karaktär, anfall. Vi gör det för att lära oss lite om detta så vi blir bättre på att skapa försvar mot liknande typer av program.

I den här uppgiften så ska vi titta på portscanning som används för att hitta öppna portar på datorer. När man väl har hittat en öppen port så kan en anfallare gå vidare och undersöka vilken tjänst som körs där och sen anfalla den. Ibland körs tjänster som är osäkra som man kan anfalla vidare men vi går inte vidare på hur man gör för att utnyttja det eftersom det hela tiden är ett rörligt specifikt mål.

Observera! Använd inte era program för att göra olagliga saker.

## Uppgift

Skapa ett fristående program som vid kommando-promten startas med java PortScan <host-start> <host-stop> <port-start> <port-stop> <result file> som scannar av alla (stream sockets) IP-adresser i intervallet <host-start> till <host-stop> för alla portar i intervallet <port-start> till <port-stop> och skriver resultatet till filen <result file>.

Frivillig utvidgning är att till varje port som är en standarport för någon tjänst även ta med tjänstens namn i filen <result file>.

Observera! En del datoransvariga anser att portskanning är dataintrång och anmäler sådana till den ISP varifrån man kört skanningen. Testa enbart programmet mot någon av datorerna:

- `www.dsv.su.se` = 130.237.161.25
- `atlas.dsv.su.se` = 130.237.161.23
- `luna.dsv.su.se` = 130.237.161.9
- `unni.dsv.su.se` = 120.237.161.27
- `vall.dsv.su.se` = 130.237.161.5

Man ska inte använda det mot anställdas datorer på SU/KTH heller eftersom IRT-teamet på SU/KTH troligtvis kommer att se att portskanning pågår och misstänka att crackers är i farten.

## Exempel

Inget exempel ges.

## Tips

Inga tips ges.

# Anfall: Trojanska hästar

## Allmänt

I den här uppgiften så ska vi titta på trojanska hästar som används för att få tillgång till resurser på en annan dator.

Observera! Använd inte era program för att göra olagliga saker.

## Uppgift

Skapa ett fristående program som vid kommando-promten startas med `java Trojan <port> <password> <email>` som startar en begränsad filserver på port <port> som enbart klienter med rätt lösenord <password> har access till och som varje gång det startar sänder den egna host-adressen till e-post-adress <email>. Den begränsade filservern ska enbart lista kataloger och filer, man ska alltså inte kunna hämta eller lägga till något.

Några frivilliga utvidgningar:

- För att installera programmet på en dator kan man tänka sig att det sänds som en bifogad fil via e-post som man (på något sätt) får mottagaren att starta och för att programmet ska fungera på datorer som inte har JRE installerat så kan man kompilera koden till native för vissa specifika operativsystem (exempelvis de mest använda), se:
  - »[GCJ: The GNU Compiler for the Javatm Programming Language](#)«
  - »[Excelsior JET](#)«
  - »[IBM: Create native, cross-platform GUI applications](#)«
  - »[IBM: Weighing in on Java native compilation](#)«
- För att programmet ska starta automatiskt varje gång en användare startar sin dator så kan programmet själv kopiera sig till en speciell katalog (olika beroende på operativsystem) eller ändra i vissa filer som körs vid uppstart (olika beroende på operativsystem) och man kan då döpa filen till något lämpligt namn (som inte innehåller ordet "trojan" :-)
- För att programmet inte ska synas vid körning kan man ... ja, tipsa gärna här

## Exempel

Inget exempel ges.

## Tips

Inga tips ges.

# Anfall: Virus och maskar

## Allmänt

I den här uppgiften så ska vi titta på virus och maskar som används för att sprida ett program till många datorer.

Observera! Använd inte era program för att göra olagliga saker.

## Uppgift

Skapa ett fristående program som vid kommando-promten startas med `java VirusWorm <email>` som kontinuerligt letar efter (nya) e-post-adresser i filer hos användaren och som sänder ett e-brev till `<email>` med en kopia av programmet själv och alla hittade e-post-adresser i meddelandet.

Den specificerade e-postadressen `<email>` måste vara din egen. Ett riktigt virus sänder ett e-brev till alla hittade e-post-adresser men det ska man absolut inte implementera.

Frivilliga utvidgningar är samma som för uppgiften [Anfall: Trojanska hästar](#).

## Exempel

Inget exempel ges.

## Tips

Inga tips ges.

# Gesällprov

## Allmänt

Gesällprovet är en uppgift för den egna kreativiteten och är ett fritt valt (men obligatoriskt) arbete där du får chansen att visa vad du går för konstruktionsmässigt (istället för minnesmässigt som på en tenta) i "lugn och ro".

## Uppgift

Följande krav måste uppfyllas:

- Ska baseras på de tekniker som kursen bygger på men får givetvis även innehålla tekniker från andra områden
- Ska vara nätverksbaserat
- Ska vara bra både till både form och funktion
- Ska inte vara en av de tidigare uppgifterna på kursen men gärna en utökning av en eller flera

av dessa

## Tips

Huvudsaken är att man gör något som är kul! Några exempel:

- E-post system som använder stark kryptering
- Chat-system som använder stark signering/verifiering och eller kryptering/dekryptering med hjälp av klasserna `SignedObject` och `SealedObject`
- WebScraping, det vill säga ett program som simulerar en browser på en certifierad och krypterad förbindelse som fungerar mot exempelvis hotmail eller mot någon annan tjänst som normalt bara går att nå via webben

## Hjälp

På denna sida finns alla hjälpresurser på kursen.

## Handledning

Kör man fast så får man hjälp via news och e-post. Notera dock att det inte är någon handledning på:

- Kunskap som ingår i förkunskapskraven (kompileringsfel med mera)
- Gesällprovet

News är tänkt som en plattform för diskussion mellan studenterna men handledarna kommer att besvara de frågor som fortfarande är obesvarade senare under de dagar som det är handledning. E-post går direkt till handledarna och de kommer att försöka svara snabbt men eftersom de ibland får väldigt många frågor kan svaren dröja. Notera att handledarna enbart arbetar inom angivna handledningstider vilket medför att man kanske inte hinner få svar samma dag på en sent ställd fråga.

## News

News bör användas för frågor som kan intressera andra:

- Läs först igenom de tidigare frågorna och svaren
- Hittar man inte en lösning så:
  - Ange ett väl beskrivande ämne för brevet
  - Var mycket tydlig vid beskrivningen av problemet
  - Sänd inte in mycket kod eller fullständiga program

News-server:

- Adress: `atlas.dsv.su.se`
- Port: 1199 (notera att detta inte är default numret och därför måste sättas manuellt)

News-grupper:

- `public.is.theory` för frågor om teoridelen
- `public.is.a.1` för frågor om uppgift 1
- `public.is.a.2` för frågor om uppgift 2
- ... och så vidare ...
- `public.is.misc` för övriga frågor

Se även [news](#) för mer detaljerad information om hur man konfigurerar en news-klient.

## E-post

E-post bör användas för specifika frågor utan intresse för andra:

- Sänd inte e-post direkt till lärarna utan använd länken nedan
- Om man ska sända med kod:
  - Sänd endast med de delar som är relevanta för frågan
  - Koden ska vara välkommenterad och välstrukturerad
- Använd "svara till alla" på eventuella följdfrågor

E-post-server:

- Ingående e-brev (POP3):
  - Adress: `mail.dsv.su.se`
  - Port: 110
- Utgående e-brev (SMTP):
  - Adress: `mail.dsv.su.se`
  - Port: 25

Notera att om man sitter uppkopplad utanför DSV's domän (exempelvis via Telia Bredband) så ska man använda leverantörers SMTP-server för utgående brev.

E-post länk:

Klicka här för handledning via e-post:

- [Teoretisk del](#)
- [Praktisk del](#)

Se även [e-post](#) för mer detaljerad information om hur man konfigurerar en e-post-klient.

## FAQ

Här följer de frågor och svar som brukar vara vanligast:

### Säkerhet

#### Fråga/svar 1

Jag får security-exception när jag försöker koppla upp mig till DSV's hemsida. Jag har inte gjort någon policy-fil men några säger att det ska fungera mot DSV utan policy-fil.

Äpplen får (default) bara koppla sig till samma domän de kommer ifrån. Om man lägger ut äpplet på nätet (under DSVs domän) och kör med appletviewer `http://www.dsv.su.se/~pierre ... osv` så kan äpplet koppla sig överallt inom `www.dsv.su.se` utan policy-fil. Kör man lokalt på den egna maskinen så måste man ha en HTTP-server igång eftersom URL-objekt kopplar sig till just en sådan server.

## E-post

På denna sida beskrivs hur man konfigurerar e-post-klienten. Följande är ett exempel på hur man konfigurerar e-post i Outlook Express, förfarandet är ungefär detsamma i andra moderna e-post/news-klienter:

- Klicka på Verktyg/Konton
- Klicka på Add/Mail
- Skriv in namn och e-post-adress
- Skriv in adressen för inkommande och utgående e-post-servrar
- Skriv in användarnamn och lösenord
- Klicka på Avsluta

Vill man använda exempelvis hotmail istället så kan man göra följande för att få all e-post som kommer till DSV vidareänd till hotmail-kontot:

- telnet triton.dsv.su.se
- emacs .forward
- Skriv in hotmail-adressen, spara och avsluta emacs
- Logga ut

De som använder ett e-postkonto på DSV kan även använda »[DSV: Webmail](#)«.

## News

På denna sida beskrivs hur man konfigurerar en news-klient. Följande är ett exempel på hur man konfigurerar news i Outlook Express, förfarandet är ungefär detsamma i andra moderna e-post/news-klienter:

- Klicka på Verktyg/Konton
- Klicka på Lägg till/News
- Skriv in namn och e-post-adress
- Ange adressen för news-servern
- Klicka på Avsluta
- Klicka på mappen Alla och markera aktuell news-server
- Klicka på Egenskaper
- Klicka på Avancerat
- Ange port nummer till news-servern
- Klicka på ok och sen stäng
- Svara Ja på frågan vill du ladda ner nyhets-grupper från news-kontot
- Dubbelklicka på alla grupper som är av intresse
- Klicka på ok

Default så visar Outlook Express bara de senaste meddelandena. Vill man ändra på det så:

- Klicka på Verktyg/Alternativ
- Klicka på mappen Underhåll
- Ändra vid Ta bort news meddelanden efter 5 dagar (exempelvis genom att ta bort markeringen helt)

Anledningen till att vi använder news istället för FirstClass är att news är en öppen mycket vanlig standard på Internet och att det även ingår som några förslag på gesällprov.

## Redovisning i detalj

Denna beskrivning utgår från att man använder DSV's datorer men man kan använda andra ställen än DSV för redovisningen. Notera dock att andra ställen kan använda en annan huvudkatalog för webpublicering och andra rättigheter på kataloger och filer.

## Steg för steg

1. Starta ett [filöverföringsprogram](#) och koppla dig till ditt DSV/Unix/Linux-konto
2. Skapa katalogstrukturen nedan och ladda upp filerna i denna struktur
3. Sätt rättigheterna på kataloger och filer så att allt blir läsbart från en browser
4. Testa att allt kan nås från en browser och att all källkod ser bra ut från en browser
5. Ange på redovisningssidan:
  - URL'en till denna katalogstruktur: `http://www.dsv.su.se/~x/is/`, där x är användarnamnet
  - Att allt är klart för redovisning

## Katalogstruktur

Man måste ha **exakt följande katalogstruktur** men använder man en annan plats än DSV så måste inte www katalogen ha det namnet.

```
www
 is
 is.zip (is zippad, ej innehållande is.zip givetvis)
 1.1
 TXT-beskrivning av procedur för uppgift 1.1
 1.2.1
 TXT-beskrivning av procedur för uppgift 1.2.1
 1.2.2
 TXT-beskrivning av procedur för uppgift 1.2.2
 2
 källkodsfiler för uppgift 2

 ... osv ...

 6
 källkodsfiler för gesällprovet
 description
 beskrivning i text och bild av gesällprovet
 docs
 javadoc-beskrivning
```

Observera:

- Katalogen www måste ha stora/versala bokstäver
- Skapa inte kataloger för frivilliga uppgifter som inte är gjorda
- Lägg ingen fil med namnet `index.html` eller `index.htm` direkt i uppgiftskatalogerna eftersom man då inte kan lista innehållet i dessa kataloger från Internet

## Beskrivning av gesällprovet

Gesällprovet ska beskrivas generellt med både text och bild (på programmet under körning) på något av följande sätt:

- HTML-dokument med bilderna inkluderade
- PDF-dokument med bilderna inkluderade
- TXT-dokument med bilderna separata

Ett HTML-dokument kan göras antingen för hand i en vanlig text-editor eller med ett högnivåverktyg.

För att skapa PDF-filer i Win32 kan man exempelvis:

1. Lägg till en skrivare, exempelvis HP Color LaserJet PS
2. Högerklicka på skrivaren och välj Egenskaper, klicka på Portar och kryssa i FILE
3. Skriv ut Word-dokumentet till denna skrivare varefter en fil `<namn>.prn` skapas
4. Gå till »[PS2PDF.com](#)« och konvertera filen till PDF

En bild (använd inte BMP-format) av ett program under körning fås enligt:

- Win32: Tryck på <CTRL><ALT><PRINT SCR> samtidigt, vilket medför att bilden på programmet läggs i urklippsbufferten som sedan lätt kan klistras in i ett HTML-dokument i en HTML-editor
- Linux/Unix: Skriv i kommandofönstret `import dump.jpeg` och klicka sen på det fönster som man vill ha en bild på (ändrar man till en annan extension så får man bilden som en annan bildtyp)

All eventuell java-kod i gesällprovet ska beskrivas med verktyget javadoc.

## Efter erhållet betyg

När man fått G/VG eller 3/4/5 på ett arbete så bör man ändra rättigheterna på källkodsfilerna så att ingen annan kan läsa dem.

# Nyheter

På denna sida visas alla uppdateringar.

| Tid              | Rum  | Uppdatering     |
|------------------|------|-----------------|
| 2004-11-22 21:45 | Allt | Nya kursen klar |

Den här sidan fungerar på både stationära och mobila klienter, har hög tillgänglighet för funktionshindrade och använder webbsäkra fonter. Den följer W3C's strikta standard WML 2.0, det vill säga XHTML Basic och CSS Mobile Profile. Den är genererad från text-filer som använder ett egendefinerat minimalt WIKI-baserat språk och en databas. Systemet och allt material är gjort av Pierre A. I. Wijkman, DSV, 2004-2005 och är skyddat av lagen om upphovsrätt.