

OOP Tentamen 2010-01-15 15.00 – 20.00

Avser kurserna: DSK1:OOP
GES:OOP
ID1013

Anvisningar

Skriv namn och personnummer på varje inlämnat blad.
Påbörja varje ny uppgift på nytt blad.
Skriv endast på ena sidan av bladen.
Skriv tydligt - oläsbara svar beaktas ej.

Max antal poäng är 30.

För att bli godkänd på tentan (minst betyg E) krävs dels minst 4 poäng sammanlagt på uppgift 1 och uppgift 2 och dels minst 15 poäng sammanlagt på hela tentan.

För högre betyg krävs:

Betyg D: minst 18 poäng samt högst en uppgift med 0 poäng.
Betyg C: minst 21 poäng samt ingen uppgift med 0 poäng.
Betyg B: minst 24 poäng samt ingen uppgift med 0 poäng.
Betyg A: minst 27 poäng samt uppgifterna lösta med korrekt användande av objektorienterade principer (t.ex. inkapsling, ej upprepning av kod).

Betyget Fx med möjlighet att komplettera ges till studenter som fått 12-14 poäng eller som fått mer än 14 poäng men missat att få 4 poäng på uppgift 1+2.

Hjälpmedel

Samtliga läroböcker om Java är tillåtna hjälpmedel.
Kompendier räknas INTE som läroböcker.

Lycka till!!!!

Lösningsförslag kommer att presenteras på kursens webbsidor.

Uppgift 1 (6 poäng)

Om man exekverar följande programrader, vad kommer att skrivas ut på skärmen (du skall i dina svar vara noga med vad som skrivs på vilken rad, alltså beakta skillnaden mellan print och println).

A

```
int tal=3;
switch (tal--){
case 1: case 2:
    System.out.println(tal++);
case 3:
    System.out.println(tal++);
case 4:
    System.out.println(tal++);
default:
    System.out.println(tal++);
}
```

B

```
for (int x=13; x<20; x+=3){
    int y = x-10;
    do {
        System.out.print(y+" ");
        y += 3;
    } while (y<x);
    System.out.println("Hej"+y);
}
```

C

```
int[]arr = new int[6];
int start=20;
for (int index=0; index<arr.length; index++){
    arr[index]=start-index;
    start-=index;
}
for (int z : arr)
    System.out.println(z);
```

Uppgift 2 (6 poäng)

Om man exekverar (interpreterar) följande Java-program - vad kommer att skrivas ut?
main-metoden ligger i class Uppgift2, startas alltså med: **java Uppgift2**

```
import java.util.ArrayList;
class Uppgift2{
    public static void main(String[]args){
        ArrayList<Gerrf> alla = new ArrayList<Gerrf>();
        alla.add(new Davvi());
        alla.add(new Kyyn("Soffa"));
        alla.add(new Davvi());
        alla.add(new Kyyn("Skrivbord"));
        for (Gerrf ge : alla)
            ge.skriv();
    }
}

class Gerrf{
    private static int tal=0;
    private String grej="Stol";

    public Gerrf(){
        tal++;
    }

    public Gerrf(String str){
        grej=str+tal++;
    }

    public void skriv(){
        for (int x=1; x<tal; x++)
            System.out.print(getGrej());
        System.out.println("Stopp");
    }
    public String getGrej(){
        return grej;
    }
}

class Kyyn extends Gerrf{
    public Kyyn(String str){
        super(str);
    }

    public void skriv(){
        System.out.println(getGrej());
    }
}

class Davvi extends Gerrf{
    public String annat="Bord";

    public String getGrej(){
        return annat;
    }
}
```

Uppgift 3 (6 poäng)

Lotto är ett spel där man gissar/tippar på sju olika nummer mellan 1 och 35. Därefter så slumpas det fram sju nummer mellan 1 och 35 i en dragning. Man jämför sedan sina nummer med den rätta raden och ser hur många rätt man fick.

I denna uppgift kan ni förutsätta att det finns en gissad/tippad rad och att de sju numren finns i en array enligt:

```
int [] minaNummer = new int[7];  
//Arrayen är ifylld och innehåller 7 stycken olika tal
```

Er uppgift är att skriva programkod som först slumpar fram den rätta raden, alltså som slumpar fram sju stycken tal mellan 1 och 35. Obs att det i den rätta raden INTE får finnas några dubletter.

Därefter skall ni kontrollera numren i arrayen ovan och se hur många rätt det blev. Antalet rätt skall sedan skrivas ut på skärmen.

Ni behöver inte tänka på att programsatserna borde stå inuti någon metod, det räcker alltså att ni i ert svar skriver de programsatser som behövs.

Uppgiften består av två delar:

Del A: Slumpa fram den rätta raden. Skall alltså vara sju stycken tal i intervallet 1 – 35. Det får INTE vara några dubletter.

Del B: Kontrollera arrayen minaNummer (se ovan) och se hur många av de numren som finns i den rätta raden (alltså bland de tal ni slumpade fram i del A). Skriv ut resultatet.

Ledning:

I klassen Random finns en metod som slumpar fram ett tal inom ett intervall:

```
public int nextInt(int n)
```

Returnerar ett slumpmässigt tal mellan 0 (inklusive) och n (exklusive). Alltså kan talet 0 bli framslumpat men inte talet n.

Klass Random ligger i java.util, ni kan förutsätta att: **import java.util.*;** är gjort.

Uppgift 4 (6 poäng)

Skriv ett litet Java-program (en klass med en main-metod) som ber användaren mata in ett antal positiva heltal och som sedan skriver ut de talen på tre olika rader. På första raden skall alla tal i intervallet 1-9 skrivas ut, på den andra raden skall alla tal i intervallet 10-100 skrivas och på tredje raden skall talen som är större än 100 skrivas ut.

Programmet frågar användaren efter tal så länge det kommer positiva tal. Så fort 0 eller ett negativt tal blir inmatat avbryts inmatningen. Därefter skrivs talen ut på tre rader. När programmet körs skall det se ut ungefär så här:

```
Tal 1> 19
Tal 2> 863
Tal 3> 43
Tal 4> 11
Tal 5> 5
Tal 6> 31
Tal 7> 2907
Tal 8> 84
Tal 9> 712
Tal 10> 8
Tal 11> 588
Tal 12> 68
Tal 13> 10
Tal 14> 3
Tal 15> -3

5 8 3
19 43 11 31 84 68 10
863 2907 712 588
```

Uppgift 5 (6 poäng)

Följande klasser finns:

```
abstract class Figur{
    private int längd;

    public int getLängd(){
        return längd;
    }
}
class Kvadrat extends Figur{

    public int getOmkrets(){
        return getLängd()*4;
    }
}
class Triangel extends Figur{

    public int getOmkrets(){
        return getLängd()*3;
    }
}
```

Klasserna innehåller även lämpliga konstruktorer, ej relevanta för denna uppgift.

Ni får (om ni så önskar) lägga till saker i klasserna ovan. Redovisa i så fall vad ni vill lägga till och var (i vilken/vilka klasser) ni vill lägga till det.

Er uppgift är att skriva en metod som får en **ArrayList<Figur>** som argument. Den ArrayListan är fylld med blandade Kvadrat- och Triangel-objekt. Ni skall sedan i metoden skriva ut vilken omkrets som är störst av alla kvadraterna i listan samt vilken omkrets som är störst av alla trianglarna i listan. Utskriften skall se ut ungefär så här:

Största kvadraten har omkretsen: 72

Största triangeln har omkretsen: 84

Ni kan förutsätta att det i argumentlistan finns minst ett Kvadrat-objekt och minst ett Triangel-objekt, det finns alltså en största både kvadrat och triangel.

OBS – metoden som ni skriver måste ju i Java ligga inuti någon klass. Det kan ni ignorera i denna uppgift, det räcker alltså med själva metoden. Utskrifterna som görs skall göras i terminalfönstret (alltså med hjälp av System.out).