

OOP Tentamen 2009-01-16 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 tal1=7, tal2=12;

if (tal1++ > 7)
    System.out.println(tal1);
else
    System.out.println(tal2);

tal2+=tal1;
System.out.println(tal1 + tal2);
```

B

```
int y=8;

for (int x=y; x>3; x-=2){
    System.out.println(y);

    for (int z=x; z<y; z++)
        System.out.print(z+" ");

    System.out.println(x);
}
```

C

```
int[] arr = new int[7];
int tal=8;

for (int x=0; x<arr.length; x++)
    if (x<tal)
        arr[x]=tal--;
    else
        arr[x]=x;

for (int x=0; x<arr.length; x++)
    System.out.println(arr[x]);
```

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**

```
class Uppgift2{
    public static void main(String[]args){
        Knyrr[]alla = new Knyrr[4];
        alla[0]=new Hooke("Delfin");
        alla[1]=new Simm("Val");
        alla[2]=new Hooke("Mandel");
        alla[3]=new Simm("Para");
        for (int x=0; x<alla.length; x++)
            System.out.println(alla[x]);
    }
}
class Knyrr{
    private static int antal=0;
    private String grej="nöt";
    public Knyrr(){
        this("Hassel");
    }
    public Knyrr(String sort){
        grej=sort+grej;
        antal++;
    }

    public String toString(){
        antal--;
        String tmp="Citron ";
        for (int x=0; x<antal; x++)
            tmp += grej + " ";

        return tmp;
    }
}
class Hooke extends Knyrr{
    private String namn="Potatis";
    public Hooke(String str){
        namn=str;
    }
    public String toString(){
        return namn;
    }
}
class Simm extends Knyrr{
    public Simm(String str){
        super(str);
    }
    public String toString(){
        return "Apelsin " + super.toString();
    }
}
```

Uppgift 3 (6 poäng)

Skriv ett Java-program som frågar användaren efter ett datum på formen ÅÅÅMMDD, t.ex. 20090116. Programmet skall sedan skriva ut datumet på formen Dag Månad År, t.ex. 16 Januari 2009.

Om användaren matar in ett felaktigt datum skall ett felmeddelande skrivas ut och användaren få försöka igen. Fel datum kan vara antingen att användaren matar in något annat än siffror eller att användaren matar in ett ogiltigt datum (t.ex. månad 14 eller dag 37). Ni kan i denna uppgift strunta i skottår och räkna med att februari alltid har 28 dagar.

Programmet skall loopa tills ett korrekt datum matats in.

Programdialogen skall se ut ungefär så här:

```
Ange datum: 20091422  
Felaktigt datum, försök igen!
```

```
Ange datum: Måndag  
Felaktigt datum, försök igen!
```

```
Ange datum: 19740230  
Felaktigt datum, försök igen!
```

```
Ange datum: 20040817  
17 Augusti 2004
```

(i exemplet ovan så var första försöket fel eftersom 14 är en felaktig månad, i andra försöket är felet att "Måndag" inte är siffror, i tredje försöket är felet att månad 2 (februari) inte har 30 dagar)

Så fort ett korrekt datum lästs in så skall resultatet skrivas ut och programmet avslutas.

Uppgift 4 (6 poäng)

Skriv ett litet Java-program som frågar efter ett antal djursorter (t.ex. “Elefant”, “Ekorre”). Programmet skall fråga efter djursorter tills man på frågan om ny sort matar in ordet “Stopp”. Därefter skall programmet fråga efter hur många djur som finns av varje sort. Dialogen kan se ut ungefär så här:

```
Ange djursort: Elefant
Ange djursort: Ekorre
Ange djursort: Kanin
Ange djursort: Tiger
Ange djursort: Gorilla
Ange djursort: Igelkott
Ange djursort: Stopp
Hur många Elefant? 7
Hur många Ekorre? 12
Hur många Kanin? 37
Hur många Tiger? 4
Hur många Gorilla? 17
Hur många Igelkott? 28
```

Efter inmatningen skall programmet skriva ut hur många djur det fanns sammanlagt samt vilket djur det fanns flest av och hur många det fanns av det djuret. Med inmatningsexemplet ovan skulle det då skrivits ut:

```
Det finns sammanlagt 105 djur.
Flest djur är Kanin med 37 stycken.
```

Det är inte viktigt att dialog och utskrift ser ut exakt så här men skall innehålla dessa delar. Ni kan förutsätta att användaren matar in minst två djursorter. Om det finns lika många djur av flera sorter som är flest så kan ni skriva ut vilken som helst av dessa.

Uppgift 5 (6 poäng)

Följande klasser finns:

```
abstract class Frukt{
    private int vikt;

    public int getVikt(){
        return vikt;
    }
}
class Apelsin extends Frukt{
    private int diameter;

    public int getDiameter(){
        return diameter;
    }
}
class Banan extends Frukt{
    private int längd;

    public int getLängd(){
        return längd;
    }
}
```

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<Frukt>** som parameter. ArrayListan är fylld med blandade Apelsin- och Banan-objekt. Ni skall sedan i metoden skriva ut en lista där först alla Apelsin-objekt skrivs ut och därefter alla Banan-objekt. Vid utskriften skall framgå vilken sorts objekt det är, objektets vikt samt för apelsinerna deras diameter och för bananerna deras längd.

Utskriften skulle t.ex. kunna se ut så här:

```
Apelsin: Vikt = 17, Diameter = 3
Apelsin: Vikt = 23, Diameter = 4
Apelsin: Vikt = 16, Diameter = 3
Banan: Vikt = 18, Längd = 9
Banan: Vikt = 15, Längd = 7
```

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