OOP Tenta 2011-01-14 16.00 – 21.00

Avser kurserna: DSK1:OOP
GES:OOP
ID1013

Anvisningar

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 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

```
String str="Älg";
int tal1=3, tal2=4;
if (tal1<tal2)
     str+=tal1++;
else
     str+=tal2--;
System.out.println(str+tal2);
if (tal1<tal2)
     str+=tal1++;
else
     str+=tal2--;
System.out.println(str+tal2);
```

B

```
for (int x=13; x<20; x+=3){
     int z=x-8;
     for (int y=x; y>z; y-=2)
          System.out.print(y+" ");
     System.out.println(x);
}
```

C

```
int[]talen=new int[7];
for (int x=0; x<talen.length; x++)
     talen[x]=10-x;
int z=1;
while (z<talen.length){
     talen[z]+=z++;
     ++z;
}
for (int tal : talen)
     System.out.println(tal);
```

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){
      Adda[]alla=new Adda[5];
      for (int x=0; x<alla.length; x++)
         if (x==1 || x==3)
            alla[x]=new Dokee();
         else alla[x]=new Bjubb();

      for (int x=0; x<alla.length; x++)
         alla[x].skriv();
   }
}

class Adda{
   private String ord="Lire";
   private int antal=2;

   public Adda(String ord, int antal){
      this.ord=ord;
      this.antal=antal;
   }

   public Adda(){
      ord="Pesetas";
   }

   public void skriv(){
      for (int x=0; x<antal; x++)
         System.out.print(ord);
      System.out.println();
   }
}

class Bjubb extends Adda{
   private static String[] ord={"Kronor","Dollar","Sedlar","Pengar"};
   private static int pos=0;

   public Bjubb(){
      super(ord[pos++], pos);
   }
}

class Dokee extends Adda{
   private String sak="Euro";

   public void skriv(){
      super.skriv();
      System.out.println(sak);
   }
}
```

Uppgift 3 (6 poäng)

Ni skall skriva programsatser som hanterar en heltalsarray. Ni kan förutsätta att ni har tillgång till en heltalsarray enligt deklarationen:

```
int[] talen;
```

Arrayen är skapad med en jämn storlek, dvs arrayen kan vara t.ex. 12 eller 14 lång men inte 13 eller 15 lång. Ni kan också förutsätta att arrayen är fylld med unika heltal i intervallet 0-999. Ni behöver således inte ha någon inläsning av tal till arrayen. Att arrayen innehåller *unika* heltal betyder att det **inte** finns några dubletter i arrayen.

Er uppgift är att sortera arrayen på ett speciellt sätt. Man vill i arrayens första ruta ha det största talet, i den andra rutan det minsta talet, i den tredje rutan det näst största talet, i den fjärde rutan det näst minsta talet osv.

Om arrayen t.ex. hade varit 8 lång och innehållit dessa tal:

```
12  814  37  116  334  418  7  274
```

så skall den efter att er kod har körts innehålla samma 8 tal, nu i ordningen:

```
814  7  418  12  334  37  274  116
```

Hade arrayen istället varit 14 lång och innehållit:

```
7  87  9  19  116  23  98  333  43  184  18  56  101  12
```

så skall den efter att er kod körts innehålla talen i ordningen:

```
333  7  184  9  116  12  101  18  98  19  87  23  56  43
```

All kod i Java måste stå inom en metod. Ni kan bortse från detta i denna uppgift, det räcker alltså att ni skriver de programsatser som utför sorteringen enligt uppgiftstexten.

Ledning: Skapa gärna en temporär hjälparray.

Uppgift 5 (6 poäng)

Följande klasser finns:

```
abstract class Djur{
   private String namn;

   public String getNamn(){
      return namn;
   }
}
class Hund extends Djur{
   private int ålder, vikt;

   public int värde(){
      return vikt*10+ålder*75;
   }
}
class Katt extends Djur{
   private boolean snäll;

   public int värde(){
      if (snäll)
         return 700;
      else return 300;
   }
}
```

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<Djur>`** som argument. Den ArrayListan är fylld med blandade Hund- och Katt-objekt.

Ni skall i metoden skriva ut namn, ålder och vikt för alla Hund-objekt som har ett värde som överstiger 500. Utskriften skulle kunna se ut så här:

Fido Ålder=4 Vikt=32
Karo Ålder=6 Vikt=8
Rufus Ålder=4 Vikt=21
Lex Ålder=8 Vikt=4

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).