Institutionen för TENTAMEN
Data- och informationsteknik 2018-08-24

PROGRAMMERINGSTEKNIK, del A
TIN213

OBS! Det kan finnas kurser med samma eller liknande namn på olika utbildningslinjer. Denna tentamen gäller *endast* för den eller de utbildningslinjer som anges ovan. Kontrollera därför noga att denna tentamen gäller för den utbildningslinje du själv går på.

TID: 08:30 - 11:30 SAL: Johanneberg

Ansvarig: Robin Adams, tel. 031 772 63 48

Betygsgränser: Sammanlagt maximalt 30 poäng.
På tentamen ges graderade betyg:
3:a 15 poäng, 4:a 20 poäng, 5:a 25 poäng

Hjälpmedel: Skansholm, *Java direkt med Swing*, Studentlitteratur, eller
Bravaco, Simonson, *Java Programming: From the Ground up*
(Understrykningar och mindre anteckningar i boken är tillåtna.)

Inga kalkylatorer är tillåtna.

Tänk på:

- att skriva tydligt och disponera papperet på ett lämpligt sätt.
- att börja varje ny uppgift på nytt blad. Skriv endast på en sida av papperet
- Skriv den (anonyma) kod du fått av tentamensvakten på *alla* blad.

De råd och anvisningar som givits under kursen ska följas vid programkonstruktionerna. Det innebär bl.a. att onödigt komplicerade, långa och/eller ostrukturerade lösningar i värsta fall ej bedöms.

Uppgift 1)

a) Förklara (säkerhets)problemet med följande kodsnutt:

```
public class C {
  private double[] a = new double[10];
  ...
  public double[] getA() {
    double[] aCopy = a;
    return aCopy;
  }
}
```

(2 p)

b) I en statistisk undersökning har deltagarna fått svara på frågor av typen "Hur många personer är det i ditt hushåll?" Svaren har lagrats i en array med heltal, med ett element per deltagare. Arrayen är inte sorterad på något sätt.Vid utvärderingen av svaren vill man skapa en frekvenstabell som anger hur många av deltagarna som angett de olika svarsalternativen, t.ex. 1 person: 10 svar, 2 personer: 12 svar, 3 personer: 9 svar, osv. Uppgiften är att skriva en klassmetod som skapar och returnerar frekvenstabellen. Metoden ska ha tre parametrar. Den första är arrayen med deltagarnas svar. De andra två parametrarna anger lägsta resp. högsta tillåtna svarsvärde. Dessa är till för att man i metoden ska kunna hoppa över orimliga värden, som t.ex. 0 personer eller 30 personer. Du ska alltså bara räkna sådana svar som ligger innanför dessa gränser. Frekvenstabellen som metoden ska returnera ska vara en array med heltal, där varje element anger antal svar. Det första elementet i arrayen ska ange antalet svar för det lägsta tillåtna värdet, det andra för det näst lägsta värdet osv. Det sista elementet i arrayen ska innehålla antalet svar för det högsta tillåtna värdet.

(6 p)

Uppgift 2)

a) Medianen av en mängd tal är ett tal sådant att antalet tal större än medianen är lika med antalet tal mindre än medianen. Om mängden innehåller ett jämnt antal tal så beräknas medianen som medelvärdet av de två "mittersta" talen. Uppgiften är att konstruera en klassmetod `median` som får en array (ett fält) med reella tal som parameter och som beräknar och returnerar medianen. Arrayen som ges som parameter får *inte* ändras av metoden. Talen i den ska alltså efter anropet ligga kvar i samma ordning som före anropet.

(6 p)

b) Skriv ett program som läser in ett antal tal från en textfil och som beräknar och visar talens median i en dialogruta. Programmet ska inleda med att be användaren skriva in filens namn i en dialogruta. (För enkelhets skull behöver du inte i programmet kontrollera att filen finns.) Beräkningen av medianen *måste* göras genom att man anropar metoden `median` från deluppgift a. (Det får du göra även om du inte löst den deluppgiften.) Du får anta att metoden `median` är deklarerad i samma klass som metoden `main`.

(6 p)

Uppgift 3) Som du vet kan ett komplext tal beskrivas med hjälp av en realdel, `re`, och en imaginärdel, `im`. (Det kan skrivas som `re+im*i`, där `i*i` = -1.) Din uppgift är nu att definiera en klass i Java som beskriver begreppet komplext tal. Klassen ska (minst) ha följande konstruktorer och metoder:

- en defaultkonstruktor,
- en kopieringskonstruktor,
- en konstruktor som har en realdel och en imaginärdel som parametrar,
- två metoder som används för att avläsa ett komplext tals realdel respektive imaginärdel,
- en metod som adderar ett annat komplext tal till det aktuella komplexa talet (det aktuella komplexa talet ska alltså förändras),
- en metod som multiplicerar det aktuella komplexa talet med ett annat komplext tal (det aktuella komplexa talet ska alltså förändras),
- en metod med namnet `equals` som undersöker om det aktuella komplexa talet är lika med ett annat komplext tal. (När man jämför variabler av typerna **`float`** eller **`double`** bör man tänka på att värdena lagras approximativt. Någon bit kan därför vara olika för variabler som man logiskt uppfattar som "lika". När man jämför två sådana variabler bör man inte jämföra dem direkt, utan istället se om absolutbeloppet av deras skillnad är mycket litet, t.ex. 10^{-15}. I så fall kan man uppfatta variablerna som lika.),
- en metod `toString` som returnerar det komplexa talet som en text med formen `"a+b*i"` eller `"a-b*i"`, beroende på om imaginärdelen är positiv eller inte.

Det ska inte vara möjligt att utifrån avläsa eller ändra ett komplext tal direkt. Allt måste göras via konstruktorer eller metoder.

(10 p)

Lösningar till tentamen, del A 2018-04-06

// Uppgift 1 a

Metoden cCopy returnerar en referens till samma array som den privata variabeln a pekar på.
Detta gör att man utifrån har möjlighet att ändra i arrayen, trots att den deklareats som private.
Metoden aCopy är felkonstruerad. Den borde skapa en kopia av arrayen och returnera en referens till denna istället.

// Uppgift 1b

```
  static int[] frekvens(int[] svar, int min, int max) {
    int[] f = new int[max-min+1];
    for (int i = 0; i < svar.length; i++)
      if (svar[i] >= min && svar[i] <= max)
        f[svar[i]-min]++;
    return f;
  }
```

// Uppgift 2

```
import java.util.*;
import javax.swing.*;
import java.io.*;
public class Statistik {

 // Deluppgift a
 static double median(double[] tal) {
    double a[] = (double[]) tal.clone();  // eller System.arraycopy
        Arrays.sort(a);
        int n = a.length;
        if ( n % 2 != 0)
                return a[(n-1)/2];
        else
                return (a[n/2-1] + a[n/2]) / 2;
  }

  // Deluppgift b
  public static void main(String[] arg) throws FileNotFoundException {
    String filnamn = JOptionPane.showInputDialog("Filens namn? ");
    Scanner Sc = new Scanner(new File(filnamn));
        double[] input = new double[100];
        int n = 0;
        while( Sc.hasNextDouble()) {
          input[n++] = Sc.nextDouble();
        }
        double data[] = new double[n];
    System.arraycopy(input,0, data,0, n);
        JOptionPane.showMessageDialog(null, "Medianen är " + median(data));
 }
}
```

// Uppgift 3

```
public class Complex {
  private static final double epsilon = 1e-15;
```

```
  private double re, im;

  public Complex() {}

  public Complex(Complex z) {
    this(z.re, z.im);
  }

  public Complex(double rr, double ii) {
    re = rr;
    im = ii;
  }

  public double getRe() {
    return re;
  }

  public double getIm() {
    return im;
  }

  public void add(Complex z) {
    re = re + z.re;
    im = im + z.im;
  }

  public void mult(Complex z) {
    double oldre = re;
    double oldzre = z.re;
    re = re*z.re - im*z.im;
    im = oldre*z.im + im*oldzre;
  }

  public boolean equals(Complex z) {
    return Math.abs(re-z.re)<epsilon && Math.abs(im-z.im)<epsilon;
  }

  public String toString() {
    if (im >= 0)
      return re + "+" + im + "*i";
    else
      return re + "" + im + "*i";
  }
}
```