Institutionen för
Data- och informationsteknik

TENTAMEN
2018-08-28

PROGRAMMERINGSTEKNIK, del B
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: Samhällsbyggnad

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) Dessa korta exempel innehåller vart och ett något fel som gör att man inte kan kompilera dem. Ange för varje exempel vad det är för ett fel (skriv radnummer också) och tala om hur man skulle kunna rätta till felet. Programexemplen är helt oberoende av varandra och varje klass och gränssnitt ligger i en egen fil med ett korrekt namn.

(6p)

a)
```
 1 public interface Fordon{
 2    public double beräknaFart(double x, double y);
 3    public void kör();
 4 }
 5
 6 public class Tåg implements Fordon {
 7    public double beräknaFart(double x, double y) {
 8       return (x*y) / (x*y + (1-x)*(1-y));
 9    }
10 }
```

b)
```
11 public class Instrument {
12    protected int n;
13    public Instrument(int n) {
14       this.n = n;
15    }
16 }
17
18 public class Fiol extends Instrument {
19    public Fiol(int n) {
20       this.n = n;
21    }
22 }
```

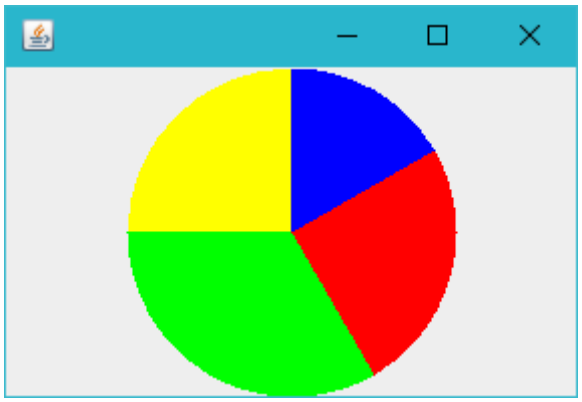
c)
```
23 public interface Bil {
24    public int hastighet();
25 }
26
27 public interface Båt {
28    public double hastighet();
29 }
30
31 public class Amfibie implements Bil, Båt {
32    public int hastighet() {
33       return 70;
34    }
35    public double hastighet() {
36       return 5.0;
37    }
38 }
```

Uppgift 2) Konstruera en klass `Diagram` som ska vara en subklass till standardklass `JPanel`. Objekt av klassen `Diagram` ska kunna visa cirkeldiagram liknande det i figuren. (Du ska inte konstruera hela programmet, bara klassen `Diagram`.) Klassen `Diagram` ska ha en konstruktor samt en metod med namnet `setValues`. Både konstruktorn och metoden ska ha två arrayer (fält) som parametrar. Den första arrayen ska innehålla de värden som ska illustreras i diagrammet. Det kan t.ex. vara antalet röster i ett val. Den andra arrayen ska innehålla de färger som ska användas i diagrammet. De båda arrayerna måste ha lika många element. Om så inte är fallet ska din klass generera en exception av typen `IllegalArgumentException`.

Klassen `Diagram` ska vara så utformad att den grafiska komponenten automatiskt ritas om när fönstret ändras eller när `setValues` eller konstruktorn anropas. Då ska cirkelns diameter bestämmas av den kortaste av komponentens bredd och höjd. (Du

behöver inte i denna uppgift ta hänsyn till eventuella borders.) Rita de olika "tårtbitarna" medurs och börja längst upp i cirkeln (klockan 12) med det värde och den färg som ligger först i arrayerna.

Tips: Låt klassen ha två instansvariabler. Den första kan vara en array där de olika mätvärdena skalats om så att deras summa blir 360. Den andra instansvariabeln kan vara en referens till arrayen med färgerna.



(12 p)

Uppgift 3) Ett företag har lagrat ett register över sina anställda i en textfil. För varje anställd finns det tre rader i filen. Den första raden innehåller den anställdas personnummer, den andra raden personens namn och den tredje namnet på den avdelning där personen arbetar.

Uppgiften är att skriva ett program som läser in informationen i filen och lägger den i en avbildningstabell i vilken söknycklarna är personnumren och värdena är den information som finns på de övriga två raderna för varje anställd. Denna information ska internt i programmet sparas i objekt av klassen `Person` som har följande utseende:

```
class Person {
    public String namn;
    public String avdelning;
    public Person(String na, String avd) {
      namn = na; avdelning = avd;
    }
}
```

Programmet ska först fråga efter filens namn i en dialogruta. Om filen inte finns ska programmet ändra texten i dialogrutan och be användaren försöka igen. Detta ska upprepas tills användaren angett ett korrekt filnamn eller valt av avsluta genom att klicka på knappen *Avbryt* i dialogrutan.

När all information i filen lästs in och lagts i avbildningstabellen ska programmet *upprepade gånger* be användaren ange ett personnummer. Inläsningen ska ske med hjälp av en dialogruta. För varje angivet personnummer ska programmet slå upp den efterfrågade personen i avbildningstabellen och visa all information om honom eller henne i en annan dialogruta. Använd en standardklass för att konstruera avbildningstabellen. Välj en klass som gör att sökningarna i tabellen blir så snabba som möjligt. Personerna behöver inte ligga sorterade i tabellen.

(12 p)

// Uppgift 1

a) Före rad 10: Metoden kör() definieras inte i klassen Tåg
   Man måste ha med en definition av denna metod eftersom klassen Tåg inte är abstrakt.

b) Rad 20: Defaultkonstruktor saknas i klassen Instrument. Den anropas automatiskt från
   konstruktorn i subklassen Fiol.
   Den bästa lösningen är att i klassen Fiol ersätta rad 20
   med ett anrop av konstruktorn i klassen Instrument:
     super(n);

c) Rad 35: Man försöker överlagra metoden hastighet(), men detta är inte tillåtet eftersom
   de två versionerna har samma parameterlista. (Det räcker inte att returtyperna är olika.)
   Den bästa lösningen här är nog att bara ha ett enda interface,
   vilket innehåller en enda version av metoden hastighet().

```
// Uppgift 2

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

class Diagram extends JPanel {
  private double p[];
  private Color c[];

  public Diagram (double val[], Color col[]) {
    setValues(val, col);
  }

  public void setValues(double val[], Color col[]) {
    if (val.length != col.length)
      throw new IllegalArgumentException("Olika längder");
    p = new double[val.length];
    c = col;
    double sum = 0;
    for (int i=0; i<val.length; i++)
      sum += val[i];
    for (int i=0; i<val.length; i++)
      p[i] =  val[i] / sum * 360;
    repaint();
  }

  @Override
  public void paintComponent(Graphics g) {
    super.paintComponent(g);
    int diam = Math.min(getWidth(), getHeight());
    int x = (getWidth() - diam)/2;
    int y = (getHeight() - diam)/2;
    int start = 90;
    for (int i = 0; i < p.length; i++) {
      int del = -(int)(p[i] + 0.5);
      g.setColor(c[i]);
      g.fillArc(x, y, diam, diam, start, del);
      start += del;
    }
  }
```

```
  }


// Uppgift 3
import java.util.*;
import java.io.*;
import javax.swing.*;

public class PersonRegister {
   public static void main(String[] arg) {
      String meddelande = "Ange filnamn";
      Scanner sc = null;
      boolean klar = false;
      while (! klar) {
        String filnamn = JOptionPane.showInputDialog(meddelande);
        if (filnamn == null)
          System.exit(0);
        try {
          sc = new Scanner(new File(filnamn));
          klar = true;
        }
        catch (FileNotFoundException e) {
          meddelande = "Felaktigt filnamn\nFörsök igen!";
        }
      }
      Map<String, Person> tab = new HashMap<>();
      while (sc.hasNext()) {
        String pnr = sc.nextLine();
        Person p = new Person(sc.nextLine(), sc.nextLine());
        tab.put(pnr, p);
      }
      while (true) {
        String pnr = JOptionPane.showInputDialog("Personnummer?");
        if (pnr == null)
          break;
        Person p = tab.get(pnr);
        if (p != null)
          JOptionPane.showMessageDialog(null, p.namn + "\n" + p.avdelning);
        else
          JOptionPane.showMessageDialog(null, "Felaktigt personnummer");
      }
   }
}
```