Lösninsförslag till tentamen FYTA11 — Javaprogrammering

Onsdag 4:e februari 2009, 13:00 – 17:00

Instruktioner

Inga hjälpmedel är tillåtna. Behandla högst en uppgift per sida, och sätt ditt namn och personnummer på varje sida. Skriv läsligt och kommentera utförligt vad du gör – det kan ge dig poäng även om resultatet blir fel. Tentamen omfattar fyra uppgifter som alla kan ge upp till tio poäng.

Uppgift 1: En liten klasshierarki

Nedan visas några enkla klasser för att representera anställda på ett företag. Basklassen är `Employee` som har ett namn och en lön. En chef, som är också en anställd, representeras av klassen `Boss` och har, förutom ett namn och en lön, också en månatlig bonus. Skriv färdigt konstruktorn och `payCheque` metoden i klassen `Boss` så att de fungerar enligt kommentarerna (`Employee` klassen får ej modifieras). Klassen `PayCheques` på nästa sida används för att skriva ut en lönelista i det väldigt lilla företaget PytteMjuk. Vad skrivs ut om man kör `main` metoden?
Skriv om hela klassen PayCheques så att utskriftsmetoden får signaturen.

```
public static void printCheques(Collection<Employee> payroll);
```

men i övrigt fungerar som innan. Använd en lämplig kontainerklass från standardbiblioteket och använd en iterator eller *förenklad* `for`-sats till loopen i utskriftsmetoden. Om du är osäker på hur metoder i standardbiblioteket är deklarerade är det tillåtet att göra välmotiverade och väldokumenterade antaganden.

```
/** This class represents an employee of a company
 *  in a managerial position. */
public class Boss extends Employee {

  private double theBonus;

  /** Create an employee with a name, salary and a
   *  monthly bonus. */
  public Boss(String name, double salary,
              double bonus) {
    super(name, salary);
    theBonus = bonus;
  }

  /** Return the monthly amount to be payed to the
   *  employee, which in the Boss case should
   *  include the bonus.*/
  public double payCheque() {
    return super.payCheque() + theBonus;
  }

  /** Return the bonus. */
  public double bonus() {
    return theBonus;
  }

}
```

```
import java.util.*;

public class PayCheques {

  public static void main(String[] args) {

    Vector<Employee> staff =
      new Vector<Employee>();
    staff.add(new Employee("John Doe",
                           10000.0));
    staff.add(new Employee("Per Persson",
                           12000.0));
    staff.add(new Boss("Jill Bates",
                       30000.0, 55000.0));
    printCheques(staff);

  }

  public static void
    printCheques(Collection<Employee> payroll) {
    for ( Employee e: payroll )
      System.out.println("Name: " +
                         e.name() +
                         ", amount: " +
                         e.payCheque());
  }

}
```

```
% java PayCheques
Name: John Doe, amount: 10000.0
Name: Per Persson, amount: 12000.0
Name: Jill Bates, amount: 85000.0
```

Uppgift 2: Sortera

Klassen `BubbleSort` nedan kan användas för att sortera ett godtyckligt fält med heltal. Skriv om klassen så att den istället sorterar ett fält med godtyckliga objekt med hjälp av ett `Comparator` objekt. Sorteringsmetoden skall ha signaturen

`public static void sort(Object [] v, Comparator cmp)`.

Gränssnittet `Comparator` är deklarerat i paketet `java.util` enl. nedan. Om `Comparator` argumentet är `null` ska metoden ändå sortera vektorn under antagandet att objekten i fältet implementerar standardgränssnittet `Comparable`.

```
import java.util.*;

public class BubbleSort {

  public static void sort(Object [] v, Comparator cmp) {
    for ( int i = v.length - 1; i > 0 ; --i )
      for ( int j = 0; j < i; ++j )
        if ( cmp == null ) {
          if ( ((Comparable)v[j]).compareTo(v[j + 1]) > 0 ) swap(v, j, j + 1);
        } else {
          if ( cmp.compare(v[j], v[j + 1]) > 0 ) swap(v, j, j + 1);
        }
  }

  public static void swap(Object [] v, int i, int j) {
    Object tmp = v[i];
    v[i] = v[j];
    v[j] = tmp;
  }

}
```

Uppgift 3: Medelvärde

Skriv ett program som skriver ut medelvärdet och variansen av de argument som ges när det startas. Exempelvis ska

```
% java Medel 0.25 10 3 -25 0
```

ge en utskrift liknande

```
Average: -2.35, Variance: 11.886547017532047
```

Om ett argument inte är ett tal ska ett snällt felmeddelande skrivas ut men medelvärdet av övriga numeriska argument ska ändå vara korrekt. Följande metod i klassen `Double` kan vara bra att känna till

```
public class Medel {

  public static void main(String [] args) {

    double sx = 0.0;
    double sx2 = 0.0;
    int n = args.length;

    for ( String arg: args ) {
      try {
        double x = Double.parseDouble(arg);
        sx += x;
        sx2 += x*x;
      }
      catch (NumberFormatException e) {
        System.out.println("'" + arg + "' is not a real number.");
        --n;
      }
    }
    sx /= n;
    sx2 = Math.sqrt(sx2/n - sx*sx);
    System.out.println("Average: " + sx + ", Variance: " + sx2);

  }

}
```

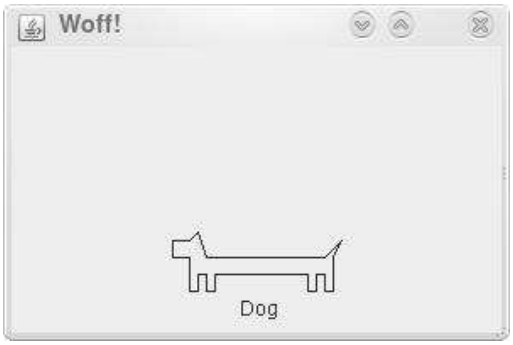
Uppgift 4: Voff!

Följande klass ritar en hundliknande form på en `Canvas`:

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
import java.net.*;

class Dog extends Canvas {
  AudioClip bark;
  Polygon dog;

  public Dog() {
    setSize(300,300);
    int [] x = {100, 100, 110, 110, 115, 115, 120, 120, 125, 125, 180,
                180, 185, 185, 190, 190, 195, 195, 200, 190, 120, 115, 110};
    int [] y = {110, 120, 120, 140, 140, 130, 130, 140, 140, 130, 130,
                140, 140, 130, 130, 140, 140, 120, 110, 120, 120, 105, 110};
    dog = new Polygon(x, y, x.length);
    try {
      bark = Applet.newAudioClip(new URL("http://www.dog.org/sounds/bark/wienerdog.au"));
    } catch (MalformedURLException e) {
      bark = null;
    }
    addMouseListener(new MouseAdapter() {
        public void mouseClicked(MouseEvent e) {
          if ( dog.contains(e.getPoint()) )
            if ( bark != null ) bark.play();
        }
      });
  }

  public void paint(Graphics g) {
    g.drawPolygon(dog);
    g.drawString("Dog", 140, 155);
  }
}
```

Skapa ett objekt av en anonym klass i anropet till `addMouseListener` så att hunden skäller varje gång man klickar på den (det räcker altså inte att man klickar i fönstret i allmänhet, man ska behöva klicka på själva hunden). Behövs det någon annan kod för att klassen ska fungera? Det går att lösa uppgiften utan att använda en anonym klass, men då får man inte full poäng. Nedan följer beskrivningar av diverse metoder i diverse klasser som kan vara bra att känna till: