

Tentamen i Programmeringsteknik I, , KandMa/Fy, 2010-10-19

- Skriv tid: 14-17.
- Hjälpmedel:
 1. Kurslitteratur, en av följande:
 - Lewis & Loftus, Java Software Solutions
 - Skansholm, Java Direkt
 - Guzdial & Ericson, Introduction to computing & programming.Litteraturen får inte innehålla överdrivet mycket anteckningar.
 2. Penna, radergummi, linjal.
- Tentamen består av fyra uppgifter och maxpoängen är 24.
- För att bli godkänd på tentamen (betyg 3) krävs minst 15 poäng
- För att få betyg 4 på tentamen krävs minst 18 poäng.
- För att få betyg 5 på tentamen krävs minst 21 poäng.

Instruktioner:

- Programmen skall vara skrivna i Java om inte annat anges.
- Programmen skall vara skrivna med vettig layout och förståeliga. Tydliggör gärna matchande måsvingar.
- Skriv varje uppgift på nytt papper.
- Skriv inte på baksidan.
- Använd inte rödpenna.
- Läs uppgifterna noga så du vet vad som skall göras.
- Gör inte mer än det som efterfrågas.
- Även ofullständiga lösningar och/eller lösningsskisser kan ge poäng om de visar att du tänkt rätt
- Läs igenom uppgiften igen när du löst den så att ser att du verkligen gjort rätt saker.
- import-satser behöver inte skrivas.

Lycka till!

Torsten

Uppgift 1 (6p)

En klass `Badkar` skall ha följande egenskaper:

- Storleken på badkaret uttryckt i antal liter
- Antal liter vatten som just nu finns i badkaret
- Modellbeteckning för badkaret

Skriv klassen `Badkar` som skall innehålla:

- Instansvariabler för de tre egenskaperna
- Två konstruktorer. En utan parameter och en med parametrar med vilka man kan skapa ett `badkar` med en preciserad storlek och modellbeteckning.
- En metod `fill` som fyller på badkaret med ett givet antal liter vatten. Det givna antalet skall ges som parameter till metoden. Obs! Metoden måste kontrollera att inte badkaret svämmar över och i det fallet bara fylla det så att det blir precis fullt.
- En metod `tom` som tömmer allt vatten från badkaret.
- En metod `getFyllnadsgrad` som returnerar hur mycket uttryckt i procent som badkaret är fyllt med vatten. Ex. ett 200-liters badkar som är fyllt med 150 liter har en fyllnadsgrad av 75%.
- En `toString`-metod för att erhålla information om badkaret.

Skriv också en `main`-metod i en klass `TestBadkar` som

- Skapar ett 200-liters badkar med modellbeteckningen `BADTUNNAN`.
- Skriver ut information om badkaret
- Fyller badkaret med 150 liter
- Beräknar fyllnadsgraden (m.h.a. metoden `getFyllnadsgrad`) och skriver ut det värdet.
- Tömmer badkaret.
- Fyller badkaret med 50 liter
- Skriver ut information om badkaret

En körning av `main`-metoden kan se ut så här:

```
Badkaret BADTUNNAN. Kapacitet=200. Fyllt med 0 liter
Nu fyllt till 90.0 procent
Badkaret BADTUNNAN. Kapacitet=200. Fyllt med 50 liter
```

Uppgift 2 (6p)**a)**Antag att klassen `Boll` existerar och ett skal av klassen visas här:

```
public class Boll {

    private double x,y,z,r;

    public Boll() { }

    public Boll(double inx, double iny, double inz,
                double inr) {
        ...
    }

    public void flytta(double dx, double dy, double dz) {
        ...
    }

    public boolean jämför( Boll b ) {
        ...
    }

} // Slut klassen Boll
```

Ange vilken/vilka av satserna (1)-(8) i mainmetoden som är felaktiga i mainmetoden:.

```
public class TestBoll {
    public static void main(String [] arg) {
        Boll.flytta(1.0,2.0,3.0,4.0);           // (1)
        Boll b = new Boll(1.0,1.0,1.0,4.0);    // (2)
        System.out.println(b.flytta(1.0,1.0,1.0)); // (3)
        double r=4.0,s=5.6, t=78.5;
        b.flytta(r,s,t);                       // (4)
        b.flytta(r+0.5,8.5,t*2);              // (5)
        b.flytta(x,y,z);                       // (6)
        Boll b2;
        boolean ok1 = b2.jämför(b);           // (7)
        boolean ok2;
        ok2 = b.jämför(new Boll(1.0,2.0,3.0,4.0)); // (8)
    }
} // Slut klassen TestBoll
```

b)En programkörning ser ut så här (**fet stil** är inmatning):

Hur många heltal vill du läsa in?

3

Ge talen:

4**12****8**

Talen i omvänd ordning är:

8**12****4**

Skriv de satser som utför detta, dvs frågar efter ett antal tal, läser in dem och skriver dem i omvänd ordning. Ingen komplett klass eller komplett mainmetod behöver skrivas.

Antag att följande sats är given:

```
Scanner scan = new Scanner(System.in);
```

c)

Antag att följande matematiska uttryck för vektoraddition skall beräknas:

$$V_4 = V_1 + V_2 + V_3$$

Du skall nu uttrycka denna addition i java m.h.a. en klass Vektor.

```
public class Vektor {

    private double x, y;

    public Vektor() {
        this.x=0; this.y=0;
    }

    public Vektor(double initX, double initY) {
        this.x=initX; this.y=initY;
    }

    public Vektor add(Vektor inVektor) {
        Vektor outVektor = new Vektor();
        double tempX, tempY;
        tempX = this.x + inVektor.x;
        tempY = this.y + inVektor.y;
        outVektor.x = tempX;
        outVektor.y = tempY;
        return outVektor;
    }

} // klassen Vektor
```

Skriv de satser i nedanstående mainmetod som utför additionen.

```
public static void main (String[] arg) {

    Vektor v1 = new Vektor(1.0,2.0);
    Vektor v2 = new Vektor(3.0,1.0);
    Vektor v3 = new Vektor(-1.0,4.0);
    Vektor v4;

    // Skriv den eller de satser som utför addition av
    // vektorerna v1,v2 och v3 och lägger resultatet i v4.
    // Det är tillåtet att deklarera och använda extra variabler
    // för att lösa uppgiften
    ...

}
```

Uppgift 3 (6p)

Resultat i fotbollsmatcher poängsätts på följande sätt:

Vunnen match ger 3 poäng, oavgjord 1 poäng, förlorad ger noll poäng.

Om ett lag har följande matchresultat: 3-2 (dvs en vinst, gjorda mål 3, insläppta mål 2), 1-1 och 0-3 får man sammanlagt 4 poäng.

Givet är följande mainmetod:

```
public static void main (String[] arg) {
    MatchResultat m = new MatchResultat();
    m.nyttResultat(3,2); // Lägg in resultatet 3-2 (3 gjorda,2 insläppta)
    m.nyttResultat(3,3); // Lägg in resultatet 3-3
    m.nyttResultat(0,4); // Lägg in resultatet 0-4
    m.nyttResultat(0,0); // Lägg in resultatet 0-0

    System.out.println(m.poäng()); // Skriv ut sammanlagda poängen.
}
```

Skriv klassen `MatchResultat` så att den passar mainmetoden. Klassen skall lagra alla resultat i två arrayer, en för gjorda mål och en för insläppta mål, från första matchen till sista matchen. Antag att arrayernas längd kan sättas till 100 eftersom man aldrig någonsin lagrar fler än 100 matchresultat. Notera att klassen består av en konstruktor och metoderna `nyttResultat` och `poäng`.

Uppgift 4 (6p)

Givet är klasserna `Bostad` och `Bostadsregister`:

```
import java.util.Scanner;
public class Bostad {
    private int rum; // antal rum
    private double boyta; // bostadens yta
    private String adress; // bostadens adress

    public Bostad() { }

    public Bostad ( String a, int r, double b) {
        this.rum=r;
        this.boyta=b;
        this.adress=a;
    }

    public int getRum() {
        return this.rum;
    }

    public double getBoyta() {
        return this.boyta;
    }

    public void läs() {
        Scanner scan = new Scanner(System.in);
        System.out.println("Adress:");
        this.adress = scan.next();
        System.out.println("Bostadsyta:");
        this.boyta = scan.nextDouble();
        System.out.println("Antal rum:");
        this.rum = scan.nextInt();
    }
}
```

```

public String toString() {
    return "Adress " + this.adress + ". Antal rum=" +this.rum
        + ". Bostadsyta=" + this.boyta;
}
} // Slut klassen Bostad

public class BostadsRegister {
    private Bostad [] arrBostad = new Bostad[10];
    private int antal;

    public BostadsRegister () { }

    public void nyBostad(Bostad b) {
        arrBostad[antal]=b;
        antal++;
    }

    public int getAntalBostäder() {
        return this.antal;
    }

    public Bostad getBostad(int index) {
        return arrBostad[index];
    }
} // Slut klassen BostadsRegister

```

a)

Skriv de satser i mainmetoden i klassen TestBostadsRegister som utför följande:

- Skapar ett tomt bostadsregister.
- Frågar användaren efter hur många bostäder som denne vill lägga in
- Frågar användaren efter data för dessa bostäder, en efter en, och lägger in dem i registret
- Skriver ut information om alla bostäder i registret
- Skriver ut information om alla 2-rumsbostäder i registret

```

public class TestBostadsRegister {

    public static void main (String[] arg) {
        Scanner scan = new Scanner(System.in);
        ...
    }
} // Slut klassen TestBostadsRegister

```

Notera: Klasserna Bostad och BostadsRegister skall inte ändras i uppgift a).

b)

Modifiera metoden `nyBostad` i klassen `BostadsRegister` så att den klarar att hantera ett godtyckligt antal bostäder.

c)

Skriv en metod `tabortBostad` i klassen `BostadsRegister` som tar bort en given bostad. Metoden skall ha ett heltal som parameter som anger index på den bostad som skall tas bort.