

Tentamen i Programmeringsteknik I 2016-10-17

Skrivtid: 8:00 – 13:00

Tänk på följande

- Skriv läsligt! Använd *inte* rödpenna!
- Skriv bara på framsidan av varje papper.
- Lägg uppgifterna i ordning. Skriv uppgiftsnummer och tentamenskod *överst i högra hörnet* på alla papper.
- Uppgift A1 skall rivas ur från tentamen och lämnas in tillsammans med övriga svar som skrivs på lösa papper.
- Fyll i försättssidan fullständigt.
- Alla uppgifter gäller programmeringsspråket Java och programkod skall skrivas i Java. Koden skall vara läslig dvs den skall vara vettigt strukturerad och indenterad. Namn på variabler, metoder, klasser etc skall vara beskrivande men kan ändå hållas ganska korta.

Observera att betyget påverkas negativt av

- icke-privata eller onödiga instansvariabler,
 - dålig läslighet,
 - upprepning av identisk kod och
 - underlåtenhet att utnyttja given eller egen tidigare skriven metod.
- För att bli godkänd (betyg 3) krävs att A-delen är i stort sett rätt löst
- För betyget 4 krävs dessutom att minst hälften av uppgifterna på B-delen och betyg 5 att alla uppgifter på B-delen är i stort sett lösta. Vi bedömningen av betyg 4 och 5 tas också hänsyn till kvalitén på lösningarna i A-delen.

Observera att B-delen inte rättas om inte A-delen är godkänd.

Lycka till!

Torsten

Del A (obligatorisk för alla)**A1.** Ringa in rätt svar och lämna in denna sida tillsammans med dina övriga svar

a) Givet koden

```
int x=36, y=10, z;
z = x/y;
z++;
```

Vilket värde har `z` efter sista satsen?

- 1. 4
- X. 4.6
- 2. 5

b) Givet är koden

```
int [] x = new int[3];
x[1]=2; x[2]=-4;
x[3]=x[1]+x[2];
System.out.println(x[3]);
```

- 1. Ger utskriften -2
- X. Medför `ArrayIndexOutOfBoundsException`
- 2. Ger utskriften null

c) Givet koden:

```
Vehicle [] theLane = new Vehicle[10];
double p = Math.random();
Vehicle v1;
Vehicle v2 = new Vehicle('W');
```

Hur många `Vehicle`-objekt skapas? (Räkna bara de objekt som man ser att koden skapar).

- 1. 2 st
- X. 1 st
- 2. 11 st

d) En konstruktor

- 1. skapar en ny klass
- X. anropas alltid när ett objekt skapas
- 2. har returtypen void

e) Vad är korrekt angående `toString`-metoden?

- 1. Har en parameter.
- X. Skriver ut en text
- 2. Returnerar en `String`

f) Vilken returtyp skall metoden nedan ha, dvs vad ska ... bytas mot?

```
public ... do(int n) {
    double [] x = new double[n];
    for (int i=0; i<n; i++) {
        x[i]=i;
    }
    return x;
}
```

- 1. int
- X. void
- 2. double []

g) Givet följande kod:

```
Turtle [] family = new Turtle[3];
int x = family.length;
family[0].move(20);
System.out.println(x);
```

- 1. Medför `java.lang.NullPointerException`
- X. Medför `ArrayIndexOutOfBoundsException`
- 2. Medför utskrift av 3

h) Givet koden:

```
double a=8.5;  
double x = methA(a);  
double y = methA(14.5);
```

Vilket av vidstående påståenden är sant?

- 1. a är en formell parameter
- X. a är en aktuell parameter
- 2. 14.5 är en formell parameter

i) Givet koden

```
ArrayList<Person> plist = new ArrayList<Person>();  
Person p = new Person();  
plist.add(p);
```

Vilken av följande satser är korrekta?

```
int x = plist.get(0);      // a  
double y = plist.get(0);  // b  
Person z = plist.get(0);  // c
```

- 1. Sats a
- X. Sats b
- 2. Sats c

j) Vilken sats ger med säkerhet en felutskrift?

```
World w = new World(); // a  
Turtle t;                // b  
t.move(20);              // c
```

- 1. Sats a
- X. Sats b
- 2. Sats c

A2.

Skriv en klass `TicketMachine` som skall representera en biljettautomat som har instansvariabler för biljettens identitet, priset och antalet tillgängliga biljetter. Namnge dem som `identity`, `price` och `numberOfTickets`. Det finns en mainmetod nedan som visar hur klassen skall fungera och en programkörning. Skriv:

- Instansvariablerna
- Konstruktorn, se mainmetoden hur den skall fungera.
- Metoden `getPrice` som returnerar biljettens pris
- Metoden `buy` med vilken man köper ett antal biljetter som då reduceras i biljettautomaten och metoden returnera `true`, annars skall inga biljetter köpas (antalet biljetter ej ändras) och metoden returnera `false`.
- Metoden `toString` skall returnera den information som framgår av programkörningen.

Här är mainmetoden som skall passa klassen `TicketMachine`:

```
public static void main(String[] arg) {
    int price = 210;
    int nrOfTickets = 20;
    TicketMachine t = new TicketMachine("Biljettautomat 1", price, nrOfTickets);
    System.out.println(t);
    System.out.println(t.getPrice());

    System.out.println(t.buy(5));
    System.out.println(t);
    System.out.println(t.buy(18));
    System.out.println(t);
}
```

Programkörning:

```
Biljettens id=Biljettautomat 1, pris=210, antal=20
210
true
Biljettens id=Biljettautomat 1, pris=210, antal=15
false
Biljettens id=Biljettautomat 1, pris=210, antal=15
```

A3.

Skriv en mainmetod som skapar 50 st biljettautomater med innehåll som nedanstående programkörning visar. Notera att varje automat laddas initialt med ett slumpmässigt antal biljetter som kan vara från 1-20. Efter skapandet av de 50 automaterna skall programmet köpa 10 st biljetter från varje maskin och räkna hur många automater som det gick bra att köpa från. Mainmetoden skall använda klassen `TicketMachine` och antag att metoden finns i en klass `TestTicketMachine`. Mainmetoden skall fungera som nedanstående programkörning visar:

50 st biljettautomater är skapade. Deras status är följande:

```
Biljettens id=Consert1, pris=240, antal=19
Biljettens id=Consert2, pris=240, antal=15
Biljettens id=Consert3, pris=240, antal=8
Biljettens id=Consert4, pris=240, antal=14
...
Biljettens id=Consert47, pris=240, antal=13
Biljettens id=Consert48, pris=240, antal=2
Biljettens id=Consert49, pris=240, antal=16
Biljettens id=Consert50, pris=240, antal=8
```

Notera: Utskrifter är utelämnade

Har försökt att köpa 10 st biljetter från varje biljettautomat.

Aktuell status efter dessa köp:

```
Biljettens id=Consert1, pris=240, antal=9
Biljettens id=Consert2, pris=240, antal=5
Biljettens id=Consert3, pris=240, antal=8
Biljettens id=Consert4, pris=240, antal=4
...
Biljettens id=Consert47, pris=240, antal=3
Biljettens id=Consert48, pris=240, antal=2
Biljettens id=Consert49, pris=240, antal=6
Biljettens id=Consert50, pris=240, antal=8
```

Notera: Utskrifter är utelämnade

Antal automater det gick att köpa från: 28

B (Endast för dem som siktar på betyget 4 eller 5)

B1.

I en datingförening har varje medlem en identitet och ett antal egenskaper, där varje sådan egenskap har ett värde, ett heltal 0-9. Genom att beräkna hur pass överensstämmande dessa värden är mellan två medlemmar kan man beräkna ett matchningsmått. Antag att antalet egenskaper alltid är 5 för varje medlem.

Ett exempel:

En medlem med identiteten A har följande egenskapsvärden: 9, 0, 0, 2, 3.

En medlem med identiteten B har följande egenskapsvärden: 9, 0, 0, 1, 6.

Matchningen beräknas genom att summera kvadraterna på skillnaderna mellan värdena för medlemmarna.

En matchning mellan A och B ger *matchningsvärdet* 10 eftersom:

$$(9-9)^2 + (0-0)^2 + (0-0)^2 + (2-1)^2 + (3-6)^2 = 0 + 0 + 0 + 1 + 9 = 10$$

Skriv en klass Member som representerar en medlem som har en identitet (*identity*) och 5 st egenskapsvärden (*properties*) som är heltal. Klassen skall fungera ihop med följande mainmetod och ge de utskrifter som programkörningen visar:

```
public static void main (String[] arg) {
    int [] propertiesA = {9,0,0,2,3};
    int [] propertiesB = {9,0,0,1,6};
    Member ma = new Member("A",propertiesA);
    Member mb = new Member("B",propertiesB);
    System.out.println(ma);
    System.out.println(mb);
    int match = ma.matchValue(mb);
    System.out.println("Matchningsvärdet mellan "
        + ma.getIdentity() + " och " + mb.getIdentity() + " : " + match);
}
```

Programkörning resulterar i följande utskrift:

```
<A 9 0 0 2 3 >
<B 9 0 0 1 6 >
Matchningsvärdet mellan A och B : 10
```

B2

Datingföreningen har delat in sina medlemmar i olika grupper. Matchningen görs för alla medlemmarna inom en grupp. Exempelvis om vi har följande (lilla) grupp med fyra medlemmars identitet och egenskapsvärden:

kim.nilsson@pollax.se	5	1	3	1	8
fia.jonsson@pollax.se	4	6	4	5	3
my.ehn@pollax.se	3	9	1	1	4
ida.lonn@pollax.se	5	7	3	8	2

I exemplet blir det 6 st par vars matchningsvärden beräknas: kim vs fia, kim vs my, kim vs ida, fia vs my, fia vs ida, my vs ida. Det lägsta värdet blir 13 vilket avser matchningsvärdet mellan paret fia och ida. I denna grupp har alltså detta par den bästa matchningen.

För att hantera olika grupper har man en klass `MemberGroup`. Nedanstående mainmetod (se nästa sida) gör en matchning för ovan exempel på grupp som namnges som `Pollax`. Den jämför alla par av medlemmar och beräknar det par som matchar bäst, dvs har det lägsta matchningsvärdet. Om det bästa matchningsvärdet är högst 20, tas det paret bort från gruppen.

```

public static void main (String[] arg ) {

    MemberGroup g = new MemberGroup("Pollax");

    Member m;
    int [] p1 = {5,1,3,1,8}; m = new Member("kim.nilsson@pollax.se",p1); g.add(m);
    int [] p2 = {4,6,4,5,3}; m = new Member("fia.jonsson@pollax.se",p2); g.add(m);
    int [] p3 = {3,9,1,1,4}; m = new Member("my.ehn@pollax.se",p3); g.add(m);
    int [] p4 = {5,7,3,8,2}; m = new Member("ida.lonn@pollax.se",p4); g.add(m);

    System.out.println("Antal medlemmar i gruppen " + g.getIdentity()+ ":" + g.getAntal() );
    System.out.println(g);

    System.out.println("Alla medlemmars matchningsvärden:");
    g.printMatchValues();

    // Beräkna det par som matchar bäst
    Member [] best = g.bestMatchPair();
    int matchValue = best[0].matchValue(best[1]);
    System.out.println("\nBästa match:" + best[0].getIdentity() + " och " +
        best[1].getIdentity() + ":Matchvärde = "+matchValue);

    // Ta bort bästa paret om matchningsvärdet är <=20
    if (matchValue<=20) {
        g.deleteMember(best[0]);
        g.deleteMember(best[1]);
        System.out.println(best[0].getIdentity()+ " och " + best[1].getIdentity() +
            " är bortagna från gruppen");
    }

    System.out.println("Antal medlemmar i gruppen " + g.getIdentity()+ ":" + g.getAntal() );
    System.out.println(g);
} // main

```

En programkörning ger följande resultat:

```

Antal medlemmar i gruppen Pollax:4
<kim.nilsson@pollax.se 5 1 3 1 8 >
<fia.jonsson@pollax.se 4 6 4 5 3 >
<my.ehn@pollax.se 3 9 1 1 4 >
<ida.lonn@pollax.se 5 7 3 8 2 >

Alla medlemmars matchningsvärden:
<kim.nilsson@pollax.se 5 1 3 1 8 > vs <fia.jonsson@pollax.se 4 6 4 5 3 >: matchVärde=68
<kim.nilsson@pollax.se 5 1 3 1 8 > vs <my.ehn@pollax.se 3 9 1 1 4 >: matchVärde=88
<kim.nilsson@pollax.se 5 1 3 1 8 > vs <ida.lonn@pollax.se 5 7 3 8 2 >: matchVärde=121
<fia.jonsson@pollax.se 4 6 4 5 3 > vs <my.ehn@pollax.se 3 9 1 1 4 >: matchVärde=36
<fia.jonsson@pollax.se 4 6 4 5 3 > vs <ida.lonn@pollax.se 5 7 3 8 2 >: matchVärde=13
<my.ehn@pollax.se 3 9 1 1 4 > vs <ida.lonn@pollax.se 5 7 3 8 2 >: matchVärde=65

Bästa match:fia.jonsson@pollax.se och ida.lonn@pollax.se:Matchvärde = 13
fia.jonsson@pollax.se och ida.lonn@pollax.se är bortagna från gruppen
Antal medlemmar i gruppen Pollax:2
<kim.nilsson@pollax.se 5 1 3 1 8 >
<my.ehn@pollax.se 3 9 1 1 4 >

```

Skriv klassen `MemberGroup` som passar ovan `main`-metod och ger det resultat som programkörningen visar. Givet är nedanstående skal (se nästa sida) till klassen `MemberGroup` där det saknas följande delar som du skall skriva:

- Konstruktorn
- Metoden `getAntal` som returnerar antal medlemmar i gruppen
- Metoden `add` med vilken man lägger till en medlem i gruppen
- Metoden `printMatchValues` som skriver ut alla par och deras matchningsvärden
- Metoden `bestMatchPair` som beräknar det par som matchar bäst. Metoden returnerar en array med två element som är de medlemmar som matchar bäst. Antag att det bara finns ETT par som matchar bäst, dvs bara ett par som har ett minsta matchningsvärde
- Metoden `deleteMember` som tar bort en medlem från gruppen. Metoden skall returnera `true` om det gick att utföra detta (dvs medlemmen existerade), annars skall `false` returneras.

```

import java.util.* ;

public class MemberGroup {
    // Instansvariablerna är givna
    private String identity;
    private ArrayList<Member> memberList;

    public MemberGroup... // Uppgift B2a
    ...

    public String getIdentity() {
        return this.identity;
    }

    public int getAntal... // Uppgift B2b
    ...

    public void add... // Uppgift B2c
    ...

    public void printMatchValues... // Uppgift B2d
    ...

    // Given metod
    public int searchMember(Member m) {
        for (int i=0; i<this.memberList.size(); i++) {
            if (m.getIdentity().equals(this.memberList.get(i).getIdentity())) {
                return i;
            }
        }
        return -1;
    }

    public Member[] bestMatchPair... // Uppgift B2e
    ...

    public boolean deleteMember... // Uppgift B2e
    ...

    // Given metod
    public String toString() {
        String s = "";
        for (int i=0; i<this.memberList.size(); i++) {
            s = s + this.memberList.get(i) + "\n";
        }
        return s;
    }
} // MemberGroup

```

På nästa sida följer en sida med hjälpmedel till tentamen...