

Tentamen OOP 2015-01-11

Anvisningar

Fråga 1 och 2 besvaras på det särskilt utdelade formuläret. Du får gärna skriva på bägge sidorna av svarsbladen, men påbörja varje uppgift på ett nytt blad. Vid inlämning ska svaren ligga sorterade efter frågornas nummer, inte efter den ordning du besvarat dem.

Betygsättning

Maxpoängen på tentan är 50.

Fx – Minst 20 poäng på tentan. Betyget kan också ges om man får minst 25 poäng, men mindre än 8 poäng sammanlagt på fråga 1 och 2.

E – Minst 8 poäng sammanlagt på fråga 1 och 2, samt minst 25 poäng totalt.

D – Minst 8 poäng sammanlagt på fråga 1 och 2, samt minst 30 poäng totalt.

C – Minst 8 poäng sammanlagt på fråga 1 och 2, minst 35 poäng totalt, samt ingen uppgift med 0 poäng.

B – Minst 40 poäng totalt, samt ingen uppgift med 0 poäng.

A – Minst 45 poäng totalt.

Hjälpmedel

Inga hjälpmedel förutom den syntaxsammanfattning som delas ut på tentan.

Lycka till!

Lösningförslag läggs upp i Moodle senast tre arbetsdagar efter tentatillfället.

Preliminär version utan
förklaringar till svaren

Fråga 1 – Flervalsfrågor (10 poäng)

Denna fråga består av 10 delfrågor av flervalstyp liknande de som funnits i testen i Moodle. På varje delfråga kan noll, ett, eller flera alternativ vara korrekta. Varje delfråga ger 1 poäng om samtliga korrekta alternativ är markerade och 0 poäng om detta inte är uppfyllt. Till skillnad från testen i Moodle kan man alltså inte få delpoäng.

A. Vilka av nedanstående programsatser kompilerera?

1. **Double d = null;**
2. `int i = null;`
3. **int index = "str".indexOf(null);** **Fungerar inte, men kompilerar gör den**
4. **String metod(){ return null; }**

B. Vad kommer följande metod att returnera om den anropas med summera(3)?

```
public int summera(int tal){
    if(tal == 1)
        return tal;
    else
        return tal + summera(tal - 1);
}
```

1. 1
2. 3
3. **6**
4. Ingenting, metoden kompilerar inte

C. Vad skrivs ut av följande kod?

```
for (int n = 0; n < 5; n++) {
    System.out.print(n % 3);
}
```

1. **01201**
2. 01230
3. 12012
4. 12301

D. Om du ska summera alla tal i en lista av heltal, vilken looptyp bör du normalt använda?

1. Den generella versionen av for: `for(init ; villkor; uppräkning){...}`
2. **Den specialiserade versionen av for: `for(typ var: lista){ ... }`**
Den är både enklare, och i vissa fall effektivare än 1
3. do-while
4. while

E. Vilka av följande programsatser är funktionellt identiska med `a=a-1`;

1. `--a;`
2. **`a--a;`**
3. `a=a--;`
4. `a-=a;`

F. Vilka variabler i samma klass kan nedanstående metod komma åt?

```
private void enMetod() { ... }
```

1. **Privata variabler**
2. **Publika variabler**
3. **Instansvariabler**
4. **Statiska variabler (klassvariabler)**

G. Vad blir resultatet av detta uttryck: `Math.max(Math.floor(2.6), Math.floor(3.1))`

1. 2 – dvs. en int
2. 2.0 – dvs. en double
3. 3 – dvs. en int
4. **3.0 – dvs. en double**

H. Antag att vi har följande variabler: `float f; double d;` Vilka av följande tilldelningar kompilerar?

1. `f=d;`
2. **`d=f;`**
3. **`f=(float)d;`**
4. **`d=(double)f;`**

I. Vad skrivs ut av nedanstående kod om variabeln `tal` är 2?

```
switch(tal) {  
  case 3:  
    System.out.print("Tre");  
  case 0:  
    System.out.print("Noll");  
  case 2:  
    System.out.print("Två");  
  case 1:  
    System.out.print("Ett");  
}
```

1. Två
2. **TvåEtt**
3. TvåTre
4. Inget, koden kompilerar inte

J. Nedanstående kod är tänkt att skriva ut DCBA, men fungerar inte när man kör den. Vilken del av koden är fel? **Obs! fel i formuleringen av frågan, "vilken" borde varit "vilka". Det räcker med ett korrekt alternativ för poäng.**

```
char[] arr = { 'A', 'B', 'C', 'D' };  
for (int index = 4; index >= 0; index--) {  
  System.out.println(arr[index]);  
}
```

1. `char[] arr = { 'A', 'B', 'C', 'D' };`
2. **`int index = 4`**
3. `index >= 0`
4. **`System.out.println(arr[index]);`**

Fråga 2 – Kodförståelse (10 poäng)

Om man exekverar följande programrader, vad kommer att skrivas ut på skärmen? Du skall i dina svar vara noga med vad som skrivs på vilken rad, alltså beakta skillnaden mellan print och println.

Delfråga A (5 poäng)

```
for (int a = 1; a < 4; a++) {  
    for (int b = 1; b < 4; b++) {  
        System.out.print(a * b + " ");  
        if (a % b == 2) {  
            System.out.println();  
        }  
    }  
}
```

Svar

```
1 2 3 2 4 6  
3 6 9
```

Delfråga B (5 poäng)

```
Cls[] arr = new Cls[4];
arr[0]=new Cls("TENTA");
arr[1]=new Cls(123);
arr[3]=new Cls("");
arr[2]=new Cls(4);

for (Cls c : arr) {
    System.out.println(c);
}
```

```
public class Cls {

    static final int IDX = 2;
    static int i = 4;
    char c = 'H';

    public Cls(int i) {
        this("" + i);
        Cls.i = i;
    }

    public Cls(String s) {
        if (s.length() > IDX) {
            c = s.charAt(IDX);
        }
    }

    public String toString() {
        return "" + c + i;
    }
}
```

Svar

N4
34
H4
H4

Fråga 3 (10 poäng)

I den serien Lunch (<http://www.lunchstriper.no/>) uppfinner Kjell, en av huvudpersonerna, vid ett tillfälle en "But-remover", ett par hörselskydd som upptäcker om den man talar med säger ordet "men" och då filtrerar bort resten av meningen. Din uppgift är nu att skriva en metod som fungerar som en förenklad version av denna uppfinning. Metoden ska ta en sträng som parameter och returnera en version av strängen där tecknen "men" och allting som kommer efter är borttaget. Du kan anta att strängen som skickas in är skriven med enbart små bokstäver.

In	Ut	Kommentar
"bra idé, men har du tänkt på ..."	"bra idé, "	Mellanslaget efter kommat finns kvar.
"bra idé!"	"bra idé!"	Inget men, så originalsträngen returneras.
"lommens läte ..."	"lom"	Du behöver inte tänkt på att "men" ska vara ett eget ord.

Lösningförslag

```
public static String taBortMenVersion1(String str) {
    int index = str.indexOf("men");

    if (index == -1) {
        // Inget men så vi returnerar strängen som den är
        return str;
    } else {
        String resultat = "";
        for(int i=0; i<index; i++){
            resultat+=str.charAt(i);
        }
        return resultat;
    }
}
```

Fråga 4 (10 poäng)

I många applikationer har man behov av buffrar av olika slag, ofta därför att de jobb som ska utföras kommer ojämnt fördelade så att man inte alltid kan ta hand om dem lika snabbt som de kommer in. När jobben kommer in placeras de i en kö och kan sedan plockas ut ur kön allteftersom man har tid att ta hand om dem. Normalt finns det också en övre gräns för hur mycket bufferten rymmer. Vad som händer om man försöker stoppa in mer i bufferten än vad det finns plats för varierar, men en strategi är att man helt enkelt kastar bort det som inte får plats.

Din uppgift är att skriva en klass som fungerar som en sådan buffert. Typen du ska buffra heter Jobb (eller Jobb om du föredrar svenska namn) och är i övrigt ointressant för uppgiften. Storleken på bufferten ska anges när man skapar ett objekt av klassen, men får inte vara mindre än 100. Om man försöker sätta storleken till mindre än 100 så ska den sättas automatiskt till 100.

- `boolean add(Jobb j)`
Tar emot ett Jobb-objekt, lägger det sist i kön och returnerar true. Om Jobb-objektet är null eller bufferten är full så händer ingenting förutom att metoden returnerar false.

- `Job get()`
Tar bort och returnerar den första Job-objektet från kön. Om bufferten är tom så returneras null.
- `int size()`
Returnerar det nuvarande antalet jobb i bufferten.

Lösningförslag

```
import java.util.ArrayList;

public class Buffer {

    private static final int DEFAULT_MAX_SIZE = 100;

    private ArrayList<Job> queue = new ArrayList<>();
    private int maxSize;

    public Buffer(int maxSize) {
        if (maxSize >= DEFAULT_MAX_SIZE) {
            this.maxSize = maxSize;
        } else {
            this.maxSize = DEFAULT_MAX_SIZE;
        }
    }

    public boolean add(Job j) {
        if (size() < maxSize && j != null) {
            queue.add(j);
            return true;
        } else {
            return false;
        }
    }

    public Job get() {
        if (size() > 0) {
            return queue.remove(0);
        } else {
            return null;
        }
    }

    public int size() {
        return queue.size();
    }
}
```

Fråga 5 (10 poäng)

På nästa sida finns ett klassdiagram över ett antal klasser som skulle kunna ingå i ett banksystem. Banken som använder systemet vill nu initiera en reklamkampanj riktad till områden där det redan bor rika kunder och det är din uppgift att ta reda på vilka områden man bör inrikta sig på. För att kunna lösa uppgiften har du tillgång till en ArrayList deklarerad som

```
ArrayList<Konto> konton = ...;
```

som innehåller alla bankens konton sorterade efter deras nuvarande saldo. Kontot med det högsta saldot står alltså först i listan, kontot med det näst högsta saldot på den andra platsen, och så vidare.

Din uppgift är nu att skriva den kod som krävs för att skriva ut namnen på de tio orter där reklamkampanjen ska genomföras. Valet av orter baseras på de konton som har högst saldon. Om varje kund bara hade ett konto och bodde på sin egen unika ort så vore detta inget problem, då vore det ju bara frågan om att skriva ut postorten för kontoinnehavarna för de tio första kontona i listan, men så är naturligtvis inte fallet. En kund kan ju ha flera konton, och flera kunder kan bo på samma postort, men banken vill ha ut tio unika orter.

Lösningförslag 1

```
ArrayList<String> orter=new ArrayList<>();

for(Kund k:kunder) {
    String möjligNyOrt = k.adress().postnummer().postort();
    if(orter.size()<10 && orter.indexOf(möjligNyOrt)==-1) {
        orter.add(möjligNyOrt);
    }
}

for(String ort: orter){
    System.out.println(ort);
}
```

Lösningförslag 2

```
ArrayList<String> orter=new ArrayList<>();

for(Kund k:kunder) {
    String möjligNyOrt = k.adress().postnummer().postort();
    if(orter.indexOf(möjligNyOrt)==-1) {
        orter.add(möjligNyOrt);
    }

    if(orter.size()==10)
        break;
}

for(String ort: orter){
    System.out.println(ort);
}
```


Lösningförslag 3

```
ArrayList<String> orter=new ArrayList<>();

int kundindex=0;
while(orter.size()<10 && kundindex<kunder.size()){
    String möjligNyOrt =
kunder.get(kundindex).adress().postnummer().postort();
    if(orter.indexOf(möjligNyOrt)==-1){
        orter.add(möjligNyOrt);
    }
}

for(String ort: orter){
    System.out.println(ort);
}
```

