

Lärare: Mats Svensson

---

TENTAMEN INFA15/SYSA01, delkurs i Mjukvaruutveckling

Tisdag 17 augusti 2010 kl. 08.00-13.00

Skrivningen omfattar totalt 100 poäng.

För Godkänt fordras minst 50 poäng och för Väl godkänt minst 80 poäng.

**Läs allt på denna sida innan du börjar skriva tentamen!**

Misstanke om fusk anmäls till rektor vid Lunds universitet. Student som fuskar riskerar att stängas av från studier under kortare eller längre tid. Man har inte rätt till studiemedel under avstängningsperioden.

**Anvisningar:**

Mobiltelefoner (avstängda!), väskor och dylikt placeras utom räckhåll under tentamen.  
Mp3-spelare e.d. får inte användas.

Fyll i tentamensomslaget fullständigt, enligt anvisningarna.  
Änge dessutom ditt folkbokföringsnummer på varje inlämnat blad.

Börja varje uppgift på nytt blad och skriv endast på en sida av pappret.  
Skriv tydligt! Oläsliga svar rättas inte.  
Skriv ej med rött.  
Sortera lösningarna i nummerordning.

Tentamensomslaget ska fyllas i och lämnas in även om du inte lämnar in några svar.

Skrivningslokalen får lämnas tidigast en timme efter skrivningens början.

Tillåtna hjälpmedel: Objects First with Java (Barnes & Kölling), engelsk ordbok

Obs. Anteckningar i böckerna är ej tillåtna. Understrykningar är tillåtna. Korsreferenser är tillåtna.

Tentamensvisning och utkvittering äger rum Tisdagen den 31 augusti kl 12.15 i sal 109.  
Därefter kan tentamen hämtas hos rättande lärare på dennes mottagningstider.

Eventuella klagomål ska lämnas inom tre veckor efter skrivningsvisning/resultatpublicering.  
Klagomål inkomna därefter beaktas ej.

Lycka till!

Lärare: Mats Svensson

Du får införa extra hjälpmetoder där du tycker att det är befogat.  
(Du får anropa de andra metoderna i uppgiften och använda det returnerade resultatet även om du inte implementerat dem eller inte implementerat dem rätt.)

### Uppgift 1 (5p)

Vad kommer att skrivas ut när följande Javakod körs?

```
int k,m;  
k = 1;  
m = k + 1;  
System.out.println("Nummer 1: k=" + k + " och" + " m=" + m);  
k = m + 2;  
m = k + 3;  
k++;  
System.out.println("Nummer 2: k=" + k + " och" + " m=" + m);  
k++;  
m++;  
System.out.println("Nummer 3: k=" + k + " och" + " m=" + m);
```

### Uppgift 2 (6p)

Vi har följande metod:

```
public boolean metod1(int a, int b) {  
    int c = a - b - 1;  
    System.out.println("a=" + a + " b=" + b + " c=" + c);  
    boolean d = (b!=c);  
    return d;  
}
```

Vad skrivs ut, samt vilket värde returneras av metoden, om följande metodanrop utförs (två fall):

- (a) metod1(8,7);
- (b) metod1(17,8);

### Uppgift 3 (7p)

Vi har följande metod:

```
public String metod2(String s1, String s2) {  
    System.out.println("Rad 1: " + s1 + " " + s2);  
    System.out.println("Rad 2: " + s2 + " " + s1);  
    String s3 = s1.substring(2)+ s2.substring(0,1);  
    System.out.println("Rad 3: " + s3);  
    String s4 = s2 + s3;  
    return s4.toLowerCase();  
}
```

Vad skrivs ut, och vilket värde returneras av metoden, om följande metodanrop utförs: metod2("Ettan", "Tvåan");

Lärare: Mats Svensson

#### Uppgift 4 (6p)

Vi har följande metod:

```
public void metod3(int x) {  
    int max = 2;  
    int[] a = {2, 3, 4};  
    for (int i=0; i<=max; i++) {  
        System.out.println("i= " + i + " ; (a[i])= " + (a[i]));  
    }  
    a[0] = a[0] + x;  
    a[1]++;  
    a[2]--;  
    for (int i=0; i<=2; i++) {  
        System.out.println("i=" + i + " ; (a[i])= " + a[i]);  
    }  
}
```

Vad skrivs ut när följande metodanrop utförs: metod3(5);

#### Uppgift 5 (10p)

Överst i klassen där följande metod är definierad står det

```
import java.util.ArrayList;
```

längre ner i klassens definition finns följande metod:

```
public void metod4 (String x, String y, String z) {  
    ArrayList<String> al = new ArrayList<String>();  
    al.add(z);  
    al.add(x);  
    al.add(y);  
    al.add(al.get(2)+x);  
    System.out.println("Size: "+al.size());  
    int i = 0;  
    for (String s : al) {  
        i++;  
        System.out.println(i+" "+s);  
    }  
    al.remove(1);  
    System.out.println("Size: "+al.size());  
}
```

Vad skrivs ut om följande metodanrop utförs: metod4("Ettan", "Tvåan", "Trean");

#### Uppgift 6 (12p)

Skriv en metod som tar två heltal som inparametrar (tal1 och tal2). Din metod skall jämföra dem och, beroende på resultatet av jämförelsen, skall den skriva ut den av följande tre meningar som beskriver förhållandet mellan dessa två heltal:

"Dessa två tal är lika."

"Tal 1 är större."

"Tal 2 är större."

Lärare: Mats Svensson

### Uppgift 7 (16p)

*Räkna antalet personer i en ArrayList som har ett givet efternamn:*

Antag att det finns redan en klass som heter *Person*. Denna klass innehåller, bland annat, en metod *surname()*, som returnerar en textsträng, personens efternamn. (Skriv inte denna åtkomstmetod, vi antar att den finns redan.) Skriv en metod som tar som inparametrar: (1) en referens till en ArrayList av objekt av typen *Person* (obs! ArrayList!), och (2) ett efternamn, av typen *String*. Din metod skall räkna antalet personer i arraylistan som har det aktuella efternamnet.

### Uppgift 8 (18p)

Antag att det finns redan en klass som heter *PostAdress*. Denna klass innehåller, bland annat, en metod *postNummer()*, som returnerar ett heltal av typ *int*, postnumret i adressen samt en metod *ort()*, som returnerar en sträng motsvarande ort till postnumret. (Skriv inte denna metod, vi antar att den finns redan.) Skriv en metod som tar som inparametrar: (1) en referens till en ArrayList av objekt av typen *PostAdress*, och (2) ett heltal av typ *int*. Din metod skall returnera en sträng som innehåller orten som postnummer överensstämmer med, heltalsparametern.

### Uppgift 9 (20p)

*Kopiera alla ogifta personer från en ArrayList av personer till en ny ArrayList:*

Skriv en metod som heter *kopieraOgifta*. Din metod skall ta som inparameter en ArrayList av typen *Person*. Vi antar att klassen (typen) *Person* är redan definierad, så du skall inte definiera den. Klassen *Person* har en åtkomstmetod som heter *isMarried()*, som returnerar det booleska värdet *true* om personen är gift, annars returnerar den värdet *false*. (Skriv inte denna åtkomstmetod, vi antar att den finns redan.) Din metod skall returnera en ny lista som innehåller endast de personer i inparameterlistan som är ogifta. Den ursprungliga listan får inte ändras.