



TEORETISKT DELPROV I DATAVETENSKAP PROGRAMMERING I JAVA, 5P

Uppgift (poäng)	Inlämnad	Poäng
1 (5)		
2 (2)		
3 (2)		
4 (6)		
5 (6)		
6 (4)		
7 (3)		
8 (4)		
Summa		

Kurs : Programmering i Java 5p. (TDBA54)

Datum : 020628

Namn (text) : _____

Personnr : _____

E-mail adress : _____
(@cs.umu.se)

Namnteckning

TEORETISKT DELPROV I DATAVETENSKAP

PROGRAMMERING I JAVA, 5P

Datum	:	020628
Tid	:	9-12
Hjälpmedel	:	inga
Antal uppgifter	:	8
Totalpoäng	:	32 (halva poängtalet krävs normalt för godkänt)

- Kryssa för de uppgifter du lämnar in.
- Lösningarna skall vara snyggt och prydligt nedskrivna. Tankegången skall vara lätt att följa. Alla antaganden som inte är uppenbara skall redovisas.

Tips!

- Försök på alla uppgifter!
- Om du inte kan lösa *en* av flera deluppgifter, gå vidare till övriga som om du löst den (med nödvändiga antaganden angivna).
- Uppgifter kan vara felformulerade, fråga om du är osäker eller tycker att något verkar konstigt.
- Det är ditt ansvar att övertyga om att du besitter den kunskap som efterfrågas.
- *Det är viktigt att du löser den givna uppgiften!*

Lycka till!

Uppgift 1 (5 poäng)

Redovisa hur det understrukna uttrycket evalueras och vilket värde det har :

Ingående variablers värden ges omedelbart före uttrycket

a)	<pre>int x = 1, y = 2; int z = <u>5-4*y+x/2</u>;</pre>
b)	<pre>int x = 1, y = 2; boolean ok = <u>x >= 0 && y > 2</u>;</pre>
c)	<pre>double w = 1, y = 3.2, z = 12.2; double res = <u>z/y/w</u>;</pre>

d) Evaluera nedanstående boolska uttryck för varje kombination av värden (T=true och F=false) på a, b och c

$$a \ || \ b \ \&\& \ c$$

a	b	c	
T	T	T	
T	T	F	
T	F	T	
T	F	F	
F	T	T	
F	T	F	
F	F	T	
F	F	F	

Uppgift 2 (2 poäng)

Förklara kort begreppet undantag (*exception*) resp. signatur i Java.

Undantag :

Signatur :

Uppgift 3 (2 poäng)

Beskriv effekterna som modifieraren har för:

a) en **metod** som är deklarerad *public*

b) en **metod** som är deklarerad *private*

c) ett **attribut** som är deklarerat *static*

d) en **metod** som är deklarerad *static*

Uppgift 4 (6 poäng)

Spåra följande loopar så att det klart framgår hur de ingående variablerna ändrar värde

```
1.  int j,sum=0;

    for (int i = 2; i < 24; i = i+2)
    {
        j = 24-i;
        while (j > i)
        {
            sum = sum + i;
            j = j - 2*i;
        }
    }

} // for
```

2. **Anrop av metoden:** `loopa(8, 128, 10);`

```
public void loopa(int start, int stop, int breakVal)
{
    int mult, count = 0;

    for (mult = start; mult <= stop; mult = mult+start)
    {
        System.out.print (mult + " ");
        count++;
        if (count % breakVal == 0)
            System.out.println();
    } // for

} // loopa
```

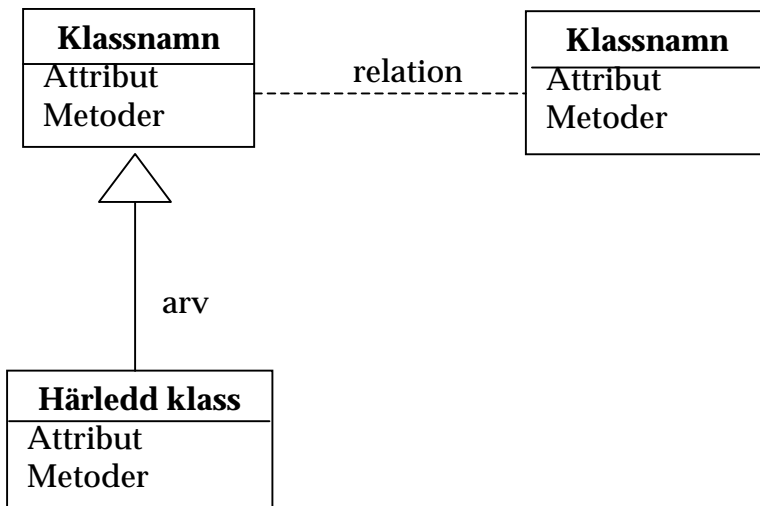
Anmärkning: `System.out.print()` skriver ut en sträng utan att göra ett efterföljande radbyte, medan `System.out.println()` genererar ett radbyte efter utskriften

Uppgift 5 (6 poäng)

Problembeskrivning:

På ditt nya jobb som kylskåpsdesigner/testare så får du i uppdrag att utarbeta ett system som simulerar ett kylskåp och olika typer av innehåll och hur de klarar sig beroende på temperaturen. Intressant är att se om gurkor tål högre temperatur än andra grönsaker, speciellt tomater. Fisk och korv har kunderna också uttryckt intresse av, och de vill också ha större hyllor och en lättare dörr, något som måste tas med i beräkningen över hur mycket ström kylskåpet använder för att hålla en viss temperatur.

Gör en analys, dvs ta fram tänkbara klasser och beroenden och rita ett klassdiagram över resultatet. I diagrammet skall ingå minst fem klasser, ett arv och en relation, t.ex 'använder', 'har', 'består av'. Det finns inga krav på vilka klasser du beskriver annat än att de skall ha med problembeskrivningen att göra. Varje klass skall ha minst ett attribut och en metod. Använd UML-notation enligt nedan:

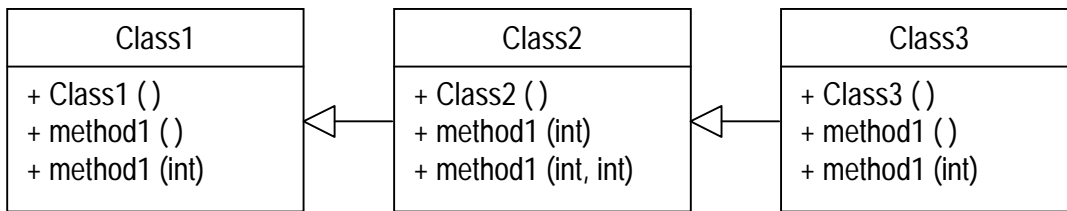


Svar lämnas nedanför och på efterföljande sida

Svar uppgift 5

Uppgift 6 (4 poäng)

Utgå från följande situation



och följande deklarerationer:

```

Class1 c1 = new Class1();
Class2 c2 = new Class2();
Class3 c3 = new Class3();
    
```

Vilken *version* av `method1` anropas efter följande satser (skriv e/t om kombinationen ej är tillåten). Motivera dina svar!

	Klass	Motivering
<code>c1.method1(1);</code>		
<code>c1.method1(1, 2);</code>		
<code>c1 = c2;</code>		
<code>c1.method1();</code>		
<code>c1.method1(1);</code>		
<code>c1.method1(2, 1);</code>		
<code>c1 = c3;</code>		
<code>c1.method1();</code>		
<code>c1.method1(1);</code>		
<code>c1.method1(2, 3);</code>		

Uppgift 7 (3 poäng)

Ge exempel och förklara skillnaden mellan formella och aktuella parametrar

Uppgift 8 (4 poäng)

Anta att koden nedan är given för klasserna `Person` och `Anstalld`. Fyll i koden som saknas för att implementera konstruktorena i `Anstalld` klassen. Du får själv bestämma vad som skall vara standardvärden.

Tips: Använd `this` och `super`.

```
public class Person
{
    private String name;
    private int yearOfBirth;

    /**
     * Instantiera ett Person objekt med namn n och födelseår y
     */
    Person (String n, int y)
    {
        name = n;
        yearOfBirth = y;
    }
}

public class Anstalld extends Person
{
    private String room;

    /**
     * Instantiera ett Anstalld objekt med namn n, födelseår y
     * och rumsnummer room
     */
    Anstalld (String n, int y, String room)
    {
        // här saknas kod
    }

    /**
     * Instantiera ett Anstalld objekt med standardvärden för
     * namn, födelseår och rumsnummer.
     */
    Anstalld ()
    {
        // här saknas kod, använd dig av konstruktorn ovan
    }
}
```

Svar lämnas på efterföljande sida (skriv kompletta konstruktorer)

Svar uppgift 8