

Tentamen i

Grundläggande programmering

STS, åk 1

2003-08-21

Skrivtid: 14.00 - 19.00

Hjälpmedel: Inga

Lärare: Anders Berglund. Anders besöker tentan cirka kl. 15.30.

Observera:

- Programkod ska skrivas enligt god programmeringssed för Java och objektorienterad programmering.
- Uppgifterna är inte ordnade efter svårighetsgrad.
- Läs igenom hela tentan först.
- Kom ihåg att fylla i försättsbladet ordentligt.
- Om du är osäker på någon fråga eller uppfattar någon uppgift som oklar, gör ett antagande som du motiverar och förklarar väl.
- På de uppgifter där du ska skriva programkod, kan du också skriva en förklaring, om du inte är säker på att din lösning är riktig. En sådan ger visserligen inga poäng, men kan hjälpa till vid rättningen att förstå tankarna bakom en lösning
- Skriv programkod rimligt indenterad och med bra variabelnamn. På så sätt underlättas rättningen.
- 50% av tentans totala poängantal garanterar betyget godkänt
- Skriv endast en uppgift per blad och skriv bara på ena sidan av papperet, dvs. ett nytt blad för varje uppgiftsnummer, medan deluppgifter (a, b, c etc.) får skrivas på samma blad.
- Fyll i försättsbladet ordentligt!
- Skriv namn på alla papper.
- Referera inte mellan olika svar.

Lycka till!

FÖRSÄTTSBLAD

Grundläggande programmering, STS, 21 aug 2003

Denna sida ska ifyllt placeras först bland de papper du lämnar in

Uppgift	Besvarad ange i förekommande fall vilka deluppgifter du har löst
1	
2	
3	
4	
5	
6	
7	
8	
9	
bonus	<i>ifylles av institutionen</i>

Namnteckning: _____

Namn: _____

Personnummer: _____

1. *Ord, bara ord (0,5 + 0,5 + 1 + 1)*

Förklara kortfattat men precist följande termer:

- syntax (i samband med programmeringsspråk)
- semantik (i samband med programmeringsspråk)
- Applet (i samband med Java)
- inkapsling (i samband med objektorienterad programmering)

2. *Video (6 + 2 + 4 + 4)*

Du extraknacker i en videobutik (studiemedlet räcker ju inte så långt ...) och får en dag en annan uppgift än att sälja filmer. Du ska skriva ett Java-program för att hantera filmer och filmlagret.

- Skriv en klass **Film**, som kan användas för att lagra data om en film. Klassen ska innehålla information om titel, huvudrollsinnehavare (en person), regissör, hyrespris, samt om filmen är uthyrd eller inte (sant eller falskt). Du ska själv avgöra vilka typer de variabler du väljer bör ha. Klassen ska dessutom ha:
 - en relevant konstruktor som läser in data från tangentbordet
 - en utskriftsmetod, som på ett enkelt sätt men tydligt sätt skriver ut all data om en viss film, samt
 - en metod som returnerar falskt eller sant beroende på om filmen är uthyrd eller ej.

Även om du anser att videobutiken skulle ha behov av ytterligare metoder, behöver du inte skriva dessa.

- Skriv ett litet huvudprogram (det får gärna vara mycket enkelt, exempelvis helt utan förklarande utskrifter) som visar att ditt program fungerar.
- Skriv klassen **Filmlager**, som håller reda på videobutikens samtliga filmer. Klassen ska innehålla två instansvariabler: en array av filmer, samt en variabel som anger antalet filmer i lagret. Klassen ska innehålla en konstruktor (som du kan göra så enkel som möjligt), samt en metod som lägger in en ny film i lagret. Parameter till denna metod ska vara ett objekt ur klassen **Film**. I denna deluppgift behöver du varken skriva ytterligare metoder (som skulle kunna behövas för videobutikens funktion) eller något huvudprogram.
- Utöka klassen **Filmlager** från föregående deluppgift med en metod som skriver ut all data på skärmen om samtliga filmer som finns inne (dvs. ej är uthyrda). Metoden ska på ett relevant sätt använda dina delresultat från den första deluppgiften.

3. *Klassvariabel (4 + 2)*

- Skriv ett utdrag ur ett program, som på ett meningsfullt sätt använder klassvariabler. Du avgör själv vad programmet ”ska handla om”. Programmet du ger behöver inte vara fullständigt, men det ska tydligt illustrera begreppet klassvariabler och visa på en meningsfull användning av det. Det räcker med att du skriver så många rader i programmet (som kan

bestå av olika metoder eller klasser, enligt ditt val), så att det är tydligt att, och på vilket sätt, klassvariabler illustreras.

Ett råd: Undvik svårigheter genom att välja ett enkelt, men tydligt exempel.

- b) Under kursen har påpekats, att klassvariabler endast bör användas i vissa sammanhang, och att det finns skäl att undvika den sortens konstruktioner i andra situationer. Varför är en alltför flitig användning av klassvariabler olämplig?

4. *1X2 (6)*

En stryktipsrad består av 13 tecken, där tecknet är något av tecknen 1, X eller 2. För att datorisera din kontroll av tipsrader vill du nu skriva en kontrollklass som kan undersöka hur många rätt du har på dina tipsrader.

Din uppgift är att skapa en klass `Tips` som beskriver veckans korrekta tipsrad. Du ska sedan kunna ge dina rader (en i taget) till ett objekt av denna klass och få reda på hur många rätt du hade. Din klass skall ha rätt rad som en instansvariabel, det ska t. ex. vara en array av tecken eller en sträng. Klassen ska ha en konstruktör som kan användas för att ange veckans rätta rad och en metod som givet din rad, returnerar antalet rätt som du hade. En main-metod kan se ut på detta sätt:

```
public static void main(String [] args) {  
    Tips t = new Tips( rätt rad här ); // skapa ett objekt med rätt rad  
    int r = t.kolla( min rad här ); // kolla min rad. Hur många rätt?  
}
```

Du skall skriva hela klassen och också ha en main-metod (som kan likna den main som finns ovan i uppgiftstexten) som visar att din klass fungerar. Huvudprogrammet behöver inte ha texter som hjälper användaren, vara genomtänkt etc., utan är enbart tänkt att kunna fungera som ett provkörningsexempel.

5. *OOP (6)*

Förklara kortfattat vad som menas med objektorienterad programmering (OOP). Din förklaring ska bl a beröra varför OOP används.

6. *Klasser (1 + 2)*

- a) Klasser, kan, som du vet ordnas hierarkiskt. Vad brukar den relation mellan två klasser kallas som hierarkin illustrerar?
- e) Det finns ytterligare relationer mellan klasser och/eller objekt som är relevanta i samband med objektorienterad programmering. Nämn två väsentliga sådana relationer, samt ge ett relevant exempel på varje. Du behöver inte skriva någon kod på denna uppgift, det är tillfyllest om du förklarar ditt exempel i ord.

7. *for* och *while* (2)

Vad är skillnaden mellan en *for*- och en *while*-loop? När kan det vara fördelaktigt att använda den ena och när är det fördelaktigt att använda den andra? Illustrera ditt resonemang med ett konkret exempel av vardera loopen. Du behöver inte skriva fullständiga program, men de programrader du visar (som ska vara fullständiga loopar) ska vara relevanta exempel.

8. *Integritet* (2)

Beskriv en situation där det vid en IT-användning sker en integritetskränkning. Vilket är det moraliska problemet? Hur kan kränkningen undvikas?

9. *Rekursion* (2 + 2)

Studera nedanstående program.

a) Ange exakt vad utskriften blir, när programmet körs.

```
import extra.*;

public class GorVadKlass {
    public static int gorVad(int x) {
        if (x == 0)
            return 0;
        if (x % 2 == 0)
            return x + gorVad(x - 1);
        return -x + gorVad(x - 1);
    }

    public static void main(String[] args) {
        Std.out.println(gorVad(3) + gorVad(5));
    }
}
```

b) Förklara i ord vad `int gorVad(int x)` beräknar.

Ledning: `%` (procenttecken) i programmet ovan anger rest vid heltalsdivision.