Tentamen i
Grundläggande programmering
STS, åk 1
2004-05-24

Skrivtid: 14.00 - 19.00

Hjälpmedel: Inga

Lärare: Anders Berglund. Anders besöker tentan cirka kl. 15.00 samt cirka kl 16.45.

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, 24 maj 2004

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

Uppgift	Besvarad. Ange i förekommande fall vilka deluppgifter du har löst	Poäng (ifylles av institutionen)
1		
2		
3		
4		
5		
6		
7		
8		
bonus	*ifylles av institutionen*	

Namnteckning: _____

Namn: _____

Personnummer:_____

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

Förklara kortfattat men precist följande termer:
a) syntax (i samband med programmeringsspråk)
b) semantik (i samband med programmeringsspråk)

2. Mer ord (2 + 2+ 1)

Förklara termerna *klass*, *objekt* och *arv* (2 + 2 + 1 p respektive). Det ska tydligt framgå vad skillnader och eventuella likheter mellan de olika termerna är. Ge gärna flera olika förklaringar till de två första termerna. Ledning: Exempelvis kan stol kan förklaras på många sätt (Här tre olika förklaringar): Någonting att sitta på. En del av ett möblemang. Någonting som är gjort av trä.

3. Ännu mera ord (2)

Beskriv skillnaden mellan instansvariabler och klassvariabler. Ge gärna ett kortfattat exempel, antingen i ord eller i Javakod som illustrerar användningen av de två begreppen.

4. Näststörst (6)

Studera nedanstående program:

```
import extra.*;

public class TabellKlass {
   private int[] minTabell = new int[100];

   public void in() {
      Std.out.println("Ge ett antal tal. Avsluta med 0.");
      int tp = Std.in.readInt();
      int index = 0;
      while (tp !=0) {
             minTabell[index] = tp;
             index++;
             Std.out.println("Ge nästa tal");
             tp = Std.in.readInt();
             Std.in.skipLine();
      }
   }

   public void skriv() {
      for (int i = 0; minTabell[i] != 0; i++)
             Std.out.print(minTabell[i] + " ");
   }
```

```
   public int näststörst() {
   . . .
   }

   public static void main(String[] args) {
      TabellKlass mittObjekt = new TabellKlass();
      Std.out.println("Ge minst tre olika tal. Avsluta med 0.");
      mittObjekt.in();
      Std.out.println("Talet " + mittObjekt.näststörst() +
           " är näst störst");
   }
}
```

En körning av programmet kan se ut på följande sätt:

```
Ge minst tre olika tal. Avsluta med 0.
3 4 6 2 0
Talet 4 är näst störst.
```

Din uppgift är nu att skriva metoden `int näststörst()`, som när den anropas returnerar det näst största talet i arrayen. Du får anta att användaren matar in tal till arrayen så att den innehåller minst tre positiva tal (ej 0), att alla tal är olika samt att 0 är lagrat sist (0 ”räknas” inte blad de tre talen).

Du får om du vill skriva ytterligare metoder eller lägga till fler variabler. Dock får du inte förändra den array som är lagrad i `mittObjekt`. Det sistnämnda villkoret innebär att flera olika anrop av `skriv()`, oavsett var i programmet de sker, sak ge samma resultat.

Ledning
För att hitta det näst största talet kan det vara bra att känna till det största.

5. Idrott (6)

I en möjlig idrott, konstskidning (som påminner om konståkning på skridskor), finns det i varje tävling både tävlingsdeltagare och domare. Domarna kan ge mellan 1 och 5 poäng och den tävlingsdeltagande som får högst poäng i snitt vinner. Vid lika antal poäng avgör lotten. För tävlingen finns det en resultattabell, som för varje tävlingsdeltagare innehåller medelvärdet av domarpoängen.

Ge förslag på tre klasser som du uppfattar som rimliga, representativa och bra för ett program som kan hantera en konstskidningstävling. För varje klass ska du ange två viktiga instansvariabler och tre viktiga metoder.

Du behöver inte koda klasserna. Formen för ditt svar (såsom Java eller svenska, tabeller eller löptext) avgör du själv. Det är en fördel, men inte ett krav, att du skriver förklaringar. Sådana ger i sig inga poäng men kan hjälpa dig om ditt svar i övrigt inrymmer oklarheter.

6. Annas Pepparkakor (6)

Som den kakälskare (kakmonster? ☺) du är har du sökt och fått ditt första jobb på Annas Pepparkakor. Efter ett tag inser du att jobbet som kvalitetskontrollant (= provsmakare) inte är nyttigt i längden. Du talar med din chef och får en ny uppgift som bygger på dina Java-kunskaper från din STS-utbildning. Du får löneförhöjning och får fria pepparkakor att ta med hem.

Din uppgift på arbetet är att ange en klasstruktur för pepparkakor som ska användas i pepparkakshus som säljs i byggsats. Delarna i huset kan var fyrkantiga (för tak och långsidor) eller femkantiga (för gavlar). Byggsatsen är enkel (nästa år får du arbeta med den avancerade byggsatsen lovade din chef över kaffe och en pepparkaka) och saknar därför skorsten, dörrar etc.

Varje del kan med andra ord representeras av fyra resp fem hörn. Hörnen är, hade du tänkt dig, angivna som punkter i ett plan och representerade av två koordinater (tal med decimaler).

Din tentauppgift är nu att skriva de klasshuvuden (i Java) för de klasser du anser behövas. För poäng på uppgiften krävs en väl vald klasstruktur där arv, samt förmodligen även någon annan relation mellan klasser/objekt används. Du ska i klasshuvudet ange klassens namn, viktiga instansvariabler samt, när behovet är påkallat, formulera samband mellan klasser/objekt i Java. Du behöver dock inte ange några metoder, varken till namn eller på något annat sätt.

7. Klockan går (8+4+4)

Studera klassen `Tid` nedan.

```
class Tid {
   private int värde;
   private int max;

   public Tid(int inMax) {
      värde = 0;
      max = inMax;
   }

   public void öka(){
      här saknas kod
   }

   public int geVärde(){
      här saknas kod
   }

   public int sättVärde(int nyttVärde) {
      värde = nyttvärde;
   }
}
```

a) Skriv en klass `Klocka` som ska innehålla timme, minut och sekund. Dessa ska på ett meningsfyllt sätt representeras av objekt ur klassen `Tid`.

   Du ska skriva en konstruktor för klassen och en metod `void tick()` som tickar fram klockan en sekund. `tick` ska hantera klockan på ett korrekt sätt, så att ett klick från exempelvis 3.24.59 blir 3.25.00 och från 8.59.59 blir 9.0.0.

Skriv också metoderna `öka()` samt `geVärde()` i `Tid` så att `dessa fungerar med tick()` såsom avsett.

b) Skriv ett huvudprogram i en separat klass som skapar två klockor, en med svensk tid och en med finsk. Du får göra programmet så att den svenska klockan alltid startar på 2.00.00. Finsk tid är en timme före den svenska. Programmet ska sedan ticka fram de båda klockorna 62 minuter.

c) Lägg till metoder för en layoutmässigt *enkel* utskrift. Även om själva utskriften får bli ful, ska utskriftsmetoderna vara en del av klasserna på ett relevant sätt. Lägg till sist till en utskrift av den svenska klockan och den finska klockan i huvudprogrammet efter framtickandet.

8. Grafiska frågor (1+1)

a) Ordna klasserna `Container`, `Window`, `JFrame` i den ordning de har i klasshierkien i Java.

b) Förklara kortfattat med precist skillnaden mellan `Panel` och `Jpanel`.