

OBJEKTORIENTERAD PROGRAMVARUUTVECKLING
för IT1 (TDA545)

TID 14.00-18.00

Ansvarig: Jan Skansholm, tel 7721012 eller 0707-163230

Resultat: Anslås senast 2006-11-17

Betygsgränser: Sammanlagt maximalt 60 poäng.
På tentamen ges graderade betyg:
3:a 24 poäng, 4:a 36 poäng och 5:a 48 poäng

Hjälpmedel: Skansholm, *Java direkt med Swing*, valfri upplaga, Studentlitteratur

Inga kalkylatorer är tillåtna.

Tänk på:

- att skriva tydligt och disponera papperet på ett lämpligt sätt.
- att börja varje ny uppgift på nytt blad. Skriv endast på en sida av papperet
- Skriv ditt personnummer på *alla* blad.
- De råd och anvisningar som givits under kursen skall följas vid programkonstruktionerna. Det innebär bl.a. att onödigt komplicerade, långa och/eller ostrukturerade lösningar i värsta fall ej bedöms.

Uppgift 1) Som du vet kan man inte ändra ett tecken i ett `String`-objekt. Därför skall du här göra en egen klass `OurString` där detta är möjligt. (Att det redan finns en annan färdig klass i Java för detta får du givetvis inte utnyttja). Klassen skall internt representera texten med ett fält (array) av tecken. Klassen skall ha en konstruktor med en text (av typen `String`) som parameter. Konstruktorn skall initiera det interna fältet. Dessutom skall klassen ha instansmetoderna:

- `int antal()` Ger antalet tecken i strängen.
- `void byt(int i, char c)` Ersätter det `i`:te tecknet med `c`. Om `i` ligger utanför gränserna för strängen skall en exception av standardklassen `IndexOutOfBoundsException` genereras.
- `void utökaMed(String s)` Lägger in tecknen i `s` efter dem som redan finns i texten. Detta kräver att du bildar ett längre fält.
- `String toString()` Returnerar innehållet i objektet som en `String`.

(12 p)

Uppgift 2) Ett primtal är ett tal som endast är jämnt delbart med 1 och med sig själv. En teknik att finna primtal går under namnet *Eratothenes säll*. Den går till på följande sätt: Man använder ett fält där elementen har typen `boolean`. Från början sätts element nummer 0 och 1 till `false` och alla övriga element i fältet till `true`. Därefter löper man igenom fältet med början på index 2. När man stöter på ett element (låt oss säga att det har nummer `i`) som har värdet `true` gör man följande. Löp igenom resten av fältet (fr.o.m. `i+1` till fältets slut) och sätt varje element som har den egenskapen att dess index är en jämn multipel av `i` till `false`. När man t.ex. kommer till element nr. 5 så skall elementen 10, 15, 20 etc. i fältet nollställas.

När denna process är avslutad visar de element i fältet som fortfarande har värdet `true` vilka index som är primtal. Skriv ett program som använder denna teknik för att beräkna och skriva ut alla primtal som är mindre än eller lika med 1000.

(10 p)

Uppgift 3) Man har i en textfil samlat uppgifter om ett antal personer. För varje person finns två rader i filen. På första raden står personens namn och adress och på andra raden finns personens ålder, längd och vikt. Längden anges i cm och vikten i kg. Man vill göra en medicinsk studie av överviktiga personer och söker därför personer vilkas s.k. body mass index (förkortat BMI) överstiger värdet 30. BMI beräknas enligt formeln m/h^2 där m är vikten i kg och h längden i m. Skriv ett program som läser filen med personuppgifter. Programmet skall skapa en ny textfil som bara innehåller uppgifterna för de personer vilkas BMI överstiger 30. Filernas namn skall läsas in från kommandoraden.

(10 p)

Uppgift 4) Antag att klassen `Flight` är definierad som på sidan 135 (117) i kursboken och att klassen `Tidpunkt` är definierad som på sidorna 68 (64) och 84 (80). (Sidorna inom parentes gäller upplagorna 3 och 4.) Antag vidare att `Tidpunkt` är utökad så att `Tidpunkt`-objekt är naturligt jämförbara och att jämförelserna sker i tidsordning. Utöka klassen `Flight` så att objekt av denna typ blir naturligt jämförbara. Jämförelserna skall ske i tidsordning avseende avgångstiden. Om två fligheter har samma avgångstid skall de hamna i bokstavsordning avseende namnet på destinationen. (För enkelhets skull kan du bortse från att det finns flightnummer.) Du behöver inte skriva om hela klassen `Flight`. Visa bara tydligt dina tillägg.

(4 p)

Uppgift 5) a) Konstruera en klass `Tärning` som skall beskriva en vanlig tärning som man använder t.ex. när man spelar Fia. Följande metoder skall finnas: `kasta`, som ger tärningen ett slumpmässigt värde mellan 1 och 6, `avläsVärde` som avläser tärningens värde och ger det som resultat samt `sättVärde` som har en parameter vilken anger vilket sida på tärningen man vill skall vara upp (parametern måste ha ett värde mellan 1 och 6, skulle så inte vara fallet skall programmet ge en felutskrift och avslutas). Vidare skall det finnas två konstruktorer, en som initierar tärningen till ett slumpmässigt värde och en som gör det möjligt att ge ett visst första värde till tärningen.

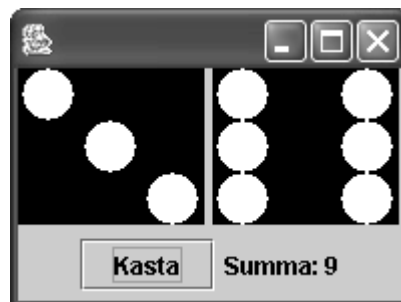
(7 p)

b) Utöka klassen `Tärning` så att objekt av denna klass blir grafiska komponenter som kan visas i fönster. En tärning skall ritas upp med *vita prickar på svart bakgrund*. (Hur tärningar skall ritas upp ser du i figuren till uppgift 6, där två tärningar visas.) Tips: Prickarna kan ritas som separata grafiska komponenter. Om du vill kan du använda dig av följande färdigskrivna klass:

```
import java.awt.*;
import javax.swing.*;
public class Prick extends JPanel {
    public void paintComponent(Graphics g) {
        super.paintComponent(g);
        Insets i = getInsets();
        int w = getWidth()-i.left-i.right;
        int h = getHeight()-i.top-i.bottom;
        int diam = Math.min(h,w);
        int x = i.left + (w-diam)/2;
        int y = i.top + (h-diam)/2;
        g.fillOval(x, y, diam, diam);
    }
}
```

(7 p)

Uppgift 6) Skriv ett program som visar två tärningar så som demonstreras i nedanstående figur. När man trycker på knappen `Kasta` skall båda tärningarna kastas så att de får slumpmässiga värden. Dessutom skall summan av värdena visas. Du bör naturligtvis använda dig av klassen `Tärning` från uppgift 5 (även om du inte löst den uppgiften)



(10 p)