



Høgskolen i Telemark

Avdeling for allmenne fag

EKSAMEN

I

801 OBJEKTBASERT PROGRAMMERING

04.01.02

Tid: 5 timer

Målform: Bokmål

Sidetall: 4 + denne framsida

Hjelpemidler: Alt skriftlig materiale

Merknader: Ingen

Vedlegg: Ingen

Eksamensresultatene blir offentliggjort på følgende internettadresse:

<http://www-bo.hit.no/af/>



Oppgavesettet inneholder tre uavhengige deler. Prosentsatsen ut for hver oppgave antyder vekt ved sensur. Les oppgaveteksten nøye, og prøv å svare på så mange spørsmål som mulig. Lykke til!

Oppgave 1 (30%)

Denne oppgaven tar utgangspunkt i boolske matriser, det vil si 2-dimensjonale tabeller der det i hver celle er lagret en boolsk verdi (true eller false). Følgende er et eksempel på en boolsk matrise med 3 kolonner og 4 rader:

true	false	false
false	true	true
false	false	false
true	true	false

Større matriser (for eksempel 600×800) kan brukes for å representere kart med to farger. Hvert element kan tegnes som ett piksel (punkt) på skjermen, der true og false svarer til ulike farger.

1-a)

Lag metoden *tegnKart* for å vise en matrise på skjermen.

```
public void tegnKart(Graphics g, boolean[][] k)
```

Parameteren *k* representerer kartet. Uttegning av kartet skal starte i øverste venstre hjørne av *g* (*x*=0 og *y*=0). Celler som inneholder *true* skal være hvite (*Color.white*), og celler som inneholder *false* skal være sorte (*Color.black*). For å tegne ett piksel med en bestemt farge i en bestemt posisjon på skjermen kan du bruke følgende hjelpemetode (skal ikke programmeres):

```
public void tegnPiksel(Graphics g, int x, int y, Color f)
```

Tips: Du finner antall rader i en matrise *k* ved *k.length* og antall kolonner ved *k[0].length*.

1-b)

I denne oppgaven skal du ikke skrive Java-kode.

Hvis matriser inneholder store "områder" av kun én verdi kan man representere verdiene mer kompakt som tekst. En strategi er å lese matrisene rad for rad og lagre sammenhengende sekvenser av true og false som henholdsvis positive og negative heltall. Vi må også ta vare på antall rader og kolonner i matrisen.

(Oppgave 1-b fortsetter på neste side)



Eksempelmatriksen over kan representeres slik:

```
"3;4;1;-3;2;-3;2;-1"
```

Tekstkonstanten skal tolkes slik: Matriksen består av 3 kolonner og 4 rader. Leser vi matriksen fra øverste venstre hjørne, rad for rad, ser vi 1 forekomst av true, 3 forekomster av false, 2 forekomster av true, 3 forekomster av false, 2 forekomster av true og til slutt 1 forekomst av false.

Foreslå en metode for å konvertere en boolsk matrise til tekstrepresentasjonen slik som beskrevet over. Angi navn på metoden, parametre og returverdier. Forklar kort hvordan metoden kan kodes, og spesielt hva slags hjelpevariable som trengs for konverteringen.

1-c)

Ut fra tekstrepresentasjonen av et kart, som definert i oppgave 1-b, kan vi regne ut hvor stor del av kartet som utgjøres av celler med verdi true. Metoden *prosentTrue* skal returnere dette i prosent:

```
public float prosentTrue(String k)
```

For eksempelmatriksen inneholder 5 av 12 celler verdien true, det vil si 41.7%.

Oppgave 2 (40%)

I denne oppgaven skal du lage en applett for å trene skoleelever i addisjon og subtraksjon. Brukergrensesnittet inneholde to knapper *btnPluss* og *btnMinus* merket med henholdsvis "+" og "-", en tekstboks *txtUt* for å vise utdata fra programmet og en tekstboks *txtInn* for å lese inn svar fra brukeren. Programmet skal ha følgende funksjonalitet:

- Når bruker klikker i en av knappene *btnPluss* og *btnMinus* skal programmet generere et tilfeldig regnestykke med tilhørende operator, og vise dette i tekstboksen *txtUt*.
- Brukeren svarer på et regnestykke i tekstboksen *txtInn*. Når bruker trykker <enter> her skal programmet gi melding "Riktig" eller "Galt" i tekstboksen *txtUt*.

Programmet skal generere tilfeldige regnestykker ved kall på *Math.random*. Brukeren må svare riktig 3 ganger på rad for å få regnestykker med svar over 10. Programmet skal dessuten kun generere regnestykker med positive svar.

Programmet skal lages som to klasser: en applett *AddSubApplett* som implementerer brukergrensesnittet og en modellklasse *MaskinLærer* som skissert under. Fullfør kodingen av klassene.



```
public class MaskinLærer {  
    // Sett inn deklarasjon av instansvariable  
  
    public String nyttRegnestykke(char oper) {  
        // Metoden tar en parameter som angir operator,  
        // '+' eller '-', og returnerer et tilfeldig  
        // regnestykke som bruker denne operatoren.  
        // Eksempel på returverdi når oper er '+': "2 + 3"  
    }  
    public String svar(int n) {  
        // Metoden tar brukerens forslag til svar på gjeldende  
        // regnestykke og returnerer kvittering  
        // "Riktig" eller "Galt".  
    }  
}
```

Oppgave 3 (30%)

I denne oppgaven skal du lage en metode *javaTilHtml* som skal produsere HTML-kode fra et Java-program:

- Nøkkelord skal settes i fet font slik: `<b>while</b>`
- Hver linje skal avsluttes med HTML-kode for linjeskift: `<br>`

Navn på Java-filen som skal leses og navn på HTML-filen som skal lages er parametre:

```
public void javaTilHtml(String javaFilnavn, String htmlFilnavn)
```

Du kan anta tilgang på en hjelpemetode (skal ikke programmeres) som avgjør om en tekststreng er et nøkkelord i Java:

```
public boolean javaOrd(String s)
```

Tips: Du kan lese Java-filen linje for linje, og anta at nøkkelord er omsluttet av blanke tegn. Du blir ikke trukket om sammenhengende blanke tegn blir slått sammen til ett blankt tegn.

Virkemåten til metoden *javaTilHtml* blir vist under ved hjelp av et eksempel. Anta filen K.java inneholder følgende Java-program:

```
public class K {  
    public K() {  
    }  
}
```

(Oppgave 3 fortsetter på neste side)



Metodekallet `javaTilHtml("K.java", "K.htm")` skal da produsere en HTML-fil K.htm med følgende innhold:

```
<html>
<body>
<b>public</b> <b>class</b> K { <br>
  <b>public</b> K() { <br>
    } <br>
  } <br>
</body>
</html>
```