



Høgskolen i Telemark

Avdeling for allmennvitenskaplege fag

**EKSAMEN
I
5608 PROGRAMMERING**

03.05.2011

Tid:	9-14
Målform:	Bokmål/nynorsk
Sidetal:	7, inkl. framside
Hjelpemiddel:	Alt skriftleg materiale
Merknader:	Vekting er gitt ved starten av kvar oppgåve
Vedlegg:	Ingen

Eksamensresultata blir offentleggjort på studentweb.



Avdeling for allmennvitenskaplege fag

Oppgave 1**Vekting: 20%**

- 1 - a) Forklar kort hva som blir skrevet ut når programmet Oppgave1a blir kjørt på en maskin.

```
public class Oppgave1a {
    public static void main(String[] args) {
        int x = 242;
        String y = x/10 + " < " + x%100 + ": ";
        y += (x/10 < x%100);
        System.out.println(y);
    }
}
```

- 1 - b) Koden nedenfor er et utsnitt hentet fra en metode der g er et Graphics-objekt. Skisser resultatet som fremkommer ved bruk av akkurat disse instruksjonene:

```
:
int x = 50, y = 50;
int f = 200;
g.drawRect(x, y, f, f);
g.drawLine(x, y, x+f, y+f);
g.drawOval(x+f/4, y+f/4, f/2, f/2);
g.drawLine(x+f, y, x, y+f);
:
```

- 1 - c) Skriv kun de instruksjonene (ikke et komplett program) som er nødvendig for å splitte innholdet i den ikke-tomme String-variabelen ord i to variabler del1 og del2, som inneholder annet hvert tegn av innholdet i ord i den samme rekkefølgen som de står i det opprinnelige ordet.

For eksempel:

```
ord = "oppgaver" ==> del1 = "opae"   og   del2 = "pgvr"
ord = "stien"    ==> del1 = "sin"    og   del2 = "te"
```

Oppgave 2**Vekting: 10%**

Skriv et komplett javaprogram som ber brukeren om å gi inn størrelsen på en *skråfigur* som deretter skal tegnes ut. Figuren skal være en slags skråstilt rett linje (se figurene under), der innlest størrelse (1-10) svarer til lengden på linja, tegnet ved hjelp av S-er i et dialogvindu.

Skråfigur, størrelse 3:

```
S
 S
  S
```

Skråfigur, størrelse 4:

```
S
 S
  S
   S
```

Skråfigur, størrelse 5:

```
S
 S
  S
   S
    S
```

Oppgave 3**Vekting: 30%**

Et firma har kundeoppfølging via telefon. Firmaet selger en rekke produkter og ønsker å undersøke hvor mye tid som brukes på kundeoppfølging av de ulike produktene i løpet av en uke. Firmaet registrer de ulike produktene innen 8 produktgrupper (kodet som 1 – 8) sammen med betjeningstid for hver oppringning. Opplysningene blir registrert på en sekvensiell tekstfil for hver ukedag, og dette dagnavnet brukes som filnavn. Registreringene gjøres etter hvert som telefonsamtalene avsluttes.

onsdag.txt
2-140
1-345
7-555
:
:
4-445
2-290

På figuren til venstre ser vi at hver linje på en slik fil inneholder produktgruppekode, etterfulgt av betjeningstid som er angitt i et helt antall sekunder.

Bindestrek brukes som skilletegn mellom de ulike dataene på ei tekstlinje.

Skriv et komplett Java-program som firmaet kan bruke til å analysere tidsbruken knyttet til oppfølging av de ulike produktgruppene kodet som 1 – 8. Programmet skal først be om og lese inn navnet på en slik registreringsfil, og så beregne gjennomsnittlig betjeningstid for hver av de ulike produktgruppene denne dagen. Resultatene bør skrives ut på en ny fil kalt `resdagnavn.txt` (altså "res" foran det opprinnelige filnavnet).

Oppgave 4**Vekting: 30%**

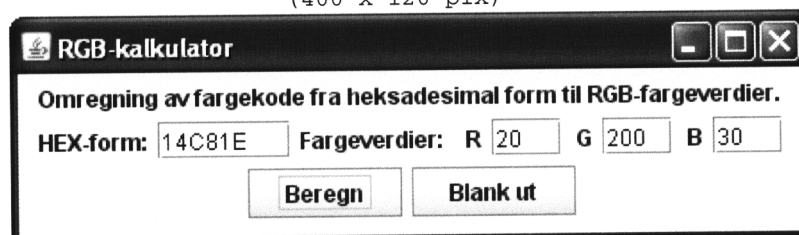
Det er ulike måter å oppgi fargekoder på, det kan f.eks. være på heksadesimal form eller på RGB-form (Rød, Grønn og Blå). Skriv et komplett Java-program som basert på et grafisk brukergrensesnitt (GUI), kan brukes til omregning av fargekode oppgitt på heksadesimal form til RGB-form.

På heksadesimal form har fargekoden 6 siffer. De to første gir rødverdi, de to midterste grønnverdi og de to siste blåverdi, der verdiene altså er gitt i det *heksadesimale* (eller 16-) tallsystemet. I det systemet brukes sifferne 0-9 som vanlig, mens de seks siste 10-15 må byttes ut med hvert sitt siffersymbol A-F slik: 10=A, 11=B, 12=C, 13=D, 14=E og 15=F.

På figuren under ser vi fra "HEX-form" at rødverdien (de to første tegnene) er "R=14", dvs. $R = 14_{16} = 1 \cdot 16 + 4 = 20$. Videre er grønnverdien "G=C8", dvs. $G = C8_{16} = 12 \cdot 16 + 8 = 200$. Blåverdien er "B=1E", dvs. $B = 1E_{16} = 1 \cdot 16 + 14 = 30$.

Det skal være et tekstfelt for innskriving av fargekode på heksadesimal form, en knapp for å beregne RGB-verdiene, og en for å kunne starte en ny omregning. De tre beregnede fargeverdiene har hvert sitt tekstfelt. Det er brukt 6, 3, 3 og 3 som lengder på tekstfeltene.

(400 x 120 pix)



Eksempel på et GUI for omregning mellom fargekoder.

Oppgave 5**Vekting: 10%**

Et dataspill blir programmert i Java. Der er det bruk for en egen klasse `Rektangel` som skal brukes til å modellere ulike firkanter. Det er behov både for vanlige rektangler og kvadrat (rektangler der sidekantene er like lange).

Klassen `Rektangel` har to instansvariabler, en for lengde og en for bredde. Siden klassen skal brukes i et dataspill, der ting foregår på skjerm, så vil alle størrelser være oppgitt i et helt antall pixler. Klassen skal ha metoder for å vise fram og endre på verdiene i instansvariablene, samt en `toString`-metode. Klassen skal ha konstruktør både for vanlige rektangler og for kvadrat, og et rektangelobjekt må kunne vise fram både sitt areal (produktet av lengde og bredde) og sin omkrets (summen av alle de fire sidelengdene).

Objekt fra `Rektangel`-klassen må kunne inngå i tabeller. Der er det aktuelt både med sortering og søk. Derfor må det være mulig å ordne slike objekt stigende etter areal.

Skriv komplett kode for klassen `Rektangel`.

SLUTT

--- NYNORSK VERSJON ---

Oppg ve 1**Vekting: 20%**

- 1 - b) Forklar kort kva som blir skriva ut n r programmet `Oppgave1a` blir kj yrd p  ein maskin.

```
public class Oppgave1a {
    public static void main(String[] args) {
        int x = 242;
        String y = x/10 + " < " + x%100 + ": ";
        y += (x/10 < x%100);
        System.out.println(y);
    }
}
```

- 1 - d) Koden nedanfor er eit utsnitt henta fr  ein metode der `g` er eit Graphics-objekt. Skisser resultatet som kjem fram ved bruk av akkurat desse instruksjonane:

```
:
int x = 50, y = 50;
int f = 200;
g.drawRect(x, y, f, f);
g.drawLine(x, y, x+f, y+f);
g.drawOval(x+f/4, y+f/4, f/2, f/2);
g.drawLine(x+f, y, x, y+f);
:
```

- 1 - e) Skriv kun dei instruksjonane (ikkje eit komplett program) som er n dvendig for   splitte innhaldet i den ikkje-tomme String-variabelen `ord` i to variablar `del1` og `del2`, som inneheld anna kvart teikn av innhaldet i `ord` i den same rekkef lgja som dei st r i det opphavslege ordet.

For eksempel:

```
ord = "oppg ver" ==> del1 = "op e"   og   del2 = "pgvr"
ord = "stien"    ==> del1 = "sin"    og   del2 = "te"
```

Oppg ve 2**Vekting: 10%**

Skriv eit komplett javaprogram som ber brukaren om   gi inn storleiken p  ein *skr figur* som deretter skal teiknast ut. Figuren skal vere ein slags skr stilt rett line (sj  figurane under), der innlest storleik (1-10) svarar til lengda p  lina, teikna ved hjelp av S-ar i eit dialogvindu.

Skr figur, storleik 3:

```
S
 S
  S
```

Skr figur, storleik 4:

```
S
 S
  S
   S
```

Skr figur, storleik 5:

```
S
 S
  S
   S
    S
```

Oppgave 3**Vekting: 30%**

Eit firma har kundeoppfølging via telefon. Firmaet sel ei rekkje produkt og ønskjer å undersøkje kor mykje tid som brukast på kundeoppfølging av dei ulike produkta i løpet av ei veke. Firmaet registrer dei ulike produkta innan 8 produktgrupper (koda som 1 – 8) saman med beteningstid for kvar oppringning. Opplysningane blir registrert på ei sekvensiell tekstfil for kvar vekedag, og dette dagnamnet brukast som filnamn. Registreringane gjerast etter kvart som telefonsamtalane avsluttast.

onsdag.txt
2-140
1-345
7-555
:
:
4-445
2-290

På figuren til venstre ser vi at kvar line på ei slik fil inneheld produktgruppekode, etterfølgd av beteningstid som er gitt i eit heilt tal sekund.

Bindestrek brukast som skiljeteikn mellom dei ulike data på ei tekstline.

Skriv eit komplett Java-program som firmaet kan bruke til å analysere tidsbruken knytta til oppfølging av dei ulike produktgruppene koda som 1 – 8. Programmet skal først be om og lese inn namnet på ei slik registreringsfil, og så rekne ut gjennomsnittleg beteningstid for kvar av dei ulike produktgruppene denne dagen. Resultata bør skrivast ut på ei ny fil kalla `resdagnamn.txt` (altså "res" framfor det opphavslege filnamnet).

Oppgave 4**Vekting: 30%**

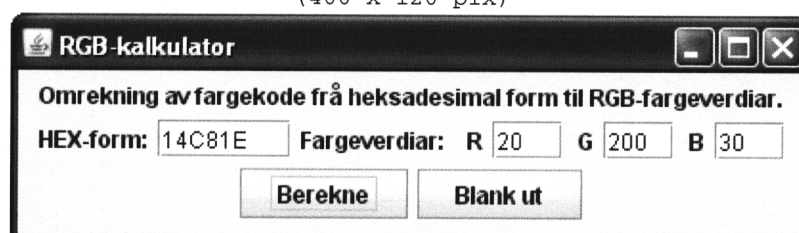
Det er ulike måtar å oppgi fargekodar på, det kan f.eks. vere på heksadesimal form eller på RGB-form (Rød, Grøn og Blå). Skriv eit komplett Java-program som basert på eit grafisk brukargrensesnitt (GUI), kan brukast til omrekning av fargekode gitt på heksadesimal form til RGB-form.

På heksadesimal form har fargekoden 6 siffer. Dei to første gir raudverdi, dei to midterste grønverdi og dei to siste blåverdi, der verdiane altså er gitt i det *heksadesimale* (eller 16-) talsystemet. I det systemet brukast sifra 0-9 som vanleg, medan dei seks siste 10-15 må byttast ut med kvart sitt siffersymbol A-F slik: 10=A, 11=B, 12=C, 13=D, 14=E og 15=F.

På figuren under ser vi frå "HEX-form" at raudverdien (dei to første teikna) er "R=14", dvs. $R = 14_{16} = 1 \cdot 16 + 4 = 20$. Vidare er grønverdien "G=C8", dvs. $G = C8_{16} = 12 \cdot 16 + 8 = 200$. Blåverdien er "B=1E", dvs. $B = 1E_{16} = 1 \cdot 16 + 14 = 30$.

Det skal være eit tekstfelt for innskriving av fargekode på heksadesimal form, ein knapp for å berekne RGB-verdiane, og ein for å kunne starte ei ny omrekning. Dei tre berekna fargeverdiane har kvart sitt tekstfelt. Det er brukt 6, 3, 3 og 3 som lengder på tekstfelta.

(400 x 120 pix)



Eksempel på eit GUI for omrekning mellom fargekodar.

Oppgave 5**Vekting: 10%**

Eit dataspel blir programmert i Java. Der er det bruk for ein eigen klasse `Rektangel` som skal brukast til å modellere ulike firkantar. Det er behov både for vanlege rektangel og kvadrat (rektangel der sidekantane er like lange).

Klassen `Rektangel` har to instansvariablar, ein for lengde og ein for bredde. Sidan klassen skal brukast i eit dataspel, der ting skjer på skjerm, så vil alle storleikar vere oppgitt i eit heilt tal pixlar. Klassen skal ha metodar for å vise fram og endre på verdiane i instansvariablane, samt ein `toString`-metode. Klassen skal ha konstruktør både for vanlege rektangel og for kvadrat, og eit rektangelobjekt må kunne vise fram både sitt areal (produktet av lengde og bredde) og sin omkrets (summen av alle dei fire sidelengdene).

Objekt frå `Rektangel`-klassen må kunne inngå i tabellar. Der er det aktuelt både med sortering og søk. Difor må det vere mogleg å ordne slike objekt stigande etter areal.

Skriv komplett kode for klassen `Rektangel`.

SLUTT