Høgskolen i Sørøst-Norge

NOEN EKSEMPLER PÅ AKTUELLE

EKSAMENSOPPGAVER

FOR

PRG1000B GRUNNLEGGENDE PROGRAMMERING

H-2017

Dette emnet er nytt, og det er ikke gitt noen eksamen her tidligere. Oppgavene er derfor hentet fra gamle eksamenssett i andre programmeringsemner, men de er tilpasset dette nye emnet. Noen av oppgavene vil bli gjennomgått som del av undervisningen og emnerepetisjonen.

Oppgavene er delt i tre grupper: 1, 2 og 3. Et eksamenssett kan bestå av to oppgaver fra gruppe 1, to fra gruppe 2 og en av oppgavene i gruppe 3. Vi ser av vektingen av gruppene at de to første oppgavene utgjør 2 x 10%, de to neste 2 x 20% og den siste 40% alene.

Til eksamen har dere 4 timers arbeidstid, dvs. 240 minutt. Det blir 24 minutt pr 10%, og altså 48 minutt på de to første, 96 minutt på de to neste og 96 minutt på den tredje og siste oppgaven. Eller avrundet til ca 1 time på de to første oppgavene, 1.5 time på de to neste og 1.5 time på den siste oppgaven.

Høgskolen i Sørøst-Norge
Handelshøyskolen
Institutt for økonomi og IT
Hallvard Eikas plass, 3800 Bø

Oppgave 1 ---- Tolk en gitt kode, eller skriv litt kode 2 x 10%

a) Forklar kort hva som blir skrevet ut når programmet `Oppgave1a` kjøres på en maskin.

```
public class Oppgave1a {
  public static void main(String[] args) {
    int t = 373;
    String u = t + ": ";
    if ( t%10 == t/100 )
      u += "s";
    else
      u += "u";
    System.out.println(u);
  }
}
```

b) Koden nedenfor er et utsnitt fra run-metoden i et EasyGraphics-program. Skisser resultatet som fremkommer ved bruk av akkurat disse setningene:

```
:
:
int x = 50, y = 50;
int f = 100;
drawRectangle(x, y, 2*f, 2*f);
drawCircle(x, y, f/2);
drawRectangle(x+f, y, f, f);
drawRectangle(x, y+f, f, f);
drawCircle(x+f, y+f, f);
:
:
```

c) Skriv kun de instruksjonene (ikke et komplett program) som er nødvendig for å bytte om på de to bakerste sifrene i det positive heltallet som (allerede) er lagret i variabelen `spesTall`.
Noen eksempel: `1232 --> 1223,  567 --> 576` og `48 --> 84`

d) Skriv kun metoden (ikke et komplett program) som tar et heltall, beregner og returnerer antall siffer i dette heltallet.

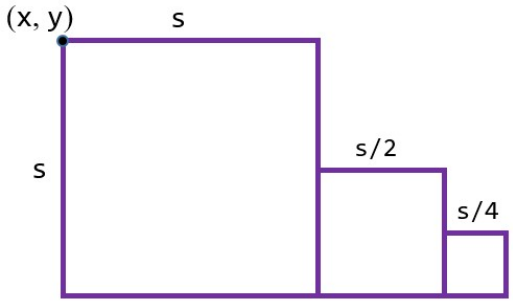
e) Hva blir utskriften fra dette programmet når det kjøres?

```
import static java.lang.System.*;
public class Oppg1e {
  public static void main(String[] args) {
    int a = 7, b = 4;
    int u = a + b;
    int v = a - b;
    a = (u - v)/2;
    b = (u + v)/2;
    out.println("a = " + a + ", b = " + b);
  }
}
```

f) En metode i en verktøyklasse skal telle antall tall som er *mindre enn* et gitt tall (parameter `søkeTall`) i en gitt tabell (parameter `tab`). Men metoden slik den er satt opp nedenfor lar seg ikke kompilere!
Skriv koden på nytt uten alle feilene, og slik at den fungerer etter hensikten.

```
private static String antall(int tab[], int søkeTall) {
  for (i=1  i<=lenght  i++);
    if (tab[i] > søketall);
      antall++;
  return antall;
}
```

g) Skriv kun EasyGraphics-metoden `kvadratTrapp` (ikke et komplett program), som til tre heltallsparametre `x`, `y` og `s`, tegner tre kvadrat stilt opp i trappefasong - som vist på figuren.



Oppgave 2 ---- Skriv et lite program 2 x 20%

a) Skriv et komplett Java-program som ber brukeren om å gi inn størrelsen på en vinkelfigur som deretter skal skrives ut i konsollet. Figuren skal være en slags L (se figurene under), der innlest størrelse svarer til lengden på hver sidekant, tegnet ved hjelp av X-er.

Vinkelfigur, størrelse 3:

```
X
X
X X X
```

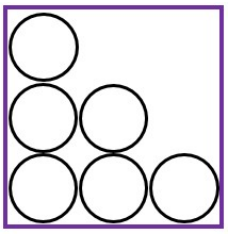
Vinkelfigur, størrelse 4:

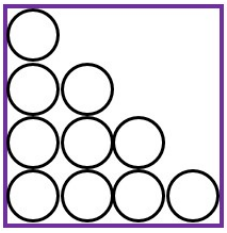
```
X
X
X
X X X X
```

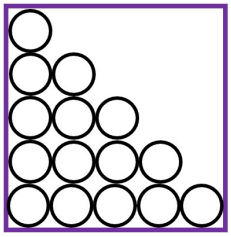
Vinkelfigur, størrelse 5:

```
X
X
X
X
X X X X X
```

b) Skriv run-metoden i et EasyGraphics-program som ber brukeren om å gi inn høyden (2-9) på en kulestabel (se fig. under) og som så fyller hele det grafiske vinduet med en slik stabel. På figurene er alle de tre vinduene like store, f.eks. 400 pix x 400 pix.

høyde = 3: 

høyde = 4: 

høyde = 5: 

c) Skriv et komplett Java-program som ber brukeren om å gi inn *størrelsen* på en *sifferfigur* som deretter skal tegnes ut. Figuren skal være ytterkanten på en firkant (se figurene under), der sidelengdene er lik størrelsen. Innlest størrelse (sifferet 2 – 9) skal også være siffertegnet som brukes til å tegne figuren i konsollet (der det er fast-font).

Sifferfigur, størrelse 2:

```
2 2
2 2
```

Sifferfigur, størrelse 3:

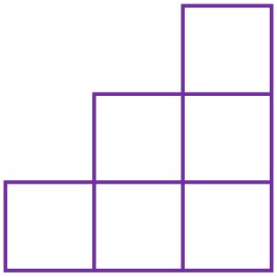
```
3 3 3
3   3
3 3 3
```

Sifferfigur, størrelse 4:

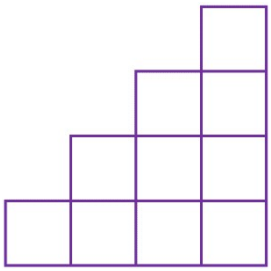
```
4 4 4 4
4     4
4     4
4 4 4 4
```

d) Skriv run-metoden i et EasyGraphics-program som ber brukeren om å gi inn antall trinn i en steintrapp (2-9), og som så tegner en slik trapp med trinn fra nedre venstre hjørne av det grafiske vinduet og oppover mot høyre. På figurene under er alle de tre vinduene like store, f.eks. 400 pix x 400 pix.

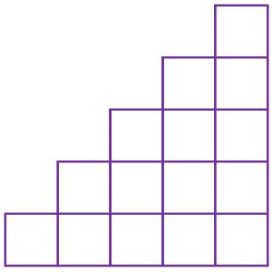
høyde = 3:



høyde = 4:



høyde = 5:



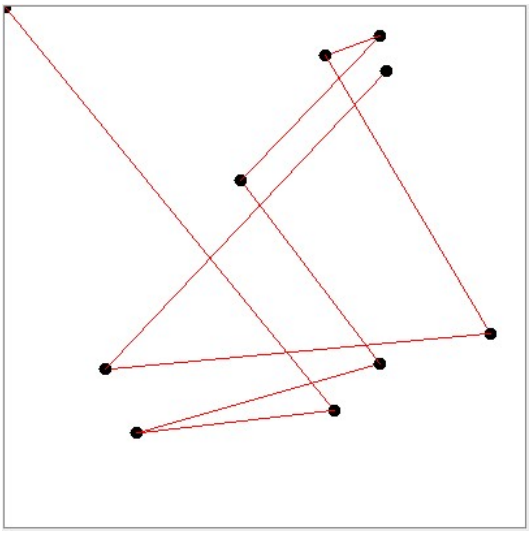
e) Skriv et komplett Java-program som ber brukeren om å gi inn ønsket *antall tall* i utskriften av en tallserie kalt Fibonacci-tallene. Disse heltallene er definert ved at neste tall er summen av de to foregående tallene i serien, og de to første tallene er begge 1. Utskriften fra tallserien skal være slik om innlest antall er 7:

```
1  1  2  3  5  8  13
```

NB! Programmet skal ikke skrive mer enn 10 Fibonacci-tall pr linje.

f) Skriv run-metoden i et EasyGraphics-program som ber brukeren om å gi inn et antall punkt på en en tilfeldig vandring (se fig. under), og som så trekker en rød linje fra origo til et tilfeldig valgt punkt (markert med svart sirkel med radius 5), og derfra videre til neste tilfeldige punkt som markeres, helt til ønsket antall punkt er tegnet i det grafiske vinduet.

På denne figuren er det tegnet en tilfeldig vandring med 10 punkt.



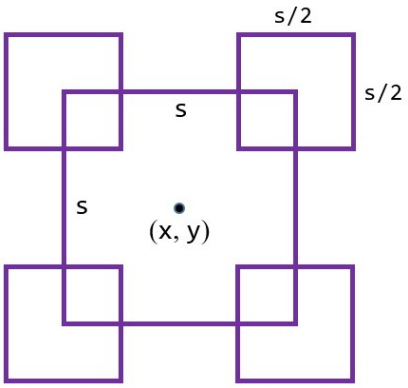
g) Skriv et komplett Java-program som ber brukeren om å gi et beløp i det gamle britiske myntsystemet, og regner om verdien til det nye myntsystemet der det er 100 *pence* i 1 *pund*. I det gamle myntsystemet (før 1971) var det slik inndeling med base i samme pundverdien:

1 pund = 20 shilling = 240 (gamle) pence.

Programmet skal be om antall mynter i de tre gamle valørene, og så regne om til verdien i pund. Resultatet av omregningen til dagens verdi skal skrives i et dialogvindu.

h) [1] Skriv kun EasyGraphics-metoden `kvadrat` (ikke et komplett program) som til tre heltallsparametre `x`, `y` og `s`, tegner et kvadrat med midtpunkt i (`x`, `y`) og sidelengder `s`.

[2] Sett opp kun de setningene som er nødvendige for å tegne et stort kvadrat med midtpunkt i (`x`, `y`) = (150, 150) og sidelengde `s` = 50, og fire små kvadrat med sentrum i hvert av de fire hjørnene til det store kvadratet, og som har `s/2` som sine sidelengder. Se figuren



[3] Lag samme figur som i [2], men anta nå at `x`, `y` og `s` alle er (allerede) innleste heltallsverdier.

Oppgave 3 ---- Skriv et større program 1 x 40%

a) På sekvensielle tekstfiler er det lagret høydedata for elever på alle barneskolene i en gitt kommune fra skoleåret 2014-15. Det er én fil for hver barneskole. Dataene er anonymisert slik at hver linje på en slik fil kun har lagret følgende data for en elev: *Kjønn* (kodet som **0**=jente og **1**=gutt), *klassetrinn* (1-7) og *høyde* i hele cm. Se eksempel på figur A under til venstre.

Figur A

skole1.txt

```
0 4 132
1 3 129
1 1 118
0 7 150
1 2 123
  : (Datalinjene fortsetter)
  :
```

Figur B

snittskole1.txt

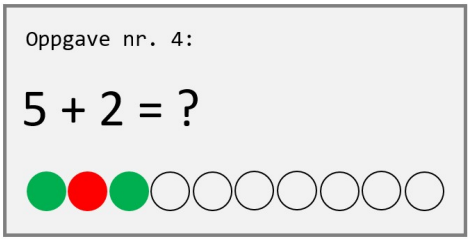
```
1: J-118 G-119
2: J-123 G-121
   :
   :
6: J-144 G-143
7: J-150 G-148
```

Kommunens helseråd ønsker en oversikt over gjennomsnittshøydene (i hele cm) for hvert kjønn på hvert klassetrinn på en aktuell barneskole.

Skriv et komplett Java-program som leser inn filnavnet for den aktuelle skolen, går gjennom hele tekstfilen, og linje for linje henter ut de relevante dataene slik at de ulike gjennomsnittene kan beregnes. Til slutt skal programmet skrive ut de to snitthøydene (Jente-snitthøyde og Gutt-snitthøyde) for hvert enkelt klassetrinn, på hver sin linje, på en ny tekstfil med «`snitt`» satt foran navnet til inndatafila. Følg oppsettet på figur B ovenfor til høyre.

b) Skriv et komplett Java-program som elever i barneskolen kan bruke til å øve på helt enkel hoderegning med pluss og minus med tall i området 1– 9 (siffer). Programmet skal lage 10 oppgaver om gangen (i hver gjennomføring får hver elev 10 oppgaver) ved å velge tilfeldig to heltall i det gitte området, velger en av de to regneoperasjonene og beregne fasitsvaret. Ett og ett regnestykke skal presenteres, og så skal programmet vente på svar fra eleven. NB! Ved minus (subtraksjon) skal programmet sørge for at svaret alltid blir positivt.

Programmet skal bruke EasyGraphics til å gjøre oppgavene mer fargerike og lettere å oppfatte for elevene. Når de vet svaret på en oppgave, skriver de tallet i inndatafeltet i det grafiske vinduet (og trykker [enter]-knappen). Dersom svaret er rett (fasit), skal den sirkelen i bunnen av vinduet som hører til oppgavenummeret bli grønn. Men dersom svaret er feil, blir den rød. Raden med sirkler gjør det lettere for eleven å se hvor mange oppgaver som er løst så langt, hvor mange som gjenstår, hvor mange svar har vært rett og hvor mange har vært feil. Se figuren under til venstre, som viser situasjonen i det eleven skal svare på oppgave nr. 4.



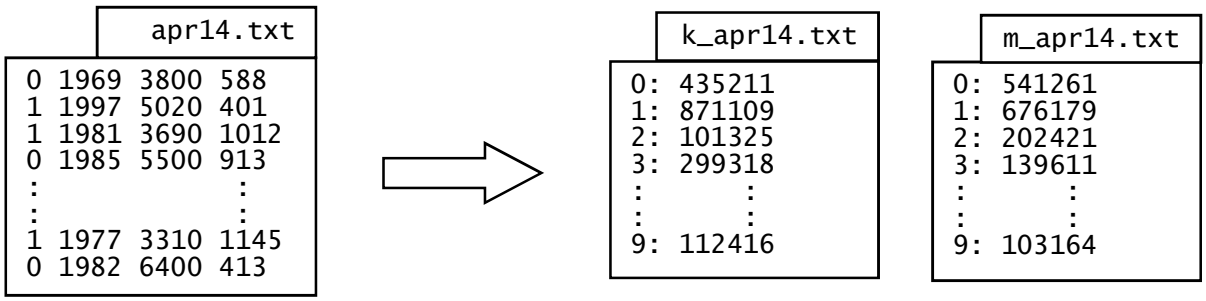
Figuren over til høyre viser avslutningen av programmet med en statusrapport og tilbakemelding til eleven. Det er ønskelig at programmet har ulike tekster til å beskrive elevens prestasjon, f.eks. kan alle 10 rette gi «Supert!», 9 rette kan gi «Flott!», og så videre nedover. Det er greit om to – tre poengsummer gir den samme tekstlige beskrivelsen.

Det grafiske vinduet er satt opp med 500 pix x 250 pix. Det er valgt fontstørrelse 25 punkt på topptekstен, og 50 punkt på oppgavene (og resultatbeskrivelsen til slutt).

c) På sekvensielle tekstfiler er det på hver linje lagret opplysninger om kunder i en nettbutikk med oppsett for hver måned som vist til venstre på figuren nedenfor. Filnavnene er sammensatt av måned og år, mens hver linje på filen består av:

kjønn (0=kvinne/1=mann), fødselsår, postnummer og kjøpesum (i hele kr)

Informasjonen på en slik fil skal splittes på to nye filer, én for kvinner og én for menn. På hver ny fil blir alle kjøpesummene summert i grupper etter det første sifferet i postnummeret, dvs. kjøpesummene brytes opp i 10 ulike deler av landet. Se figuren nedenfor til høyre, som først viser total kjøpesum for kvinner inndelt etter første siffer i postnummeret og deretter en tilsvarende fil med samme opplysninger om menns kjøp i samme måned.



Skriv et komplett Java-program som kan brukes til å splitte informasjonen på en slik fil, og så beregne de totale kjøpesummene fordelt etter første siffer i postnummeret. Kundenes alder er ikke relevant her (brukes ikke). Programmet skal først be om og lese inn navnet på den aktuelle registreringsfilen, så opprette to ny filer og skrive ut de aktuelle oversiktene.

d) En studentgruppe har vinlotteri annenhver fredag. Da selger de lodd fra en gammeldags loddblokk, der de river ut og selger papirlapper som er nummerert 1 – 100. De selger alltid minst 10 lodd, men det er ikke alle lodd som blir solgt slik at trekningen på en av fredagene f.eks. kan foregå i området 48 – 77. Det blir aldri solgt fra mer enn én loddbok. Det trekkes alltid ut tre ulike vinnertall (3 premier) blant de solgte loddene. Kun ett tall blir trukket og presentert i vinduet om gangen. Neste tall blir trukket når trekningsansvarlig gir beskjed.

Figur: Vinduet etter at tekningen er ferdig, med tredje og siste tall på plass i utskriften.

Til å gjennomføre trekningen skal det lages et EasyGraphics-program som kan vise trekningsresultatet ved ett vinnertall per linje, samt premienummer (1-3).

Programmet skal lese inn både det minste aktuelle loddnummeret og det største aktuelle loddnummeret for trekningen. Disse to tallene skal skrives ut som en «kvittering» for at trekningen skjer blant de faktisk solgte loddnumrene.

I figuren er vinduet satt opp med 400 x 400 piksler, og det er brukt fontstørrelsene 20 og 40 på teksten. Legg vekt på at programmet faktisk leverer trekningsresultat, deretter på at utseendet er mest mulig likt med det som er vist på figuren.

NB! Det blå rutenettet skal ikke være en del av utskriften, det er kun til hjelp under utvikling av programmet. Rutene er 50 x 50 piksler. Fontene er Arial, og de to fontstørrelsene er 70 pkt og 35 pkt. Som vist på figuren svarer 70 pkt ca til 50 piksler i høyden, mens sifrenes bredde er ca 40 piksler («ca» fordi det er proporsjonalfont). Fontstørrelsen 35 pkt svarer til ca 25 piksler i høyden, og ca 10 piksler i bredden.