

INF100 Grunnkurs i programmering

Semesteroppgave 3 høsten 2016

Institutt for informatikk
Universitetet i Bergen

October 26, 2016

Formål:	Få erfaring med bruk av filer, løkker, metoder, tabeller og generell øving i å skrive programmer.
Innlevering:	.ZIP-fil med java-kildekode for alle oppgavene (lever kun .java-filer, ikke .class-filer). Vennligst navngi .ZIP-filen [Brukernavn]-semopp3.zip, der [Brukernavn] byttes ut med ditt brukernavn (3 bokstaver fulgt av 3 tall).
Innleveringsfrist:	10.11.2016, klokken 10:00

Merk at besvarelsen på denne oppgaven teller på sluttkarakteren.

Hvor stor prosentandel hver oppgave og deloppgave teller er gitt i tittelen til oppgaven/deloppgaven. Det vil bli gitt trekk for generelle problemer med koden, som at den ikke kompilerer, ikke er pakket og levert på korrekt måte, manglende kommentering, med mer. Det vil også bli lagt vekt på valg av variabelnavn. Det er en fordel i forhold til poengsum å levere det arbeidet dere har fått gjort, selv om dere ikke skulle få helt til en oppgave. Les nøye gjennom hver oppgave, sørg for at programmet gjør det som oppgaven ber om og har all funksjonalitet og eventuell håndtering av feil som det er krav om. Vennligst unngå bruk av norske tegn i koden og kommentarene dere leverer. Vi forutsetter individuell løsning av oppgavene.

Oppgave 1: Labyrintvandring (25 %)

Vi ønsker et program som lar brukeren vandre i en labyrint. Labyrinten er gitt av en todimensjonal `char`-tabell. I tabellen er spillerens startposisjon markert med `'s'`. Programmet skal da finne denne posisjonen, og bytte ut verdien i tabellen med et mellomrom. Vegger er markert med `'*'`. Programmet skal iterativt skrive ut labyrinten med spillerens posisjons markert med en `'s'`, og så spørre spilleren hvor den vil gå (nord, sør, øst eller vest), og oppdatere spillerens posisjon, helt til spilleren velger å avslutte. Spilleren skal ikke kunne gå gjennom vegger. Skriv en metode `kanGaaTil` som returnerer `true` dersom spilleren kan

gå til den nye posisjonen, og ellers returnerer **false**, og ordne så programmet stopper spiller fra å gå steder blokkert av vegger. Bruk labyrinten gitt nedenfor. Kall programmet `LabyrintVandring.java`.

Labyrint:

```
int labyrintBredde = 4;
int labyrintHoejde = 5;
char[][] labyrint = {
    { '*', '*', '*', '*' },
    { '*', ' ', ' ', ' ', '*' },
    { '*', ' ', ' ', '*', '*' },
    { '*', 'S', '*', '*' },
    { '*', '*', '*', '*' },
};
```

Eksempel på kjøring:

Labyrint:

```
****
*  *
* **
*S**
****
```

Hvor vil du gaa? Skriv nord, soer, vest eller oest for aa gaa i respektiv retning, eller avslutt for aa avslutte.

nord

Labyrint:

```
****
*  *
*S**
* **
****
```

Hvor vil du gaa? Skriv nord, soer, vest eller oest for aa gaa i respektiv retning, eller avslutt for aa avslutte.

nord

Labyrint:

```
****
*S *
* **
* **
****
```

Hvor vil du gaa? Skriv nord, soer, vest eller oest for aa gaa i respektiv retning, eller avslutt for aa avslutte.

vest

Veien er blokkert!

Labyrint:

```
****
*S *
* **
* **
```

```

****
Hvor vil du gaa?  Skriv nord, soer, vest eller oest for aa gaa i respektiv
retning, eller avslutt for aa avslutte.
oest
Labyrint:
****
* s*
* **
* **
****
Hvor vil du gaa?  Skriv nord, soer, vest eller oest for aa gaa i respektiv
retning, eller avslutt for aa avslutte.
avslutt

```

Oppgave 2: Labyrintskattejakt (25 %)

Nå ønsker vi å legge på skatter som spilleren kan plukke opp i labyrinten. For å gjøre spillet mer fleksibelt ønsker vi også at labyrinter nå skal lastes fra brukergitte filer. Formatet på labyrint-filene er som følger: Første rad består av et heltall som gir labyrintens bredde, og andre av et heltall som gir labyrintens høyde. Hver følgende rad inneholder tegnene i bredden i labyrinten, sammenhengende. La brukeren oppgi banen til filen som labyrinten skal leses fra. Skriv en metode `lesLabyrintFraFil` som leser labyrinten fra en fil. Metoden skal returnere en `char[][]`-tabell. Dersom programmet ikke kan lese filen så skal programmet avslutte med en feilmelding. Merk at unntak må håndteres på innsiden av metoden, og det holder ikke å kaste unntak videre fra `main`-metoden. Dersom de første to radene i filen ikke er heltall som er større enn 0 så skal programmet avslutte med en feilmelding. Dersom det er for få rader i filen eller en rad er for kort i forhold til hva som ble spesifisert i de første to radene så skal programmet avslutte med en feilmelding. På kartet skal vegger være markert med '*', og gull med 'g'. Antall gull som spilleren har plukket opp skal skrives ut hver runde sammen med labyrinten. Når spiller lander på et felt med gull så skal gullet byttes ut med et mellomrom og spillerens gullteller økes. Når spilleren har plukket opp alt gullet så skal spillet avslutte med melding om at spilleren tok alt gullet. Lag en kopi av programmet du utviklet i oppgave 1, og kall det videreutviklede programmet `LabyrintSkattejakt.java`.

Labyrint-fil: testLabyrint1.txt

```

6
5
*****
* g  *
* ** *
*s*g *
*****

```

Eksempel på kjøring:

Oppgi filen som labyrinten skal leses fra:

testLabyrint1.txt

6

5

Spiller-gull: 0

Labyrint:

* g *

* ** *

*s*g *

Hvor vil du gaa? Skriv nord, soer, vest eller oest for aa gaa i respektiv retning, eller avslutt for aa avslutte.

nord

Spiller-gull: 0

Labyrint:

* g *

*s** *

* *g *

Hvor vil du gaa? Skriv nord, soer, vest eller oest for aa gaa i respektiv retning, eller avslutt for aa avslutte.

nord

Spiller-gull: 0

Labyrint:

*sg *

* ** *

* *g *

Hvor vil du gaa? Skriv nord, soer, vest eller oest for aa gaa i respektiv retning, eller avslutt for aa avslutte.

oest

Du tok 1 gull!

Spiller-gull: 1

Labyrint:

* s *

* ** *

* *g *

Hvor vil du gaa? Skriv nord, soer, vest eller oest for aa gaa i respektiv retning, eller avslutt for aa avslutte.

oest

Spiller-gull: 1

Labyrint:

```

*  s  *
* ** *
* *g *
*****
Hvor vil du gaa?  Skriv nord, soer, vest eller oest for aa gaa i respektiv
retning, eller avslutt for aa avslutte.
oest
Spiller-gull: 1
Labyrint:
*****
*  s*
* ** *
* *g *
*****
Hvor vil du gaa?  Skriv nord, soer, vest eller oest for aa gaa i respektiv
retning, eller avslutt for aa avslutte.
soer
Spiller-gull: 1
Labyrint:
*****
*   *
* **g*
* *g *
*****
Hvor vil du gaa?  Skriv nord, soer, vest eller oest for aa gaa i respektiv
retning, eller avslutt for aa avslutte.
soer
Spiller-gull: 1
Labyrint:
*****
*   *
* ** *
* *gs*
*****
Hvor vil du gaa?  Skriv nord, soer, vest eller oest for aa gaa i respektiv
retning, eller avslutt for aa avslutte.
vest
Du tok 1 gull!
Du tok alt gullet i labyrinten!
Spiller-gull: 2

```

Oppgave 3: Labyrintmonstre (25 %)

Vi legger inn monstre i labyrinten. Disse er markert med 'm'. Lag en kopi av programmet du utviklet i oppgave 2 og kall den `LabyrintMonstre.java`. Når spiller faller på et felt med et monster så byttes feltet i kartet ut med et mellomrom. Monsteret utfordrer så spilleren til kamp med terninger. Spilleren og monsteret kaster hver runde en 6-sidet terning. Dersom spilleren får høyest

verdi så gir monsteret opp og lar spilleren passere. Dersom begge terninger har lik verdi så er det uavgjort, og det blir omkamp. Dersom spillerens terning har lavest verdi så kan spilleren bestikke monsteret til omkamp ved å gi 1 gull. Når spilleren er tom for gull så spiser monsteret spilleren, og spillet avslutter. Programmet skal ha en metode `terningkast` som returnerer tilfeldige terningkast.

Labyrint-fil: `testLabyrint2.txt`

```
6
5
*****
* gm *
* ** *
*s*gm*
*****
```

Labyrint-fil: `testLabyrint3.txt`

```
9
9
*****
*   m   g*
* *****
*   g   *
*m*****
* *   g*m*
* * *** *
*g*s   *
*****
```

Eksempel på kjøring:

Oppgi den filen som labyrinten skal lastes fra:
`testLabyrint2.txt`

```
6
5
Spiller-gull: 0
Labyrint:
```

```
*****
* gm *
* ** *
*s*gm*
*****
```

Hvor vil du gaa? Skriv nord, soer, vest eller oest for aa gaa i respektiv retning, eller avslutt for aa avslutte.

nord

```
Spiller-gull: 0
Labyrint:
*****
* gm *
```

```

*s** *
* *gm*
*****
Hvor vil du gaa?  Skriv nord, soer, vest eller oest for aa gaa i respektiv
retning, eller avslutt for aa avslutte.
nord
Spiller-gull: 0
Labyrint:
*****
*sgm *
* ** *
* *gm*
*****
Hvor vil du gaa?  Skriv nord, soer, vest eller oest for aa gaa i respektiv
retning, eller avslutt for aa avslutte.
oest
Du tok 1 gull!
Spiller-gull: 1
Labyrint:
*****
* sm *
* ** *
* *gm*
*****
Hvor vil du gaa?  Skriv nord, soer, vest eller oest for aa gaa i respektiv
retning, eller avslutt for aa avslutte.
oest
Et monster blokkerer veien din, og utfordrer deg til aa kaste hoeyeste terning.
Du og monsteret kaster terninger.
Ditt kast: 5
Monsterets kast: 4
Du kastet hoeyere enn monsteret, saa monsteret ga opp.
Spiller-gull: 1
Labyrint:
*****
*  s *
* ** *
* *gm*
*****
Hvor vil du gaa?  Skriv nord, soer, vest eller oest for aa gaa i respektiv
retning, eller avslutt for aa avslutte.
oest
Spiller-gull: 1
Labyrint:
*****
*  s*
* ** *
* *gm*
*****
Hvor vil du gaa?  Skriv nord, soer, vest eller oest for aa gaa i respektiv

```

```

retning, eller avslutt for aa avslutte.
soer
Spiller-gull: 1
Labyrint:
*****
*      *
*  **s*
* *gm*
*****
Hvor vil du gaa?  Skriv nord, soer, vest eller oest for aa gaa i respektiv
retning, eller avslutt for aa avslutte.
soer
Et monster blokkerer veien din, og utfordrer deg til aa kaste hoyeste terning.
Du og monsteret kaster terninger.
Ditt kast: 1
Monsterets kast: 3
Monsteret kastet hoyere enn deg.
Du brukte 1 gull til aa bestikke monsteret til aa ta en omkamp.
Du og monsteret kaster terninger.
Ditt kast: 4
Monsterets kast: 1
Du kastet hoyere enn monsteret, saa monsteret ga opp.
Spiller-gull: 0
Labyrint:
*****
*      *
*  ** *
* *gs*
*****
Hvor vil du gaa?  Skriv nord, soer, vest eller oest for aa gaa i respektiv
retning, eller avslutt for aa avslutte.
vest
Du tok 1 gull!
Du tok alt gullet i labyrinten!
Spiller-gull: 1

```

Oppgave 4: Postpakker (25 %)

Tenk deg at et postkontor ønsker å få utviklet programvare til å håndtere informasjon om pakker. Design en **Postpakke**-klasse, som representerer postpakker. Klassen skal inneholde følgende informasjon som feltvariabler: mottakerens fulle navn (tekststreng), adresse (tekststreng) og postnummer (heltall), samt vekten til pakken (flytverdi). Det skal være en konstruktør som tar parametre for hva alle feltvariabler skal settes til og som setter dem lik disse, og det skal være en hentemetode per feltvariabel. Alle feltvariablene skal være private, og hentemetodene skal være offentlig. Kall klassefilen **Postpakke.java**. Skriv et program som lar bruker registrere postpakker representert med **Postpakke**-klassen. Programmet skal først spørre hvor mange postpakker som skal registreres, og skal

så for hver av dem spørre brukeren om mottakerens navn, adresse og postnummer, samt pakkens vekt, og registrere disse. Til sist skal programmet skrive ut alle de registrerte pakkene. Kall programmet `Postsystem.java`.

Eksempel på kjøring:

Hvor mange postpakker skal registreres?

3

Pakke 1

Oppgi mottakerens navn:

Ola Normann

Oppgi mottakerens adresse:

Testveien 123

Oppgi mottakerens postnummer:

1234

Oppgi pakkens vekt:

1,5

Pakke 2

Oppgi mottakerens navn:

Kari Normann

Oppgi mottakerens adresse:

Testveien 50

Oppgi mottakerens postnummer:

1234

Oppgi pakkens vekt:

1,6

Pakke 3

Oppgi mottakerens navn:

Julenissen

Oppgi mottakerens adresse:

Nordpolenveien 1

Oppgi mottakerens postnummer:

9999

Oppgi pakkens vekt:

0,1

Pakker:

Pakke 1

Mottakers navn: Ola Normann

Mottakers adresse: Testveien 123

Mottakers postnummer: 1234

Pakkens vekt: 1.5

Pakke 2

Mottakers navn: Kari Normann

Mottakers adresse: Testveien 50

Mottakers postnummer: 1234

Pakkens vekt: 1.6

Pakke 3

Mottakers navn: Julenissen

Mottakers adresse: Nordpolenveien 1

Mottakers postnummer: 9999

Pakkens vekt: 0.1