

IN1010 V19, Obligatorisk oppgave 7

Innleveringsfrist: Tirsdag 14.05. kl 23:59

Innledning

I denne oppgaven skal du bruke JavaFX for å lage et grafisk brukergrensesnitt (GUI) for labyrintprogrammet du lagde i obligatorisk oppgave 5. Før du begynner å programmere bør du legge en plan for hvordan brukergrensesnittet kan bygges opp på en fordelaktig måte både med tanke på logikk og visuelt design. Det kan være nyttig å tegne noen utkast for hånd.

Del A: Presentasjon av utvei

Skriv programmet slik at du bruker klassen `javafx.stage.FileChooser` for å la brukeren finne/velge filen med labyrinten.

Når brukeren har valgt labyrintfilen som skal åpnes, skal labyrinten vises grafisk. For den grafiske representasjonen av labyrinten skal programmet bruke klassen `javafx.scene.layout.GridPane`.

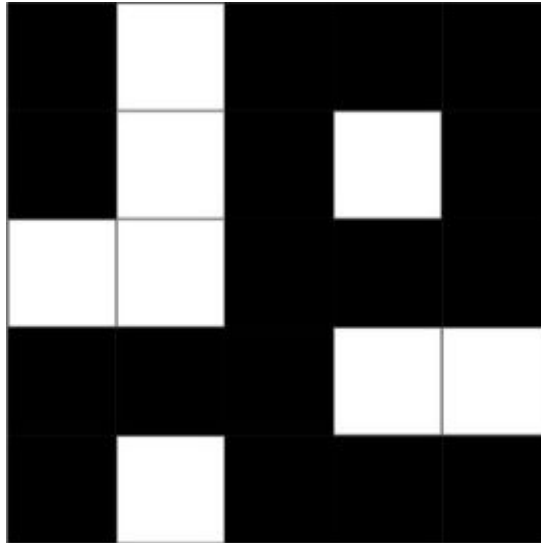
Når brukeren klikker på en hvit rute, skal du bruke programmet fra oblig 5 for å finne alle løsningene fra denne ruten. Du skal ta én løsning fra beholderen med løsninger, og vise denne i labyrinten. Er det flere enn én løsning, kan du vise den første, med informasjon om hvor mange løsninger som totalt ble funnet.

Konvertering av løsningsstrenger

I programmet fra oblig 5 skulle hver løsning være gitt på et slikt format:

`(1, 1) --> (1, 2) --> (0, 2)`

Det kan være vanskelig å hente ut relevant informasjon fra denne strengen for å utheve de riktige rutene i grensesnittet. Gitt at du har fulgt formatet ovenfor kan du derfor velge å benytte deg av en [ferdig utgitt metode](#) for å konvertere en løsningsstreng til et todimensjonalt array med boolske verdier. Verdier som er `true` indikerer at ruten er en del av en løst utvei. Verdier som er `false` indikerer at ruten ikke er en del av løsningen.



Figur 1: Labyrinten i fil 7.

Et eksempelkall på `losningStringTilTabell` i fil 7 (fra testfilene til oblig 5) ser slik ut:

```
String losning = "(1, 1) --> (1, 2) --> (0, 2)"; // mulig utvei fra (1,1) i fil 7
losningStringTilTabell(losning, 5, 5);
```

Dette vil returnere et `boolean[][]`-objekt med følgende verdier (T representerer true, . representerer false):

```
. . . . .
. T . . .
T T . . .
. . . . .
. . . . .
```

Metoden kan du finne [her](#).

Relevante Trix-oppgaver: [Alle oppgaver om GUI](#).

Oppsummering

Du skal levere alle klasser som skal til for at hovedprogrammet skal fungere, inkludert nødvendige klasser fra oblig 5 og eventuelle listeklasser du har tatt i bruk.

Alle delene av programmet må kompilere og kjøre på lfi-maskiner for å kunne få oppgaven godkjent. Unngå bruk av packages (spesielt relevant ved bruk av IDE-er som IntelliJ).

Ikke levér zip-filer!

Det går an å laste opp flere filer samtidig i Devilry.

Merk: Resten av oppgaven er frivillig.

Del B (valgfri, men sterkt anbefalt): [Vise flere utveier](#)

Om det er flere løsninger, er det nyttig å kunne se alle løsningene til labyrinten. Utvid GUI-et med knapper slik at løsningene kan vises en etter en når brukeren trykker på en knapp. I tillegg til på vise antall løsninger kan det også være lurt å vise hvilket løsningsnummer som presenteres akkurat nå.

Del C (valgfri tilleggsoppgave): GUI med FXML

Har du lyst på en ekstra og litt annerledes utfordring? Da kan du skrive en løsning som benytter seg av FXML, et XML-format (*Extensible Markup Language*) spesiallaget for å bygge opp GUI i Java. Vil du ha en innføring i FXML kan du ta en titt på [denne guiden](#) (skrevet av Andreas Bergem).