Skattejakt

Januar 2025

I denne oppgaven skal dere lage tre klasser: Rute, Brett og Skattejakt. Programmet skal la bruker velge posisjon fra et 9*9 spillebrett, hvor én tilfeldig rute er en skattekiste. De skal få ti forsøk på å gjette seg fram til svaret, og få oppgitt hvor langt unna de befinner seg skattekista. Hvis de gjetter hvor skattekista er, så har de vunnet.

1 Rute

Klassen Rute skal representere hver rute i brettet, og ha en metode for å returnere avstand til en annen rute. Ruten skal ha:

- En **konstruktør** som tar inn **rad** (int) og **kolonne** (int) som parameter, og lagrer disse i egne instansvariabler.
- En metode `avstandTil(Rute annen_rute)` som tar inn et **Rute**-objekt som parameter, og returnerer avstand til denne ruta. **Hint:** metoden skal returnere absoluttverdien til rutas rad minus den andre rutas rad **pluss** absoluttverdien til rutas kolonne minus den andre rutas kolonne. Importer Java-biblioteket **java.lang.Math**, og bruk funksjonen **Math.abs()** for å få absoluttverdien til en verdi.

2 Brett

Brett-klassen skal ha et 9*9 rute stort todimensjonalt array som representerer spillbrettet, og en peker til ruta som har skattekista. Et brett skal ha funksjonalitet for å plassere skattekista på en tilfeldig plass i brettet, returnere avstand til skattekista og sjekke om en rute er skattekista. Lag:

- En **konstruktør** som ikke tar inn noen parametre, men som har instansvariabler `Rute[][] brett` som er en todimensjonal array med 9 rader og 9 kolonner, og som inneholder 9*9 Rute-objekter og `Rute skattekiste` som initialiseres til null. Metoden
  `plasserSkattekiste()` skal også kalles inne i konstruktøren (gjør dette etter dere har lagd metoden).

- En metode `plasserSkattekiste()` som setter instansvariabelen `skattekiste` til å peke på en tilfeldig rute på spillebrettet. **Hint:** Importer **java.util.Random** på toppen av filen, og lag et **Random**-objekt. Kall på `.nextInt()` for å få et tilfeldig tall. Pass på at dette tallet er gyldig.
- En metode `avstandTilSkattekiste(int rad, int kolonne)` som returnerer ruta som ligger på indeks `[rad][kolonne]` sin avstand til skattekista. Husk å bruke metoden dere lagde i Rute-klassen.
- En metode `erSkattekiste(int rad, kolonne)` som sjekker om ruta som ligger på indeks `[rad][kolonne]` er skattekista.

3 Skattejakt

Skattejakt-klassen skal representere selve spillet, og er den eneste klassen som det skal opprettes et objekt av i hovedprogrammet. Et Skattejakt objekt skal ha et Brett og en metode som lar bruker spille spillet. Lag:

- En **konstruktør** som ikke tar inn noen parametre. Lag en instansvariabel `Brett brett` som opprettes til å peke på et Brett-objekt.
- En metode `spill()` som har en løkke som kjører 10 ganger, og lar bruker oppgi en posisjon på spillbrettet (du trenger ikke sjekke om indeksen er gyldig, anta at den alltid er det). Hvis oppgitt posisjon er posisjonen til skattekista, skal bruker gratuleres, og metoden skal returnere. Hvis ikke, skal det oppgis hvor langt unna bruker er skattekista. Hvis løkka blir ferdig, vet vi at bruker har tapt, og det skal printes ut en passende melding. Print også ut posisjonen til skattekista når bruker har tapt.

Lag et hovedprogram, og sjekk at `spill()` metoden fungerer som forventet.