

Lab 2 – FourGame

Denna labben går ut på att skapa ett fyra-i-rad-spel där två spelare skall kunna spela mot varandra i turordning. Lägg in lämpliga meddelanden (`println("lämpligt meddelande")`) där det behövs för att leda spelarna genom de olika momenten.

Uppgift 1 – Board

I den första uppgiften skall vi skapa en klass för brädet, kallad Board.

- Storleken på brädet skall sättas i klassens konstruktör. Brädet modelleras av en två-dimensionell matris för tecken, `char[][]`.
- Brädet skall skrivas ut i terminalen med metoden `show()`.
- Skall ha `get(int i, int j)` och `set(int i, int j)` metoder för tecken i arrayen.
- Skall ha en `clear(int i, int j)` metod för att nollställa en position, alltså göra den tom.

TIPS: För att testa om klassen Board fungerar, skapa en main-klass vid namn FourGame, som skapar en Board i sin main().

Uppgift 2 – dropPiece

Dags att göra pjäser! En metod kallad `dropPiece(char piece, int column)` skall skapas i Board-klassen för att släppa en pjäs i en column som spelaren får välja. Just nu är det bara en spelare med en typ av pjäs.

- Metoden skall ta emot en `char` piece och en `int` column. Om den är utanför tillåtet intervall (Hur många kolumner spelplanen består av) så skall den be spelaren ange nytt värde.
- Om den är innanför intervallet skall metoden se efter vart pjäsen landar när den släppts. Om kolumnen är full skall den be spelaren välja en annan kolumn.
- Om ovanstående är uppfyllt skall metoden placera en lämplig symbol på rätt position i kolumnen.

Uppgift 3 – turn

Nu skall de tidigare metoderna köras med en metod kallad `turn()`, som hanterar turordningens gång och ordning. Denna kan skrivas i main-programmet. Vi räknar bara med en spelare fortfarande.

- Metoden skall skriva ut brädets nuvarande utseende.
- Sen skall spelaren få lägga sin pjäs genom att välja en kolumn (Tips: **Scanner** och kör `dropPiece()`).

Uppgift 4 – Checker

Skapa en ny klass kallad Checker som skall kolla om någon fått fyra i rad.

checkRows(char symbol, Board board)

- En metod som kollar rad för rad om spelaren fått fyra-i-rad vågrätt.

TIPS: Använd nested for-loops.

checkColumns(char symbol, Board board)

- En metod som kollar kolumn för kolumn om spelaren fått fyra-i-rad lodrätt.

TIPS: Liknande som `checkRows()`, men ändra på ordningen hos for-looparna.

(För VG-betyg)

`checkDiagonalRight(char symbol, Board board)` och `checkDiagonalLeft(char symbol, Board board)`

- En metod som kollar diagonalerna om spelaren fått fyra-i-rad.

Uppgift 5 – Fler spelare

Nu skall det gå att spela två (eller fler om du vill) spelare. Detta kan göras på många olika sätt. Två exempel:

1. Låta `turn()` ta emot ett in-argument `Player p` som talar om vems tur som spelas.
 2. Låta en variabel ändras beroende på vems tur det är. En typ av flagg-symbol för att visa vems tur det är.
- Hur ni än väljer att göra det möjligt med flera spelare så skall varje spelare använda ett särskilt tecken som sina pjäser. Därför behöver vi skapa en `Player`-klass som har en instans-variabel **`char`** `piece`, med dess tecken för en pjäs.

TIPS: Om du vill fylla en array med en särskild symbol så görs detta lätt med `Arrays.fill(char[], char)` som går att använda om man importerar **`java.util.Arrays`**; För mer info om hur `Arrays.fill` fungerar: <https://www.geeksforgeeks.org/arrays-fill-java-examples/>