Vier gewinnt

Teste das Programm und beschreibe das Verhalten, beim
- Bewegen der Maus:

_____

- Mausklick:

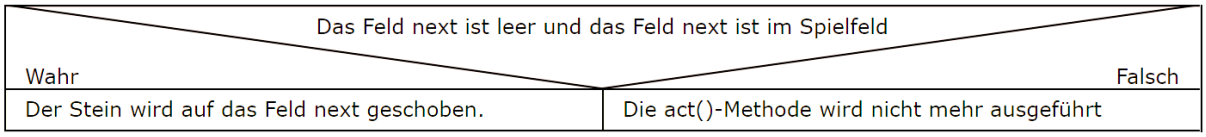
_____

- Ruhen der Maus:

_____

1. Steine sollen unten liegen bleiben

Programmiere eine bedingte Anweisung in der Methode **Stein.act()**, die folgenden Algorithmus umsetzt:



Verwende hierzu die Methoden **gameGrid.isEmpty(next)**, **gameGrid.isInGrid(next), this.setLocation(next)** und **this.setActEnabled(false)**

2. Steine sollen erst nach einem Mausklick fallen

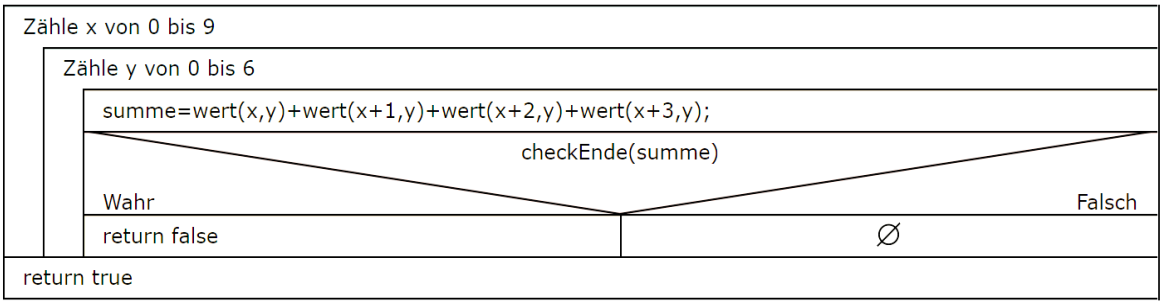
1. Damit die Steine nicht gleich beim Entstehen herunter fallen, muss im Konstruktor der Klasse Stein die Methode **this.setActEnabled(false)** aufgerufen werden.
2. Füge in der Methode **mouseEvent(…)** der Klasse Spielfeld die Methoden **activeActor.setActEnabled(true)** und **this.delay(1000)** ein, so dass sie nach einem Linksklick aufgerufen werden.

3. Wertung der Steine

1. Ersetze den Rumpf der Methode **weiter()** durch folgenden Code:

a) Deklariere eine lokale Variable summe vom typ int.

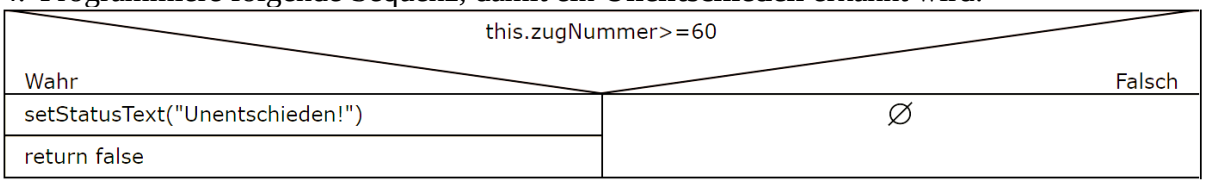
b



2. Begründe, welche Sieg-Kombination mit dieser Methode erkannt wird:

_____

3. Kopiere den Inhalt der y-Zählschleife dreimal und passe ihn so an, dass die anderen Sieg-Kombinationen erkannt werden.
4. Programmiere folgende Sequenz, damit ein **Unentschieden** erkannt wird:



Vier gewinnt

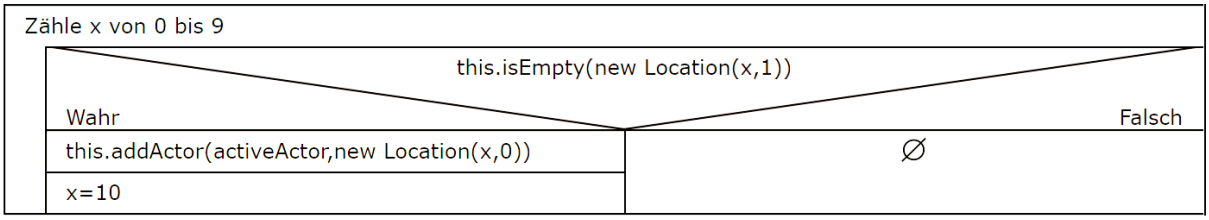
4. Fehlerbeseitigung

1. Untersuche, was passiert, wenn du mehr als 6 Steine aufeinander stapelst:

   _____

   _____

2. Ersetze die Zeile **this.addActor(activeActor,new Location(0,0));** in der Methode **neuerStein()** durch folgende Sequenz:



   Beschreibe die Funktion dieser Sequenz:

   _____

   _____

3. Ersetze den ersten Block in der Methode **mouseEvent(…)** durch folgenden Code:

```
if (mouse.getEvent() == GGMouse.move) {  // Maus wurde bewegt
    Location newLoc=toLocation(mouse.getX(), 0);
    if (isEmpty(toLocation(mouse.getX(),getCellSize()+1))){
        activeActor.setLocation(newLoc);
    }
}
```

   Beschreibe die Funktion dieser Sequenz:

   _____

   _____

5. Optimierung des Codes

Die Abbruchbedingungen der x- und y-Schleifen passen nur, wenn das Spielfeld 7x10 Felder groß ist. Mit den Methoden **getNbHorzCells()** und **getNbVertCells()** können die Zahl der Spalten und Zeilen ausgelesen werden.

Ersetze im Code die Zahlen 10,7 und 60 durch die entsprechenden Methoden.