
















## Aufgaben mit JAVA-Karol:


Speichert das Projekt `2016_Karolneu_Schueler` bei Euch ab.

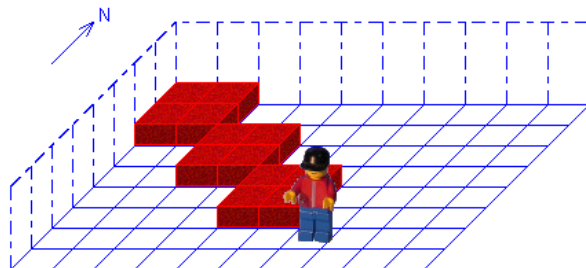
1.  Öffnet das Projekt. Ihr seht, dass es aus zwei verschiedenen Klassen besteht. Versucht zuerst einmal, verschiedene Welten zu erzeugen und anschließend verschiedene Roboter in diesen Welten (dazu müsst ihr angeben, in welcher Welt der Roboter existieren soll) .
2.  Verstreut auch mal ein paar Ziegel in der erzeugten Welt und probiert die einzelnen Methoden der Klasse `ROBOTER` aus. Ich hoffe ihr erinnert euch.


Ergänzt jetzt die Klasse `ROBOTER` an der von mir markierten Stelle um folgende Methoden. Geht dabei davon aus, dass sich der Roboter an seiner Startposition befindet und keine sonstigen Hindernisse vorhanden sind.

3.  ...Erzeugt mit Hilfe einer Wiederholung die Methode (der Name sollte selbsterklärend sein) **ViererReiheLegen**.
4.  Erstellt eine Methode **ReiheZurWand**, die Karol von einer beliebigen Position aus eine Reihe Ziegel bis zur Wand legen lässt.
5.  Erstellt eine Methode **FuenferQuadrat** die ein Quadrat der Seitenlänge 5 mit Hilfe der Methode **ViererReiheLegen** erstellt.
6.  Bringt dem Roboter die Methoden **SchrittNachRechts** und entsprechend **SchrittNachLinks** bei.
7.  Erstellt mit Hilfe der zweiten Methode eine Methode **MauerBauen**, welche ihn eine Mauer der Breite 5 und der Höhe 4 bauen lässt.
8.  Erstellt eine Methode **HundertZiegel**, welche unseren Roboter einhundert Ziegel hintereinander am Rand seiner Welt entlang ablegen lässt.
9.  Erstellt nun eine Methode **InDieEcke**, die unseren Roboter von einer beliebigen Stelle aus in seine ursprüngliche Ecke gehen lässt. Achtung: Es gibt eine Methode **IstBlickNorden()**, die ihr dazu braucht.
10.  Karol soll eine komplette Schicht Ziegel legen. Formuliert die entsprechende Methode **EbeneLegen**. Tipp: Lasst Karol einen Stein in die Anfangsecke legen und sich drauf stellen. Übrigens, wenn ihr zwei Bedingungen gleichzeitig prüfen wollt geht das mit `&&` (UND) bzw. `||` (ODER)

11.  Erstelle nun eine Klasse `TEST`, die eine Welt `deutschland` anlegt und in dieser Welt einen Roboter `mustermann`. Schreibe anschließend eine Testmethode, die deinen Roboter ein, zwei der bereits programmierten Methoden ausführen lässt.
12.  Erstelle (in der Klasse `ROBOTER`) eine Methode `ReiheSammeln`, die alle Ziegel auf dem Weg zur Wand entfernt, mitzählt, wie viele aufgesammelt wurden und diesen Wert zu Schluss am Bildschirm ausgibt.  
 Teste deine Methode, indem du in der Testmethode erst ein paar Ziegel in der erzeugten Welt verstreut (den Befehl dazu findest du in der Klasse `WELT`) und dann die Sammel-Methode für deinen Roboter aufrufst.
13.  Erstelle eine Methode `zufallsschritt`, die abhängig von einer erzeugten Zufallszahl den Roboter einen Schritt machen lässt oder nicht. Verwende dazu den Befehl `(int) (Math.random()*2)` um zufällig eine 0 oder 1 zu erzeugen.  
 Erstelle nun in der Testklasse eine Methode `Wettrennen`, die zwei Roboter gegeneinander zu einem Wettrennen antreten lässt und am Bildschirm ausgibt, wer das Rennen gewonnen hat. Welcher Roboter dabei einen Schritt macht oder nicht, soll durch Zufall erfolgen.
14.  Verbessere dein Programm, indem du sie 20mal gegeneinander antreten lässt und mitzählst, wer wie oft gewonnen hat. Nach jedem Rennen müssen beide Roboter natürlich zurück auf ihre Startposition, bzw. das Rennen in die andere Richtung starten.
15.  Erstelle (siehe oben) eine Methode `AllesSammeln`, welche die ganze Ebene säubert und wieder am Bildschirm ausgibt, wie viele Ziegel gesammelt wurden.

16.  Erstelle eine möglichst effektive Methode `DreiQuadrate`, die folgendes Muster legen lässt.



17.  Erstelle eine möglichst effektive Methode, die folgendes Muster legen lässt.

