

Check und Prepare

Vorbereitende Übungen



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Autoren: Nhan Huynh und Darya Nikitina
Fachbereich: Informatik
Übungsblatt: 01

Version: 20. November 2021
Semesterübergreifend

V1 FopBot



Beschreiben Sie kurz in ihren eigenen Worten worum es sich bei FopBot handelt, wie die Welt aufgebaut ist und welche Grundfunktionen jeder Roboter beherrscht.

Für alle Aufgaben auf diesem und allen weiteren Übungsblättern: In der Abschlussklausur werden Sie keine Hilfsmittel zur Verfügung haben. Üben Sie also schon zu Beginn auch ohne Entwicklungsumgebung und nur mit Stift und Papier zu programmieren. Abschließend können Sie dann ihre vollständige Lösung in die Entwicklungsumgebung übertragen und überprüfen.

V2 Liegen geblieben



Betrachten Sie den folgenden Codeausschnitt/führen Sie ihn selbst einmal aus:

```
1 Robot bot = new Robot(0,2,UP ,0);  
2 bot.move();  
3 bot.turnLeft();  
4 bot.putCoin ();
```

V3 Rechteck



Schreiben Sie ein Programm, welches zwei Roboter putbot und pickbot erstellt. Dabei soll putbot mit Coins ein Rechteck der Höhe 5 und der Breite 3 zeichnen. Es sollen nur die Seiten des Rechtecks gezeichnet werden, die restlichen innen liegenden Felder des Rechtecks bleiben unberührt. Nachdem das Rechteck gezeichnet wurde, soll pickbot alle Coins wieder einsammeln. Überlegen Sie sich, wie Sie das Programm mit nur einer Schleife pro Roboter gestalten können.

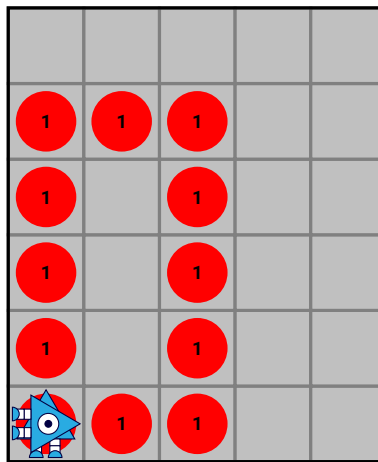


Abbildung 1: Fertiggestelltes Rechteck durch putbot

V4 Bedingungen I



Betrachten Sie folgenden Codeausschnitt:

```
1 Robot bot1 = new Robot(3, 1, UP, 1);
2 bot1.move();
3 if (bot1.isNextToACoin()) {
4   bot1.pickCoin();
5 } else {
6   bot1.putCoin();
7 }
```

Beschreiben Sie in eigenen Worten, welchem Zweck dieser Codeausschnitt dient. Erweitern Sie außerdem den Code so, dass bot1 nur einen Coin ablegt, wenn er auch mindestens einen besitzt.

V5 Variablen



Legen Sie eine Variable `int a` an und setzen Sie ihren Wert auf 127. Jetzt legen Sie eine weitere Variable `int b` an und setzen Ihren Wert auf 42. Was gibt nun der Ausdruck `int c = a % b` wieder? Beschreiben Sie in Ihren eigenen Worten, welche Berechnung mit dem `%` Operator durchgeführt wird.

V6 Bedingungen II



Ihr Kommilitone ist etwas tippfaul und lässt deswegen gerne einmal Klammern weg, um sich Arbeit zu sparen. Er hat in seinem Code eine Variable `int number` angelegt, in der er eine Zahl speichert. Ist diese Zahl kleiner als 0, so möchte er das Vorzeichen der Zahl umdrehen und sie anschließend um 1 erhöhen. Ist die Zahl hingegen größer als 0, so möchte er die Zahl verdoppeln. Dazu schreibt er folgenden Code:

```
1 if(number < 0) number = -number;
2 number = number + 1;
3 else number = number * 2;
```

Kann der Code so ausgeführt werden? Beschreiben Sie den Fehler, den ihr Kommilitone begangen hat.

Nachdem Sie ihren Kommilitonen auf den obigen Fehler hingewiesen haben, überarbeitet er seinen Versuch. Wie sieht es mit folgender Variante aus?

```
1 if(counter > 0) counter = counter * 2;
2 if(counter < 0) counter = -counter;
3 counter = counter + 1;
```

Da müssen Sie wohl selbst ran. Erstellen Sie ein Codestück, um den Sachverhalt korrekt zu implementieren.

V7 Schleifen I



Schreiben Sie den folgenden Ausdruck mithilfe einer `for`-Schleife:

```
1 Robot bot = new Robot(0, 0, UP, 1);
2 int i = 3;
3 while (i < 9) {
4     bot.move();
5     i = i + 1;
6 }
```

V8 Schleifen II



Ihr klammerfauler Kommilitone hat auch diesmal wieder zugeschlagen und versucht den Code aus vorheriger Aufgabe kürzer zu schreiben. Was sagen Sie dazu?

```
1 Robot bot = new Robot(0, 0, UP, 1);
2 int i = 3;
3 while (counter < 9) bot.move(); i = i + 1;
```

V9 Anzahl an Umdrehungen



Legen Sie eine Variable `int numberOfTurns` an und setzen Sie ihren Wert zu Beginn auf 0. Erstellen Sie dann einen neuen Roboter und platzieren Sie ihn an der Stelle (8, 2). Er schaut dabei nach links und besitzt keine Coins. Lassen Sie den Roboter nun geradewegs auf die Stelle (0, 2) zusteuern und alle Coins auf seinem Weg aufsammeln. Liegen mehrere Coins auf einer Stelle, so soll er alle Coins aufsammeln. Bei jedem Aufsammeln, erhöhen Sie den Wert von `numberOfTurns` um 1. Hat er am Ende die Stelle (0, 2) erreicht, soll er sich `numberOfTurns`-mal nach links drehen.

V10 Vorsicht Wand!



Gehen Sie in dieser Aufgabe davon aus, dass Sie einen Roboter `wally` an der Position (0, 0) erstellt haben und er nach rechts schaut. An der Position (0, 0) befindet sich eine vertikale Wand die den Weg nach $(x + 1, 0)$ versperrt, die x -Koordinate ist allerdings unbekannt. Schreiben Sie ein kleines Programm, mit dem Sie den Roboter bis vor die Wand laufen lassen, direkt vor der Wand einen Coin ablegen, um dann wieder an die Ausgangsposition (0, 0) zurückzukehren.

Hinweis: Es gibt die Funktion `isFrontClear()`, mit der getestet werden kann, ob sich in Blickrichtung des Roboters vor ihm direkt eine Wand befindet.

V11 Codeverständnis



Beschreiben Sie ausführlich, welches Verhalten der nachfolgende Code umsetzt. Bei Fragen zur Funktionalität einzelner Methoden, werfen Sie einen Blick in die entsprechenden Vorlesungsfolien.

```
1 Robot robot = new Robot(0, 0, RIGHT, 99999);
2
3 int counter = 0;
4 while (robot.getX() < World.getWidth() - 1) {
5     robot.move();
6     counter = counter + 1;
7 }
8
9 robot.turnLeft();
10
11 for (int i = 0; i < counter; i++) {
12     if (i % 2 == 0 && robot.hasAnyCoins()) {
13         robot.putCoin();
14     }
15     robot.move();
16 }
17
18 robot.turnLeft();
19 while (robot.isFrontClear()) {
20     robot.move();
21 }
22
23 robot.turnLeft();
24 while (robot.getX() != 0 || robot.getY() != 0) {
25     robot.move();
26 }
27
28 robot.turnOff();
```

V12 Navigator



Gegeben seien vier Variablen:

```
int startX    int startY
int destinationX  int destinationY
```

Ihr Roboter befindet sich zu Beginn an der Position (startX, startY) und schaut in eine beliebige Richtung. Schreiben Sie ein Programm, das ihn von dieser Position auf die Position (destinationX, destinationY) laufen lässt.