

Check und Prepare

Lösungsvorschläge



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Autoren: Nhan Huynh und Darya Nikitina
Fachbereich: Informatik
Übungsblatt: 00

Version: 03. Juni 2021
Semesterübergreifend

Information: Dieses Übungsblatt legt den Grundstein für alle weiteren Java Übungsblätter. Von Ihnen wird erwartet, dass Sie sich intensiv mit dem Blatt beschäftigen, da wir alle hier beschriebenen Formalitäten auf allen weiteren Übungsblättern als gegeben voraussetzen.

V0 Ist IntelliJ starbereit?

Importieren Sie die Vorlage V00 aus Moodle und führen Sie die Klasse `IntelliJReady.java` aus. Wenn Sie alles korrekt installiert haben, wird Ihnen in der Konsole unten Ihre installierte Java Version ausgegeben. Achten Sie darauf, dass hier die Version 17 in Verbindung mit Temurin (AdoptOpenJDK) ausgegeben wird.

Die Ausgabe der Konsole kann folgendermaßen aussehen:

```
java.home = C:\Program Files\Eclipse Foundation\jdk-17.0.3-hotspot
java.version = 17.0.3
java.runtime.version = 17.0.3+7
```

V1 Erste Schritte mit FopBot



Öffnen Sie nun die Klasse `FirstStepsBot.java`. Dort finden Sie eine Stelle, welche mit `TODO` gekennzeichnet ist. Fügen Sie hier Ihren Code ein, der folgendes umsetzt:

1. Erstellen Sie einen Roboter namens `alice`, der auf der Position (4,4) steht und nach rechts blickt. Er hat zu Beginn drei Coins in seiner Tasche.
2. Lassen Sie `alice` nun zwei Schritte nach vorne laufen.
3. Drehen Sie `alice` nun so, dass er nach oben blickt.
4. Lassen Sie `alice` einen Schritt nach vorne laufen.
5. Legen Sie einen Coin von `alice` ab.
6. Lassen Sie `alice` zwei Schritte nach vorne laufen.
7. Legen Sie zwei Coins mit `alice` ab.
8. Drehen Sie `alice` nun so, dass er nach links blickt.
9. Lassen Sie `alice` zwei Schritte nach vorne laufen.
10. Lassen Sie `alice` den Coin aufheben.
11. Lassen Sie `alice` einen Schritt nach vorne laufen.

Hinweis: Sie können durch den folgenden Verweisen mehr über die Funktionsweise von FopBot erfahren:

- Sourcecode: <https://github.com/FOP-2022/FOPBot>
- Dokumentation: <https://fop-2022.github.io/FOPBot/fopbot/package-summary.html>

Um mögliche Verständnisprobleme zu klären kann sich auch ein Blick in die Dokumentation und den Sourcecode lohnen.

Lösungsvorschlag:

```
1 import fopbot.Direction;
2 import fopbot.Robot;
3 import fopbot.World;
4
5 public class FirstStepsBot {
6     public static void main(String[] args) {
7         World.setSize(10, 10);
8         World.setDelay(200);
9         World.setVisible(true);
10        World.putCoins(4, 7, 1);
11        // 1.
12        Robot alice = new Robot(4, 4, Direction.RIGHT, 3);
13        // 2.
14        alice.move();
15        alice.move();
16        // 3.
17        alice.turnLeft();
18        // 4.
19        alice.move();
20        // 5.
21        alice.putCoin();
22        // 6.
23        alice.move();
24        alice.move();
25        // 7.
26        alice.putCoin();
27        alice.putCoin();
28        // 8.
29        alice.turnLeft();
30        // 9.
31        alice.move();
32        alice.move();
33        // 10.
34        alice.pickCoin();
35        // 11.
36        alice.move();
37    }
38 }
```

Information: Die Direction eines FopBot wird mithilfe von einer Enumeration umgesetzt. Zugriff auf die jeweiligen Richtungen erhält man durch Direction.X, wobei für X jeweils UP, DOWN, LEFT und RIGHT eingesetzt werden kann. Damit man nicht jedes Mal Direction schreiben muss, kann man folgendes importieren:

```
import static fopbot.Direction.*;
```

Mehr Information dazu kann mithilfe des folgenden Verweises gefunden werden:

- <https://docs.oracle.com/javase/tutorial/java/java00/enum.html>

V2 Quadrat



Öffnen Sie nun die Klasse `Square.java`. Dort finden Sie eine Stelle, welche mit `TOD0` gekennzeichnet ist. Fügen Sie hier Ihren Code ein, der folgendes umsetzt:

Zu Beginn platzieren Sie zwei Roboter in der Welt, von denen beide 20 Coins besitzen. Der erste Roboter befindet sich in Position $(0, 0)$ und blickt nach rechts, der andere befindet sich in Position $(9, 9)$ und blickt nach links. Ihre Aufgabe ist es nun, ein (nicht ausgefülltes) Quadrat mithilfe der beiden Roboter, durch ablegen von Coins, zu zeichnen. Dabei soll sich am Ende des Programms jeder Roboter im Startpunkt des jeweils anderen befinden. In Abbildung 1 finden Sie einen Vorher-Nachher-Vergleich dieser Situation.

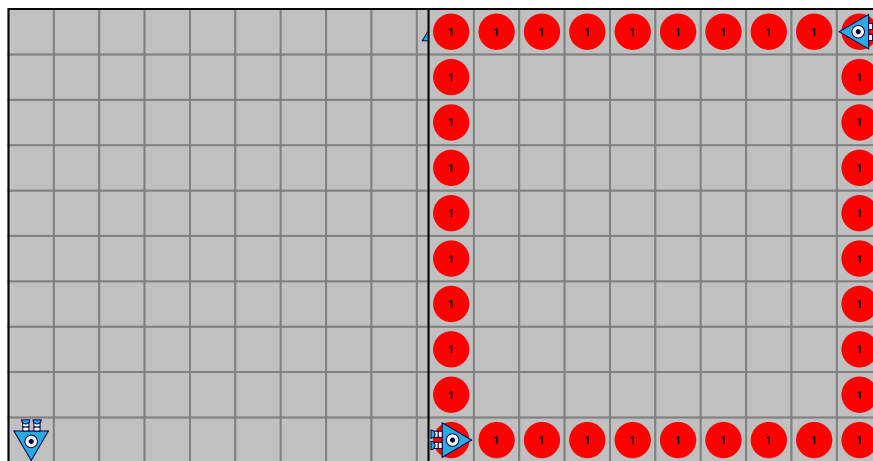


Abbildung 1: Vorher-Nachher-Vergleich

Verbindliche Anforderung: Das Laufen und Ablegen von Coins darf nur innerhalb einer Schleife umgesetzt werden, in der in jedem Durchlauf jeder der Roboter genau einen Coin ablegt! Lediglich das Drehen der Roboter darf außerhalb einer Schleife geschehen.

Lösungsvorschlag:

```
1 import fopbot.Direction;
2 import fopbot.Robot;
3 import fopbot.World;
4
5 public class Square {
6
7     public static void main(String[] args) {
8         World.setSize(10, 10);
9         World.setDelay(100);
10        World.setVisible(true);
11
12        Robot alice = new Robot(0, 0, Direction.RIGHT, 20);
13        Robot bob = new Robot(9, 9, Direction.LEFT, 20);
14
15        while ((alice.getX() != 9 || alice.getY() != 9) &&
16              (bob.getX() != 0 || bob.getY() != 0)) {
17            alice.putCoin();
18            bob.putCoin();
19
20            // Reached a corner
21            if (!alice.isFrontClear()) {
22                alice.turnLeft();
23            }
24            if (!bob.isFrontClear()) {
25                bob.turnLeft();
26            }
27
28            alice.move();
29            bob.move();
30        }
31    }
32 }
```