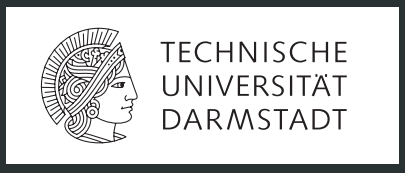
Funktionale und objektorientierte Programmierkonzepte
Übungsblatt 00



Prof. Karsten Weihe

Wintersemester 22/23	v1.0
Themen:	Programmieren in Java mit Hilfe von FOPBot
Relevante Foliensätze:	01a
Abgabe der Hausübung:	28.10.2022 bis 23:50 Uhr

Hausübung 00 — Gesamt: 5 Punkte

Java-Einstieg mit FOPBot

> **Beachten Sie die Seite *Verbindliche Anforderungen für alle Abgaben* im Moodle-Kurs.**
>
> Verstöße gegen verbindliche Anforderungen führen zu Punktabzügen und können die korrekte Bewertung Ihrer Abgabe beeinflussen. Sofern vorhanden, müssen die in der Vorlage mit `TODO` markierten `crash`-Aufrufe entfernt werden. Andernfalls wird die jeweilige Aufgabe nicht bewertet.

Die für diese Hausübung relevanten Verzeichnisse sind `src/main/java/h00` und ggf. `src/test/java/h00`.

H1: Matrikelnummer in Moodle — 1 Punkt

In dieser Aufgabe erhalten Sie einen Punkt dafür, dass Sie Ihre Matrikelnummer *fehlerfrei* in Moodle eintragen oder bereits eingetragen haben. Wenn Sie Ihre Matrikelnummer bereits in Moodle eingetragen haben, vergewissern Sie sich *dringend*, dass diese korrekt ist!

> **Hinweis:**
>
> In unserem Studierenden-Guide finden Sie im Abschnitt *Vorbereitung → Eintragen Ihrer Matrikelnummer* eine Anleitung dazu, wie Sie Ihre Matrikelnummer in Moodle eintragen.
>
> **Wenn Sie Ihre Matrikelnummer nicht oder nicht korrekt in Moodle hinterlegt haben, kann Ihre Zulassung und Ihr Bonus unter Umständen nicht korrekt verbucht werden!**

H2: Einrichten von Java und Ihrer Entwicklungsumgebung

Zum Bearbeiten unserer Hausübungen verwenden Sie eine Java-Entwicklungsumgebung. Hierzu zählt zum Beispiel *IntelliJ*: Die Verwendung von IntelliJ wird von uns *dringend* empfohlen und unterstützt.

> **Hinweis:**
>
> In unserem Studierenden-Guide finden Sie im Abschnitt *Vorbereitung → Installieren von Java* und *Vorbereitung → Installieren von IntelliJ* Guides zur Installation von Java bzw. IntelliJ.

H3: Herunterladen und Importieren unserer Vorlagen

In der FOP stellen wir Ihnen für *jede* Hausübung eine Vorlage bereit, die zur Bearbeitung der jeweiligen Hausübung verwendet werden *muss*.

> **Hinweis:**
>
> In unserem Studierenden-Guide finden Sie im Abschnitt *Vorbereitung → Installieren von Java* und *Vorbereitung → Installieren von IntelliJ* Guides zur Installation von Java bzw. IntelliJ.
>
> Beachten Sie weiter die auf jedem Übungsblatt am Anfang erwähnte Seite *Verbindliche Anforderungen für alle Abgaben* sowie die mitgelieferten Hinweise.

H4: Erste Schritte mit FOPBot 4 Punkte

Aus der Vorlesung kennen Sie die Welt „FOPBot“. In dieser Hausübung implementieren Sie ein erstes Java-Programm, das einen Roboter erzeugt, durch die Welt bewegt und dabei Münzen ablegen lässt.

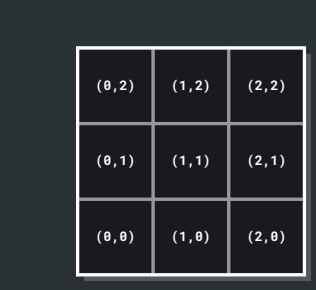
> **Wir verfolgen „Abschreiben“ und andere Arten von Täuschungsversuchen!**
>
> Disziplinarische Maßnahmen treffen nicht nur die, die abschreiben, sondern auch die, die abschreiben lassen! Beachten Sie die Seite *Grundregeln der Wissenschaftsethik* des Fachbereichs Informatik.

> **Hinweis:**
>
> Screenshots aus der Welt der Roboter dürfen Sie unbedenklich mit anderen Studierenden teilen – nur eben nicht den Quelltext oder eine übersetzte Variante des Quelltexts!

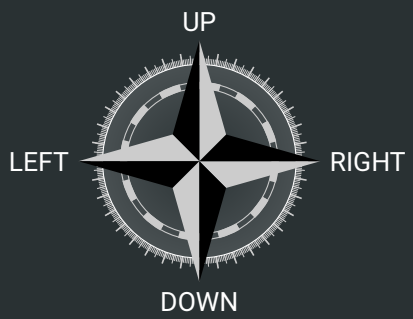
In der Vorlage zu dieser Hausübung haben wir bereits eine Anweisung vorgegeben, die einen Roboter `robby` einrichtet, der in der unteren rechten Ecke der Welt mit Blickrichtung nach unten startet und zwölf Münzen besitzt:

```
1  Robot robby = new Robot(4, 0, DOWN, 12);
```

Zur Erinnerung: Bei „FOPBot“ wird die Position des Roboters in der Welt durch eine X- und eine Y-Koordinate dargestellt. Die Koordinaten beginnen bei 0 und die X-Koordinaten starten von links, die Y-Koordinaten starten von unten.



(a) Koordinaten einer „FOPBot“-Welt



(b) Richtungen in der FOPBot-Welt

Abbildung 1: Erinnerung zur FOPBot-Welt

Ihre Aufgabe besteht nun darin, eine Reihe von Anweisungen zu schreiben, mit denen Sie diesen Roboter drehen, Vorwärtsschritte machen lassen und Münzen ablegen lassen. Diese Anweisungen schreiben Sie direkt hinter die oben zitierte Anweisung, das heißt, in den Zwischenraum hinter dem Semikolon, das die oben zitierte Anweisung abschließt, und vor der schließenden Klammer „}“. Wie in den Folien, sollten auch bei Ihnen nicht mehrere Anweisungen in einer Zeile sein, obwohl der Compiler das ohne Fehlermeldung übersetzen würde und der Ablauf des Prozesses bei Aufruf des Programms auch derselbe wäre. Setzen Sie die Kurzspezifikation „hinter dieser Anweisung“ stattdessen also um als „alle eigenen Anweisungen unter dieser Anweisung, aber vor jeder eigenen Anweisung ein *Return*, also ⏎“.

Entfernen Sie die folgende Zeile, bevor Sie Ihr Programm ausführen und Ihre Lösung abgeben:

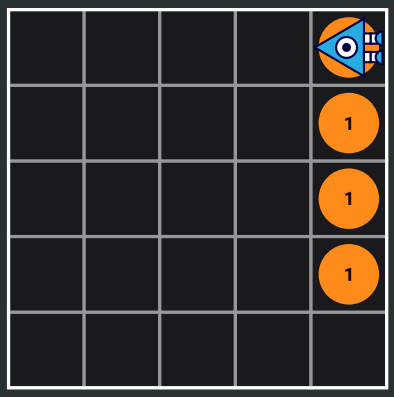
```
1  crash(); // TODO: H4 - remove if implemented
```

Nun zu den konkreten Anweisungen, die Sie zu schreiben haben: Lassen Sie den Roboter `robby` auf seiner Startposition zuerst nach oben drehen, dann an der rechten Wand entlang in die rechte obere Ecke der Welt laufen und dann nach links drehen. Danach lassen Sie ihn stufenartig zur linken unteren Ecke hinabsteigen, das heißt, ein Schritt nach links, dann Linksdrehung, dann einen Schritt hinab, dann Rechtsdrehung, und diese vier Anweisungen wieder von vorne. Unmittelbar nach jedem Vorwärtsschritt lassen sie ihn eine Münze ablegen.

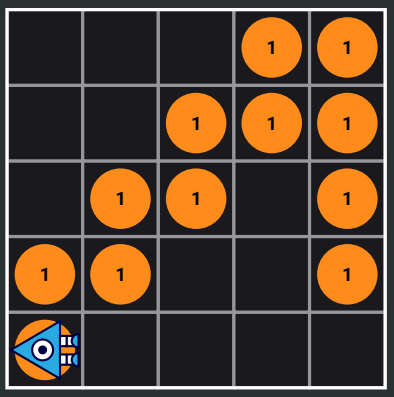
Verbindliche Anforderungen:

1. Für den Weg von der rechten unteren hoch zur rechten oberen Ecke (inklusive Münzen ablegen) schreiben Sie eine `for`-Schleife, die viermal durchlaufen wird. Für den Treppenabstieg (inklusive Münzen ablegen) schreiben Sie eine weitere `for`-Schleife, die ebenfalls viermal durchlaufen wird.
2. Es ist zu jedem Zeitpunkt nur ein Roboter in der Welt (es darf also kein zweiter Roboter erstellt werden).

(a) Startzustand: Der Roboter startet an Position (4,0) mit Blick nach unten.



(b) Zwischenzustand nach der ersten `for`-Schleife und nachfolgender Linksdrehung.



(c) Endzustand nach der zweiten `for`-Schleife.

Hinweise:

1. Sie dürfen in *dieser* Hausübung davon ausgehen, dass die Welt immer die Größe 5x5 besitzt.
2. Implementieren Sie Ihr Programm *zuerst* (beachten Sie *danach* aber die verbindlichen Anforderungen) ohne Schleifen, wenn Sie Schwierigkeiten haben, Schleifen zu verwenden. Ihnen werden sicherlich Wiederholungen auffalen, die Sie in einer Schleife zusammenfassen können.
3. Lassen Sie testweise bei der Entwicklung Ihres Programms nach jeder Anweisung, mit der Sie den Zustand des Roboters ändern, den neuen Zustand ausgeben, um zu überprüfen, ob der neue Zustand des Roboters tatsächlich der jeweils erwartete ist. Dazu können Sie beispielsweise `System.out.print(String)` verwenden. Die Anweisung `System.out.print(robby.getX());` gibt beispielsweise die aktuelle Spaltennummer auf der Konsole aus. Es bietet sich an, für jeden momentanen Zustand des Roboters jeweils alle Daten in einer eigenen Zeile auszugeben und für den nächsten Zustand eine neue Zeile aufzumachen. Nutzen Sie `System.out.println`, wenn Sie nach dem String direkt einen Zeilenumbruch setzen möchten.