

Funktionale und objektorientierte Programmierkonzepte

Übungsblatt 00



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Prof. Karsten Weihe

Übungsblattbetreuer:
Wintersemester 22/23
Themen:
Relevante Foliensätze:
Abgabe der Hausübung:

Ruben Deisenroth
v1.0-SNAPSHOT
Programmieren in Java mit Hilfe von FopBot
01a
28.10.2022 bis 23:50 Uhr

Hausübung 00 Java-Einstieg mit FopBot

Gesamt: 5 Punkte

Beachten Sie die Seite *Verbindliche Anforderungen für alle Abgaben* in unserem 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

Tragen Sie zunächst Ihre Matrikelnummer **vollständig und fehlerfrei** in Moodle ein. Falls Sie das schon getan haben, vergewissern Sie sich **dringend**, dass diese wirklich eingetragen und korrekt ist! Achten Sie insbesondere auf mögliche **Tippfehler** oder **Zahlendreher**!

Anleitung: Unter <https://moodle.informatik.tu-darmstadt.de/user/profile.php> finden Sie Ihr Moodle-Profil. Dort melden Sie sich wie gewohnt mit Ihren Daten an. Dann klicken Sie in der oberen rechten Ecke auf den kleinen Pfeil nach unten und wählen im Dropdown-Menü das Feld *Profil*. Klicken Sie dann auf *Profil bearbeiten*. Navigieren Sie nun zum Feld *ID-Nummer* in der Kategorie *Optionale Einträge* und tragen dort Ihre **7-stellige** Matrikelnummer ein. Bestätigen Sie mit *Profil aktualisieren* und vergewissern Sie sich ein weiteres Mal, dass Ihre Matrikelnummer korrekt in Ihrem Profil eingetragen ist.

Achtung: Die Kategorie *Optionale Einträge* ist hier irreführend! Fehlende oder falsch eingetragene Matrikelnummern können dazu führen, dass wir Ihre erreichte Zulassung und erreichten Bonus sowie Klausurergebnisse nicht richtig verbuchen können!

H2: Einrichten Ihrer Entwicklungsumgebung

Zum Bearbeiten der Hausübungen der FoP benötigen Sie eine Java-Entwicklungsumgebung. Im Moodle-Kurs haben wir Anleitungen zum Einrichten einer solchen Umgebung unter den gängigsten Betriebssystemen zusammengestellt. Sie finden diese im Abschnitt „Informationen zur Software und zu Hausübungen“. Richten Sie sich gemäß den Anleitungen aus dem Moodle-Kurs Ihre Java-Entwicklungsumgebung auf Ihrem Computer ein.

Wie für jede Hausübung stellen wir Ihnen im Moodle-Kurs eine Vorlage zur Verfügung, in die Sie Ihren Quelltext einsetzen. Die Verwendung dieser Vorlage ist verbindlich! Beachten Sie auch alle weiteren Hinweise zu den Abgabemodalitäten der Hausübungen aus dem Abschnitt „Informationen zur Software und zu Hausübungen“ im Moodle-Kurs. **Für Abgaben, die nicht den Abgabevorgaben entsprechen, können keine Punkte vergeben werden!**

Laden Sie nun also die Vorlage zur nullten Hausübung herunter. Die Vorlagen für die weiteren Übungsblätter können dann sobald wie jeweils möglich im Moodle-Kurs im Abschnitt zum jeweiligen Übungsblatt heruntergeladen werden.

H3: 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.

Screenshots der World mit Ihren Robotern darin können Sie unbedenklich mit anderen teilen und in Foren posten, um zu klären, ob Ihr Programm das tut, was es soll. Quelltext und übersetzten Quelltext dürfen Sie selbstverständlich nicht teilen, posten oder sonst wie weitergeben außer an die Ausrichter und Tutor:innen der FoP 22/23!

Wir verfolgen „Abschreiben“ und andere Arten von Täuschungsversuchen. Disziplinarische Maßnahmen treffen nicht nur die, die abschreiben, sondern **auch die, die abschreiben lassen**. Allerdings werden wir nicht unbedingt zeitnah prüfen, das heißt, es hat noch nichts zu bedeuten, wenn Sie erst einmal nichts von uns hören.

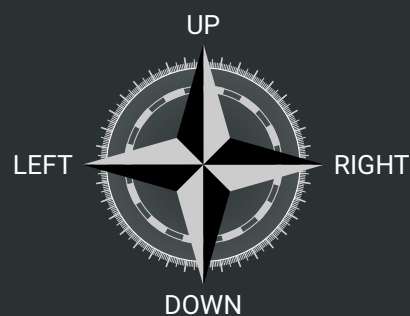
In der Vorlage zu dieser Hausübung haben wir bereits eine Anweisung vorgegeben, die einen Roboter `robby` einrichtet, der in der unteren rechten Ecke der Welt mit Blickrichtung nach unten startet und zwölf Münzen besitzt:

```
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.


(0,2)	(1,2)	(2,2)
(0,1)	(1,1)	(2,1)
(0,0)	(1,0)	(2,0)

(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 .

Den Kommentar „// TODO H00 Implement your solution here“ können Sie löschen oder im Quelltext belassen, das macht keinen Unterschied. Wie Sie allerdings aus Kapitel 01a, Folie 204 ff. wissen, darf keine Anweisung, die Sie gerne ausführen lassen möchten, danach in derselben Zeile kommen, weil alles nach „//“ auskommentiert ist, das heißt, vom Compiler ignoriert und daher auch nicht in auszuführenden Code übersetzt wird.

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:

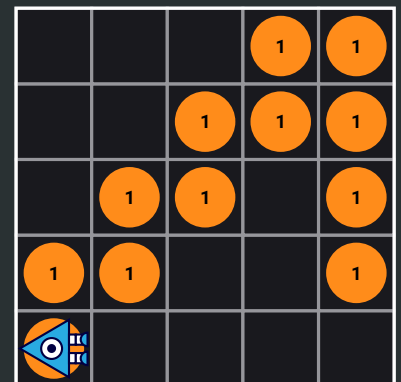
1. Für den Weg von der rechten unteren hoch zur rechten oberen Ecke (inklusive Münzen ablegen) schreiben Sie eine **for**-Schleife, die viermal durchlaufen wird. Für den Treppenabstieg (inklusive Münzen ablegen) schreiben Sie eine weitere **for**-Schleife, die ebenfalls viermal durchlaufen wird.
2. Es 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. Wenn Sie mit der Anforderung, dass die beiden Wege durch Schleifen zu realisieren sind, nicht auf Anhieb klarkommen, dann lösen Sie die Aufgabe zunächst einmal ohne Schleifen. Ihnen werden dann sicherlich Wiederholungen in Ihren Anweisungen auffallen, die Sie dann zu den beiden geforderten Schleifen zusammenfassen können.
3. Lassen Sie sich 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 `System.out.print(...)` und `System.out.println(...)` verwenden. Zum Beispiel schreibt die Anweisung

```
System.out.print(robby.getX());
```

die aktuelle Spaltennummer des Roboters auf die Konsole, analog `getY` und `getNumberOfCoins`. Für die Richtung des Roboters ist es etwas komplizierter: `System.out.print(robby.getDirection().name());` Mit der Anweisung `System.out.print(" ");` zwischen zwei „echten“ Ausgaben können Sie diese durch ein Leerzeichen trennen. Bei `System.out.println(...)` statt `System.out.print(...)` wird am Ende der Ausgabe noch ein Zeilenumbruch hinausgeschrieben (vgl. Kapitel 01b, Folie 182). 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.