



Alexander Mainka, Kim Berninger,
Dustin Glaser, Jan Bormet und
Julius Hardt
(Gesamtleitung: Prof. Dr. Karsten Weihe)

Wintersemester 20/21
v1.0.1

Übungsblatt 0

Themen: Einführung Java, Eclipse und FopBot

Relevante Foliensätze: 01a

Abgabe der Hausübung: 13.11.2020 bis 23:50 Uhr

Wichtiger Hinweis: Es ist immens wichtig, dass Sie sich intensiv und gewissenhaft mit den hier vorgestellten Themen beschäftigen, da sie die Grundlage für die Hausübungsabgaben in diesem Semester darstellen. Auf dieser und der nächsten Seite erhalten Sie einige allgemeine Informationen, diese gelten für alle Hausübungen und wir setzen diese Informationen als gegeben voraus.

Aufbau der Übungsblätter

Die Übungsblätter gliedern sich immer in zwei Teile – die vorbereitenden Übungen **V** und die Hausübungen **H**. In den vorbereitenden Übungen sollen die Konzepte der Vorlesung eingeübt werden. Die Aufgaben sind meist kürzer und sollen Sie praktisch an die Thematik heranführen. Außerdem bilden sie die Grundlage für die Hausübungen. Die Schwierigkeiten der einzelnen Übungen sind durch Sterne dargestellt (mindestens 0 und maximal 3), je mehr Sterne desto komplexer eine Aufgabe. Für ein sicheres Bestehen (das heißt 50 % der Punkte) sollten Sie zumindest die Zwei-Sterne Aufgaben gut meistern können und die Drei-Sterne Aufgaben spätestens nach kleineren Hilfstellungen. Die Hausübungen sind meist etwas schwieriger als die vorbereitenden Übungen und werden am Ende bewertet. Zu den vorbereitenden Übungen werden wir Lösungsvorschläge auf moodle bereitstellen. Die Lösungen der Hausübungen stellen wir Ihnen ebenfalls bereit, jedoch erst nach der Abgabefrist der jeweiligen Hausübung.

Programmieren mit Stift und Papier

Bei Fragen zu den Übungen bieten wir Ihnen unterschiedlichste Hilfsangebote. Beachten Sie jedoch, dass Sie in der Abschlussklausur keine Hilfsmittel zur Verfügung haben werden. Üben Sie also schon zu Beginn auch ohne Entwicklungsumgebung und nur mit Stift auf einem Blatt Papier zu programmieren. Sie lernen das Ganze aber nur wenn Sie sich selbst intensiv mit dem Thema auseinander setzen, nutzen Sie Hilfe also nur wenn es gar nicht anders geht. Nur aus eigenen Fehlern lernen Sie richtig!

Hilfsangebote

- Sprechstunden: Auf der Kursseite finden Sie eine Übersicht über die Sprechstunden, die angeboten werden. Diese können Sie aufsuchen und den Tutorinnen und Tutoren gezielt Fragen zu den Inhalten stellen.
- Forum: In moodle sind Foren für die Übungsblätter eingerichtet worden. Stellen Sie dort Ihre Fragen mit einem möglichst aussagekräftigen Titel, so können auch alle anderen Kursteilnehmer maximal davon profitieren.

Abgabe der Hausübungen

Zur Hausübungsabgabe nutzen wir die entsprechenden Module in moodle. Die Abgabefristen sind auf jedem der Blätter vermerkt, sie liegen jedoch immer Freitagabends um 23:50 Uhr Serverzeit. **Es werden nur Bearbeitungen bewertet, die auf moodle hochgeladen wurden** (keine Repository-Links, keine Dropbox, keine E-Mails etc.). Sollte Ihnen beim Upload der Status „verspätet“ angezeigt werden ist dies kein Problem. Solange Sie etwas auf moodle hochladen können, akzeptieren wir es noch. Halten Sie sich bei der Abgabe der Dateien an die folgenden von uns gegebenen Konventionen! Die Konventionen für Hausübungen in Java finden Sie in dieser Hausübung, die für Racket in Hausübung 5. Bei allen Abgaben handelt es sich um **Einzelabgaben**! Auch wenn sie zusammenarbeiten, müssen sie unterschiedliche und eigen bearbeitete Lösungen abgeben!

Plagiarismus

Der Fachbereich Informatik misst der Einhaltung der Grundregeln der wissenschaftlichen Ethik großen Wert bei. Zu diesen gehört auch die strikte Verfolgung von Plagiarismus. Sollten wir ein Plagiat entdecken, wird dies entsprechend von uns geahndet.

https://www.informatik.tu-darmstadt.de/studium_fb20/im_studium/studienbuero/plagiarismus/plagiarismus_1.de.jsp

Jeder der nachfolgenden Punkte ist vor allem für die FOP relevant, da sie Formen von Plagiarismus darstellen und damit verboten sind:

- Das Abschreiben von Lösungen von Hausübungen und Übernehmen von Lösungs-Code zu Programmieraufgaben.
- Das Übernehmen von fremderdachten Lösungsansätzen ohne korrekte Zitierung.
- Das Ausgeben eigener Hausübungslösungen vor dem Abgabetermin an andere.

Beachten Sie vor allem auch den letzten Punkt. Wenn Sie Ihre Lösung an jemand anderen weitergeben, so machen Sie sich selbst eines Plagiats schuldig und müssen ebenfalls mit Maßnahmen rechnen.

V Vorbereitende Übungen

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.

0.1 Entwicklungsumgebung Eclipse installieren

In dieser ersten vorbereitenden Übung, gehen wir mit Ihnen Schritt für Schritt die Installation von Java und der Entwicklungsumgebung Eclipse durch.

0.1.1 JDK installieren

Zu Beginn installieren wir das sogenannte Java Development Kit (kurz: JDK). Wir verwenden in dieser Veranstaltung OpenJDK 11 von AdoptOpenJDK, da sich dieses bereitgestellte JDK einfach auf allen gängigen Plattformen installieren lässt. Besuchen Sie zunächst folgenden Link: <https://adoptopenjdk.net/>.

Achten Sie nun darauf, dass Ihr Betriebssystem korrekt erkannt wird. Dies ist auszumachen an der Überschrift *Download for....* Ist in der Überschrift ein falsches Betriebssystem genannt, so klicken Sie auf den Button *Other platforms* und wählen Ihr Betriebssystem manuell aus. Wird Ihnen das korrekte Betriebssystem angezeigt, achten Sie darauf, dass der Radiobutton *OpenJDK 11 (LTS)* bei *1. Choose a Version* ausgewählt ist. Bei *2. Choose a JVM* wählen Sie *HotSpot* aus. Den Download des Installers starten Sie dann über den Button *Install JDK*. Anschließend installieren Sie das JDK über den soeben heruntergeladenen Installer.

Sollten Sie eine Linux-Distribution als Betriebssystem verwenden, so bietet es sich an, AdoptOpenJDK über ihr Paketverwaltungssystem zu installieren. Folgen Sie dabei folgender Anleitung: <https://adoptopenjdk.net/installation.html#linux-pkg>. Das zu installierende Paket heißt `adoptopenjdk-11-hotspot`.

Weitere Hinweise zur Installation des Java Development Kits finden Sie auf dieser Seite: <https://adoptopenjdk.net/installation.html>

0.1.2 Eclipse installieren und konfigurieren

Wir verwenden die *Eclipse IDE for Java Developers*. Die aktuelle Version ist Eclipse 2020-09 und lässt sich hier herunterladen: <https://www.eclipse.org/downloads/>.

Die Installation von Eclipse setzt eine Java-Installation voraus. Auf den Poolrechnern ist Eclipse bereits installiert und mit dem Befehl `eclipse &` im Terminal aufrufbar.

Für die Installation von Eclipse findet sich unter folgendem Link eine sehr hilfreiche Bilderstrecke: <https://www.eclipse.org/downloads/packages/installer>

Ist Eclipse erfolgreich installiert, so müssen wir es anschließend so konfigurieren, dass Eclipse unsere vorher installierte JDK verwendet. Dazu starten wir zunächst die Eclipse IDE. Nun wählen wir in der Menu Bar den Eintrag *Window* aus. Anschließend klicken

wir auf *Preferences*. Im soeben neu geöffneten Fenster klappen wir den Eintrag *Java* aus. Danach klicken wir auf *Installed JREs* und vergewissern uns, dass die Checkbox des Eintrags *jdk11-...-hotspot* angekreuzt ist. Anschließend wird mit *Apply and Close* gespeichert.

0.2 Eclipse startklar machen

0.2.1 Importieren von Vorlagen

Von uns bereitgestellte Vorlagen in moodle lassen sich in Eclipse leicht importieren. Laden Sie dazu die Vorlage herunter und speichern Sie sie an einem beliebigen Ort.

Achten Sie darauf, dass das zip-Archiv, welches Sie aus moodle herunterladen, nicht automatisch entpackt wird. Dies passiert beispielsweise in den Standardeinstellungen mit Safari unter macOS. Überführen Sie ggf. im Finder den entpackten Ordner mit einem Rechtsklick und *Compress “<Dateiname>”* zurück in ein zip-Archiv. Um sich bei zukünftigen Hausübungen Arbeit zu sparen, können Sie das automatische Entpacken von Archiven in Safari deaktivieren. Wählen Sie dazu in der Menüleiste *Safari* und öffnen Sie mit *Preferences* die Einstellungen. Entfernen Sie dann ganz unten den Haken bei *Open “safe” files after downloading*.

Machen Sie dann einen Rechtsklick im *Package Explorer* von Eclipse und wählen *Import* aus. Wählen Sie dann *General*, dann *Existing Projects into Workspace*, dann *Select archive file* und dann *Browse*. Wählen Sie nun die heruntergeladene Vorlage im zip-Format aus und drücken Sie *Finish*.

0.2.2 Projekt ausführen

Um nun das importierte Projekt auszuführen, öffnen Sie die Klasse (eine .java Datei in der Ordnerstruktur), in der die Methode `public static void main(String[] args)` zu finden ist (wird von uns angegeben in den Hausübungen). Diese dient als Einstiegspunkt. Mit einem Rechtsklick klicken Sie auf diese Klasse im *Package Explorer* und wählen *Run as* und dann *Java Application* aus, um die Klasse am Einstiegspunkt auszuführen. Das Drücken des grünen Play-Buttons in der oberen Leiste führt die zuletzt ausgewählte Klasse aus, die einen Einstiegspunkt besessen hat (dabei wird der gesamte Workspace betrachtet, also auch andere Projekte). Das heißt besitzt die Klasse, die aktuell im Eclipse-Editor offen ist, einen Einstiegspunkt, so wird diese beim Drücken ausgeführt. Andernfalls wird die Klasse ausgeführt, die zuletzt im Editor offen war und einen Einstiegspunkt hat.

0.2.3 fopbot einbinden

Sie müssen sich nicht um das Einbinden von FopBot bei den Hausübungen kümmern. In den Vorlagen, die wir Ihnen zur Verfügung stellen, ist dies bereits erledigt und Sie können den Code einfach ausführen.

V1 Ist Eclipse startbereit?

☆☆☆

Importieren Sie die Vorlage `v03` aus moodle und führen Sie die Klasse `EclipseReady.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 11 in Verbindung mit AdoptOpenJDK ausgegeben wird.

V2 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.

V3 Quadrat



Öffnen Sie nun die Klasse `Square.java`. Dort finden Sie eine Stelle, welche mit `TODO` 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.

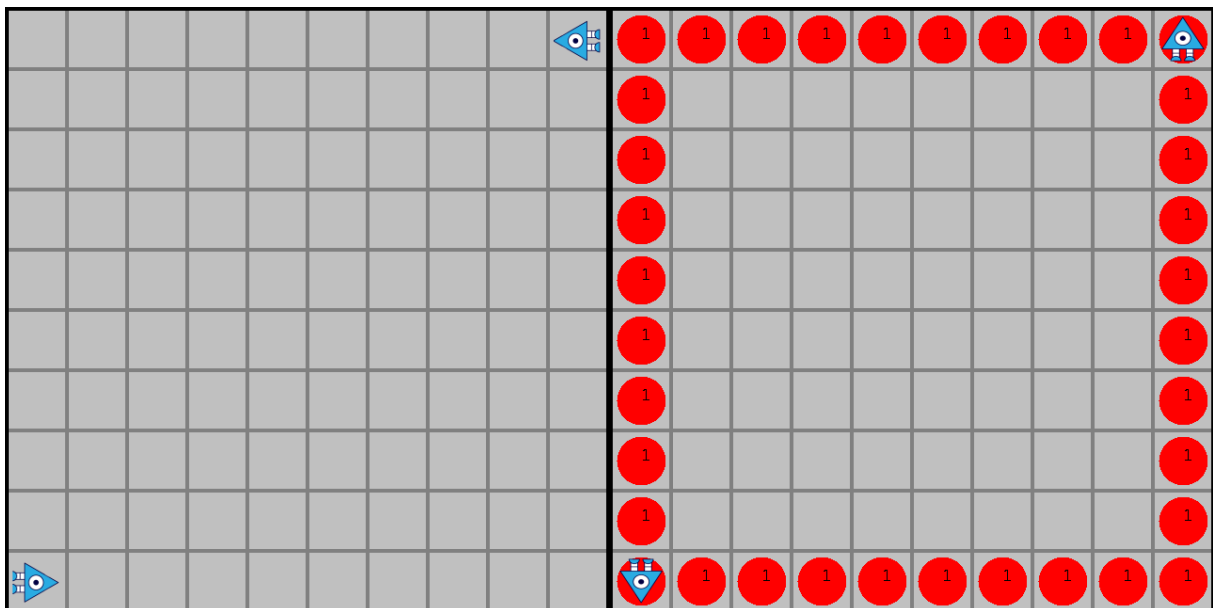


Abbildung 1: Vorher-Nachher-Vergleich

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.

H Nullte Hausübung

Gesamt 5 Punkte

Abgabe von Java-Hausübungen

Für alle Java-Hausübungen geben wir Ihnen Codevorlagen vor, die Sie unbedingt benutzen müssen! Wie Sie diese Vorlagen korrekt importieren können, sehen Sie nochmal in Aufgabe 0.2.1.

Für jedes Übungsblatt geben Sie ein eigenes Java-Projekt mit folgender Namenskonvention ab: `Hnr_ln_fn`. Dabei ersetzen Sie *nr* durch die aktuelle Übungsblattnummer (bei einstelligen Zahlen ergänzen Sie eine führende 0), *ln* durch Ihren Nachnamen und *fn* durch Ihren Vornamen. Ein Projekt für die Abgabe dieser Hausübung kann also z. B. **H00_Mustermann_Max** heißen.

Nach dem Herunterladen der Vorlage wählen Sie die Vorlage im Package Explorer mit einem Rechtsklick aus und drücken erst *Refactor*, dann *Rename* um das Projekt nach der vorgegebenen Namenskonvention zu benennen.

Zur Abgabe exportieren Sie bitte Ihr gesamtes Projekt, indem Sie das Projekt im Package Explorer mit Rechts anklicken und *Export*, dann *General* und abschließend *Archive File* wählen. Wählen Sie alle Inhalte Ihres Projekts aus und benennen Sie das Archiv wie das Projekt nur mit Endung **.zip**. Achten Sie darauf, dass die Optionen *Save in zip format* sowie *Compress the contents of the file* selektiert sind. Geben Sie am Ende genau diese zip-Datei ab. Nutzer anderer IDEs als Eclipse müssen darauf achten, dass sich im zip-Verzeichnis der Projektordner befindet und alle Java-Sourcen im Unterverzeichnis *src* stehen, Eclipse macht das automatisch.

Vorsicht: Bitte denken Sie daran, das Projekt genau wie oben beschrieben umzubenennen! Es genügt **nicht**, nur das am Ende exportierte Zip-Archiv umzubenennen!

Achtung: Wenn Sie eine der oben beschriebenen Anforderungen zur Abgabe verletzen, verlieren Sie automatisch einen Punkt für die jeweilige Hausübung! Haben Sie die oben beschriebenen Anforderungen bei mindestens 2 Hausübungen verletzt, so werden wir Ihre darauffolgenden Abgaben automatisch mit 0 Punkten bewerten, sofern diese zum wiederholten Male (also zum dritten Mal oder häufiger) die Anforderungen zur Abgabe verletzen. Sind Sie sich unsicher, nutzen Sie das Forum um nachzufragen.

Hinweis zu Namen mit Umlauten:

Sollte Ihr Vor- oder Nachname ein Sonderzeichen enthalten, so kann es sein, dass Sie in Eclipse den Fehler `Error: Could not find or load main class Klassenname` erhalten.

Sollten Sie also einen Umlaut oder Sonderzeichen in Ihrem Namen besitzen, so ersetzen Sie diesen am besten (ß zu ss, ä zu ae, ö zu oe und ü zu ue).

H1 Matrikelnummer in Moodle

1 Punkt

Tragen Sie zunächst Ihre Matrikelnummer 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: Sie finden Ihr Moodle-Profil unter
<https://moodle.informatik.tu-darmstadt.de/>

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 Die ersten Roboter

4 Punkte

Importieren Sie zunächst die Vorlage, wie in 0.2.1 beschrieben, und benennen Sie diese entsprechend der Anforderungen in H um.

In dieser Aufgabe werden Sie zwei Roboter implementieren. Als Vorlage dient eine quadratische Welt mit zufälliger ungerader Seitenlänge. Erst wird ein Roboter in der letzten Spalte der mittleren Zeile erstellt. Der Roboter soll anfangs nach oben, also nach Norden, schauen und bekommt genau eine Münze. Dieser läuft dann in die mittlere Spalte der obersten Zeile, legt seine Münze ab, begibt sich zurück auf sein Startfeld und schaut wieder nach oben. Dann wird ein weiterer Roboter erstellt - diesmal in der 0-ten Spalte der mittleren Zeile ohne Münzen, erneut mit Blickrichtung nach oben. Der Roboter bewegt sich dann auf das Feld mit der Münze, sammelt die Münze auf, läuft zurück auf sein Startfeld, legt die Münze dort ab und dreht sich ebenfalls nach oben. Wir implementieren dies in zwei Schritten der folgenden Teilaufgaben. Schreiben Sie Ihren Code nur an Stellen, an denen der Kommentar `//TODO implement` steht. Verändern Sie insbesondere keinen vorhandenen Code.

Verbindliche Anforderung: Alle Roboter dürfen in dieser Aufgabe nur diagonal laufen! Genauer: Ein Roboter darf niemals zwei aufeinanderfolgende Schritte in dieselbe Richtung machen.

Hinweis „Methode“: Für dieses Übungsblatt ist es noch nicht notwendig zu verstehen, was genau eine Methode ist. Sie können das Wort Methode zunächst einmal als eine Stelle im Quellcode auffassen, an die Sie Ihren Code schreiben können. Genaueres zum Thema Methoden erfahren Sie in den Foliensätzen 01b, 01e und 03c.

H2.1 Rechter Roboter

2 Punkte

Implementieren Sie als erstes den rechten Roboter. Schreiben Sie den Code in die Methode `public static void rightRobot(int size)`. Die Variable `size` enthält die Größe der Welt. In Foliensatz 1a wurde Ihnen das Koordinatensystem von FopBot vorgestellt. Denken Sie an die Nummerierung der Zeilen und Spalten jeweils von 0 bis `size - 1`. Erstellen Sie im ersten Schritt das Robot Objekt an der vorgegebenen Stelle mit genau einer Münze und implementieren Sie dann das in H2 beschriebene Verhalten des rechten Roboters. Denken Sie daran, dass Ihre Implementation mit jeder beliebigen ungeraden Weltgröße größer oder gleich 3 zurechtkommen muss!

Verbindliche Anforderung: Schritte (also Aufrufe von `move()`) dürfen nur innerhalb von Schleifen gemacht werden. Verwenden Sie **genau zwei for-Schleifen** und keine anderen Schleifen. Tipp: Gehen Sie pro Schleifendurchlauf einmal horizontal und einmal vertikal. Außerdem darf sich der Roboter nur über Aufrufe von `move()` bewegen. `setX()` und `setY()` sind verboten!

Zur Erläuterung des Verhaltens des rechten Roboters hier einige Bilder mit Beispielgröße 7:

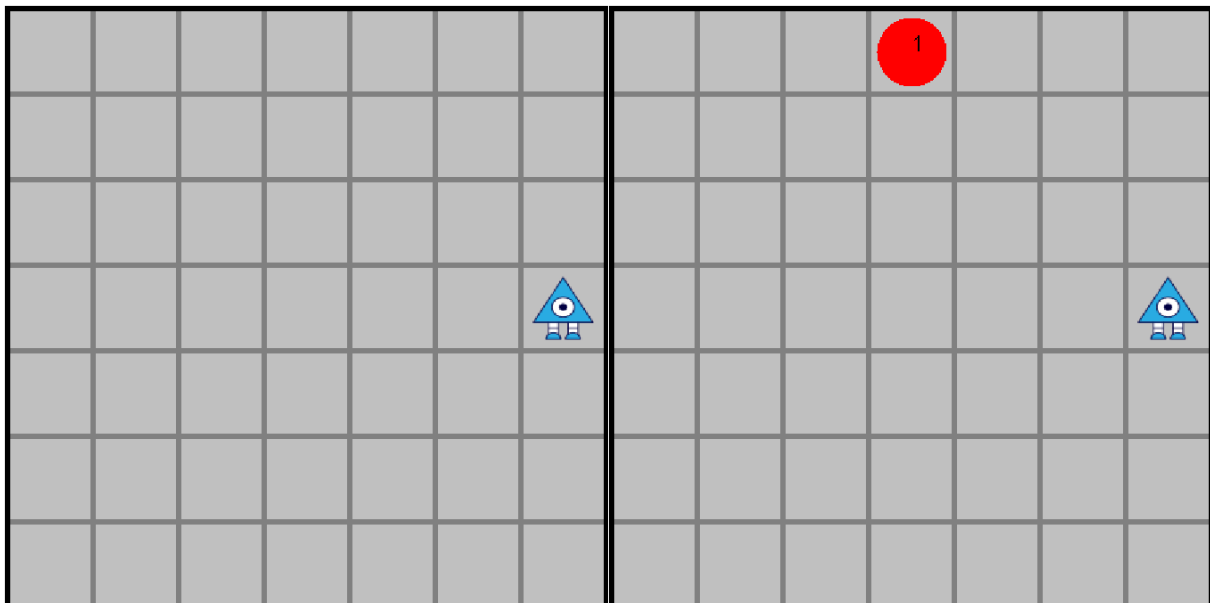


Abbildung 2: Rechter Robot nach dem Erstellen

Abbildung 3: Ende der Methode `rightRobot()`

H2.2 Linker Roboter

2 Punkte

Implementieren Sie als nächstes den linken Roboter. Schreiben Sie den Code in die Methode `public static void leftRobot(int size)`. Die Variable `size` enthält die Größe der Welt. Sie können davon ausgehen, dass der rechte Roboter seine Arbeit bereits korrekt erledigt hat, also die Methode `rightRobot()` vorher bereits aufgerufen wurde und die Münze am richtigen Platz liegt. Implementieren Sie hier also **nur** den linken Roboter!

Erstellen Sie im ersten Schritt das Robot Objekt an der vorgegebenen Stelle ohne Münzen und implementieren Sie dann das in H2 beschriebene Verhalten des linken Roboters. Denken Sie daran, dass Ihre Implementation mit jeder beliebigen ungeraden Weltgröße größer oder gleich 3 zurechtkommen muss!

Verbindliche Anforderung: Schritte (also Aufrufe von `move()`) dürfen nur innerhalb von Schleifen gemacht werden. Verwenden Sie **genau zwei while-Schleifen** und keine anderen Schleifen. Tipp: Gehen Sie pro Schleifendurchlauf einmal horizontal und einmal vertikal.

Außerdem darf sich der Roboter nur über Aufrufe von `move()` bewegen. `setX()` und `setY()` sind verboten!

Zur Erläuterung des Verhaltens des linken Roboters hier einige Bilder mit Beispielgröße 7:

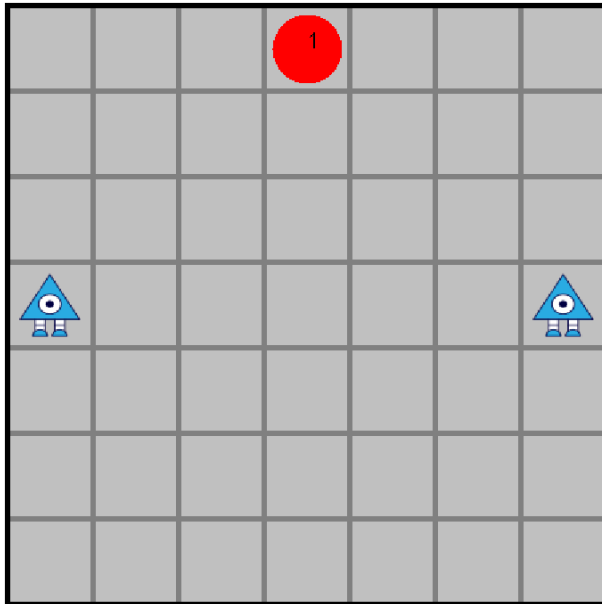


Abbildung 4: Linker Roboter nach dem Erstellen

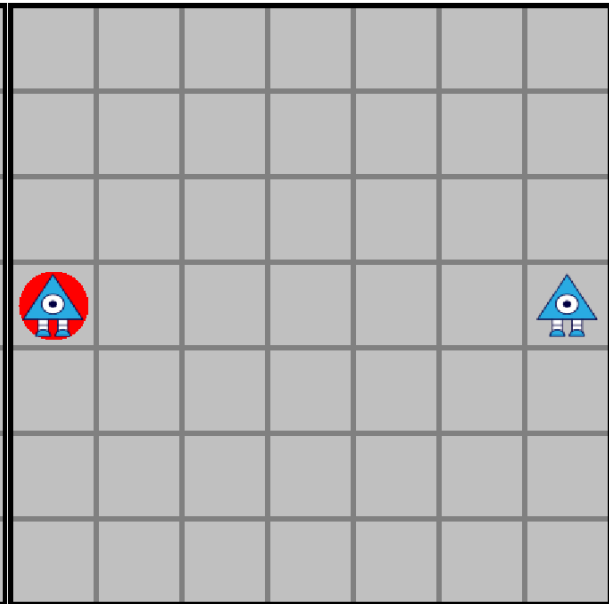


Abbildung 5: Ende der Methode `rightRobot()`