

Funktionale und objektorientierte Programmierkonzepte

Übungsblatt 02



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Entwurf

Achtung: Dieses Dokument ist ein Entwurf und ist noch nicht zur Bearbeitung/Abgabe freigegeben. Es kann zu Änderungen kommen, die für die Abgabe relevant sind. Es ist möglich, dass sich **alle** Aufgaben noch grundlegend ändern. Es gibt keine Garantie, dass die Aufgaben auch in der endgültigen Version überhaupt noch vorkommen und es wird keine Rücksicht auf bereits abgegebene Lösungen genommen, die nicht die Vorgaben der endgültigen Version erfüllen.

Hausübung 02

Cleaning Convoy

Gesamt: 32 Punkte

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/h02` und ggf. `src/test/java/h02`.

Verbindliche Anforderungen für die gesamte Hausübung:

- Die vorgegeben Roboter-Klassen sind wie vorgeschrieben zu verwenden, insbesondere ist die Verwendung der Superklasse `Robot` untersagt.

Hinweise (für die gesamte Hausübung):

- In der `main`-Methode der Klasse `Main` finden sie ein Beispiel zur Verwendung der `ControlCenter`-Klasse.
- Sie können die `main`-Methode beliebig verändern, um Ihren Code zu testen. Sie müssen diese vor der Abgabe nicht zurücksetzen.

Einleitung

Nachdem Sie in der letzten Hausübung selber einen Roboter steuern mussten, um eine Welt aufzuräumen, soll dies in dieser Hausübung vollautomatisch über eine Kolonne an Robotern geschehen, die eine Welt erst auf darin vorhandene Münzen untersucht und diese anschließend automatisch einsammelt.

Hierzu sind zwei Arten an Robotern vorgegeben, die Sie verwenden müssen:

- **ScanRobot:** Diese Roboter dienen zur Erkennung von Münzen, sind jedoch nicht in der Lage, diese aufzusammeln.
- **CleanRobot:** Diese Roboter können Münzen aufsammeln, jedoch nicht feststellen, ob sich auf dem aktuellen Feld eine Münze befindet. Da der Versuch des Aufsammelns ohne Münze auf dem Feld zu einem Crash führt, müssen diese Roboter also genau angesteuert werden, damit dies vermieden wird.

Diese Roboter können Sie bis auf die nicht vorhandene Funktionalität identisch zu normalen Robotern vom Typ Robot verwenden.

Ausblick:

Dies wird durch Vererbung umgesetzt, mit diesem Grundkonzept der objektorientierten Programmierung werden Sie sich in Hausübung 03 befassen. Für unsere Zwecke ist ein tieferes Verständnis nicht notwendig, für interessierte ist der relevante Foliensatz Kapitel 01f der FOP.

Die Kontrolleinheit wird von der `ControlCenter`-Klasse repräsentiert, in der Sie im Verlaufe der Hausübung die Hilfemethoden der Hauptmethode `CleanWorld` implementieren.

Um die Verwaltung der Roboter dynamisch an die Größe der Welt anpassen zu können und einfacher zu gestalten, werden Sie diese im Laufe der Hausübung in Arrays speichern.

H1: Erstellen der Roboter-Arrays**4 Punkte**

Zuerst müssen die Roboter in der Welt platziert werden. Damit später noch auf sie zugegriffen werden kann, sollen ihre Referenzen in einem Array gespeichert werden.

Hinweise:

- Auf die Abmessungen der Welt können sie über die Methoden `getWidth` und `getHeight` der Klasse `World` zugreifen.

H1.1: ScanRobots**3 Punkte**

Implementieren Sie die Methode `initScanRobots`. Diese erhält keine Parameter und soll ein Array an `ScanRobots` zurückgeben, die folgendermaßen zu initialisieren sind:

- Die Roboter sollen die unterste Zeile der Welt bis auf das erste Feld füllen, auf keinem Feld darf sich mehr als ein Roboter befinden.
- Die Roboter sollen alle nach oben ausgerichtet sein.
- Die Roboter besitzen zu Beginn keine Münzen.

H1.2: CleanRobots**1 Punkt**

Implementieren Sie die Methode `initCleaningRobots`. Diese erhält keine Parameter und soll ein Array an `CleanRobots` zurückgeben, die folgendermaßen zu initialisieren sind:

- Die Roboter sollen die erste Spalte der Welt bis auf das erste Feld füllen, auf keinem Feld darf sich mehr als ein Roboter befinden.
- Die Roboter sollen alle nach rechts ausgerichtet sein.
- Die Roboter besitzen zu Beginn keine Münzen.

Verbindliche Anforderungen (für beide Teilaufgaben):

- Die Größe der Arrays muss minimal sein.
- Die Arrays dürfen keine `null`-Einträge enthalten.
- Die Summe der Koordinaten der Roboter im Array muss mit steigendem Index im Array wachsen.

H2: Platzieren der Münzen**3 Punkte**

Implementieren Sie die `void`-Methode `placeCoinsInWorld` in der Klasse `Main`, diese erhält ein Array vom Typ `int`-Array und soll ihrem einzigen Parameter entsprechend Münzen in der Welt platzieren.

Hierbei gilt, dass sich nach dem Methodenaufruf an der Koordinate (i, j) in der Welt genau so viele Münzen befinden, wie es in dem übergebenen Array `coins` am Eintrag `coins[i][j]` angegeben ist.

Hinweis:

Münzen können in der Welt mithilfe der Methode `putCoins` der Klasse `World` platziert werden. Der erste und zweite Parameter sind hierbei die x - und y -Koordinaten und der dritte Parameter ist die Anzahl der zu platzierenden Münzen.

H3: Kolonne marsch!

12 Punkte

H3.1: Scannen der Welt

6 Punkte

<Aufgabentext>

H3.2: Bewegung der Putzkolonne

6 Punkte

<Aufgabentext>

H4: Flip and rotate

10 Punkte

<Aufgabentext>

H4.1: Invertierung der Roboter

3 Punkte

<Aufgabentext>

H4.2: Rotation der Roboter

4 Punkte

<Aufgabentext>

H4.3: Verschleiß behandeln

3 Punkte

<Aufgabentext>

H5: Tests?

3 Punkte

<Aufgabentext>