Algorithmen und Datenstrukturen
Prof. Claudia Fohry
M.Sc. Lukas Reitz

Universität Kassel
Sommersemester 2022
Abgabe: 04.07.2022

Zusatzblatt 7

Auf einem Spielfeld der Größe $n \times n$ sind Edelsteine verteilt, von denen ein Roboter alle Vorkommen einsammeln und zu einer Fabrik bringen soll. An jeder Stelle des Spielfelds können sich keine, ein oder mehrere Edelsteine befinden. Einige Stellen auf dem Spielfeld benötigen mehr Zeit zum Passieren als andere Stellen. Der Roboter steht zu Beginn in der Fabrik auf Position `x=n/2, y=n-1` (Mitte der untersten Zeile) und kann bis zu 3 Edelsteine tragen. Um bis zu 3 Edelsteine einzusammeln, muss der Roboter zu einem oder mehreren Edelsteinen fahren, diese abtragen und anschließend zurück zur Fabrik fahren und dort die gesammelten Edelsteine entladen. Wenn der Roboter seine eingesammelten Edelsteine in der Fabrik entlädt, beginnt sie für $\frac{n}{2}$ Zeiteinheiten die Edelsteine zu verarbeiten. Wird vor Ablauf dieser Zeit erneut ein Edelstein entladen, so muss der Roboter vorher die restliche Zeit abwarten.

Ihre Aufgabe ist es, ein Java-Programm zu schreiben, welches den Roboter möglichst effizient steuert, d.h., dass die benötigte Zeit zum Einsammeln und Verarbeiten aller Edelsteine möglichst gering ist. Sie sollten also bspw. auf geschickte Weise die Wege von der Fabrik zu den Edelsteinen und ggf. auch zwischen den Edelsteinen, sowie die Reihenfolge der einzusammelnden Edelsteine bestimmen.

Vorgegeben ist ein Java-Paket `robot`, welches die Klassen `Robot`, `World` und `Factory` enthält. Sie dürfen keine Klassen dieses Pakets verändern und auch keine neuen Klassen und Pakete in dieses Paket einfügen. Sehen Sie sich diese Klassen genau an. Sie sollten verstehen, wie das Spielfeld generiert wird und wie die drei Klassen miteinander arbeiten. Sie dürfen alle `public`-Methoden verwenden. Falls Sie sich unsicher sind, ob Sie eine der Klassen auf eine gewisse Weise verwenden dürfen, fragen Sie bitte im Moodle-Forum nach. Zusätzlich zum Java-Paket ist in Moodle ein Rahmenprogramm `RobotController.java` vorhanden, welches das Einlesen des Parameters n, die Initialisierung der Spielwelt und die Ausgabe der benötigten Zeiteinheiten beinhaltet. Sie sollen das Rahmenprogramm nur ergänzen.

Sie erreichen 15 Punkte, wenn der Roboter alle Edelsteine einsammelt und in die Fabrik bringt. Die volle Punktzahl wird erreicht, wenn sich der Roboter zusätzlich immer von Fabrik zu den Edelsteinen und zurück, und zwischen den Edelsteinen auf Wegen mit minimaler Fahrtdauer fortbewegt.

(40 Punkte)

Wettbewerb

Alle Programme, bei denen der Roboter alle Edelsteine einsammelt und in die Fabrik bringt, dürfen am Programmierwettbewerb teilnehmen. Das Programm mit der geringsten benötigten Anzahl an Zeiteinheiten wird prämiert! Um am Wettbewerb teilzunehmen, ist eine separate Abgabe erforderlich: Geben Sie dazu Ihr Programm zusätzlich als Aufgabenblatt 13 ab. Genauere Informationen und Regeln für die Teilnahme finden Sie auf der Moodle-Seite.

Abgabe: Beachten Sie die Hinweise zur **Abgabe** am Ende des ersten Aufgabenblatts.