



Technische Universität Darmstadt
 Fachbereich Informatik
 Prof. Dr. Andreas Koch

Allgemeine Informatik 1 im WS 2007/08

Übungsblatt 9

Bearbeitungszeit: 07.01. bis 13.01.2008

Aufgabe 1: Wegweiser

Schreiben Sie eine Roboterklasse, deren Roboter sich selbstständig anhand von Wegweisern auf Kreuzungen durch die Welt bewegen können. Vervollständigen Sie hierzu die auf der Übungswebseite als **uebung09-1.task** vorgegebene Klasse **TourGuide**:

```
class TourGuide extends Robot {

    void turnRight() {
        // siehe uebung09-1.task
    }

    int numBeepers() {
        // siehe uebung09-1.task
    }

    void walk() {
        // ...
    }
}
```

Bei Aufruf der Methode **walk()** soll der Roboter so lange dem durch Wegweiser vorgegebenen Weg folgen, bis eine Kreuzung erreicht wird, die keinen Wegweiser mehr besitzt. Wegweiser werden durch Beeper an Kreuzungen repräsentiert:

- 0 Beeper bedeuten: der Roboter soll sich einen Schritt nach vorne bewegen.
- 1 Beeper bedeutet: der Roboter soll sich nach links drehen und sich anschließend einen Schritt nach vorne bewegen.
- 2 Beeper bedeuten: der Roboter soll sich nach rechts drehen und sich anschließend einen Schritt nach vorne bewegen.
- 3 oder mehr Beeper bedeutet: die Kreuzung besitzt keinen Wegweiser.

Sei beispielsweise vor Aufruf der Methode **walk()** die Welt aus Abbildung 1 gegeben, dann soll der nach Osten schauende Roboter an Position (1, 1) nach Aufruf der Methode **walk()** dem in Abbildung 2 eingezeichneten Weg folgen und an Position (3, 5) stoppen.

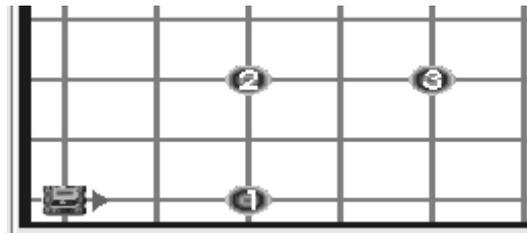


Abbildung 1: Vorgegebene Welt

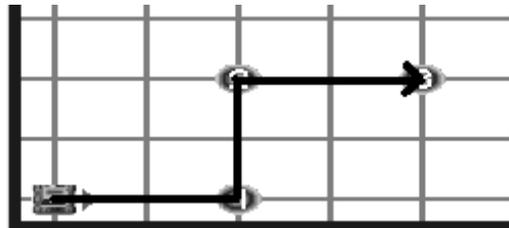


Abbildung 2: Weg in vorgegebener Welt.

Aufgabe 2: Gefangenenlager

Der Roboter **Hans** hat eine Straftat begangen und wurde deshalb in ein Gefängnis gesteckt. Er befindet sich gerade auf dem Außenhof an Position (3, 3) und ihm ist langweilig. Die Gefängnisgrundfläche beschreibt ein Rechteck. An jeder der vier Ecken liegt ein Beeper als Begrenzung.

Hans möchte nun seine neue Heimat besser kennen lernen und will die Größe (Flächeninhalt) des gesamten Lagers wissen. Leider kann er selbst nicht das gesamte Lager von innen vermessen, weil er sich nur auf dem Hof frei bewegen kann.

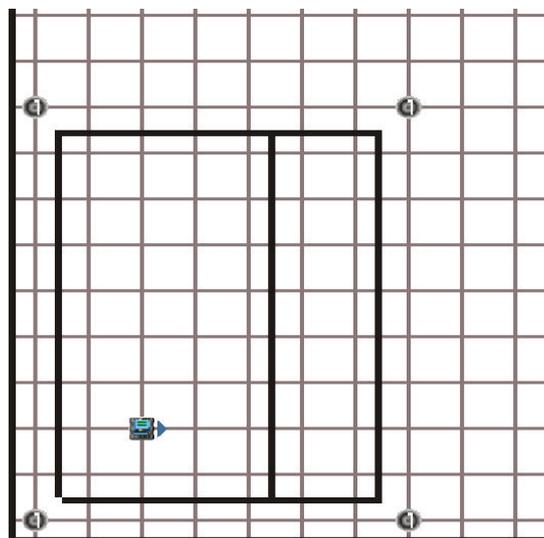


Abbildung 3: Ausgangssituation

Zum Glück hat **Hans** Freunde draußen, mit denen er durch den Zaun kommunizieren kann. Diese Freunde werden für ihn die Länge der Seiten ermitteln und ihm die Zahlen mitteilen, so dass er dann die Fläche ausrechnen kann.

Teilaufgaben

- a) Starten Sie für diese Aufgabe mit dem Quelltext, der für Sie auf unserer Webseite als **uebung09-2.task** bereitgestellt wird. Er enthält 2 Klassen und einen **task**-Bereich.
- b) Die Klasse **Freund** hat nur eine Methode: **int zaehleSchritte()**. Roboter dieser Klasse können die Schritte bis zum nächsten Beeper in Blickrichtung zählen und die Anzahl der Schritte zurückgeben (jedes **move()** entspricht einem Schritt). Programmieren Sie diese Methode!
Achtung: Bedenken Sie, dass ein Roboter auch auf einem Beeper erzeugt werden kann.
- c) Die Klasse **Gefangener** besteht aus zwei Methoden: **int berechneFlaeche (int seiteA, int seiteB), Freund rufeHelfer(direction dir)**. **berechneFlaeche** gibt die Fläche des Rechtecks zurück, dessen Seiten **seiteA** bzw. **seiteB** lang sind. Die Methode **rufeHelfer** erzeugt einen Roboter vom Typ **Freund** an Position (1, 1) und gibt diesen Roboter mittels **return** zurück. Der Roboter soll dabei in die übergebene Richtung **dir** schauen und keine Beeper haben. Implementieren Sie die beiden Methoden.
- d) Das Hauptprogramm soll nur den Roboter **Hans** vom Typ **Gefangener** ohne Beeper und nach Osten schauend an Position (3, 3) erzeugen. Sonst werden keine weiteren Roboter direkt im **task** erstellt.
- e) **Hans** erstellt sich nun zwei Helfer (unter Verwendung der entsprechenden Methode aus Teilaufgabe c), die die Länge je einer Seite des Gefängnisareals messen sollen.
- f) Dann berechnet **Hans** die Fläche des Gefängnisses und gibt sie mittels **System.out.print()** auf dem Bildschirm aus.
- g) Zum Schluss soll sich **Hans** ausschalten.