



Technische Universität Darmstadt
 Fachbereich Informatik
 Prof. Dr. Johannes Fürnkranz

Allgemeine Informatik 1 im WS 2006/07

Übungsblatt 7

Bearbeitungszeit: 11.12. bis 17.12.2006

Aufgabe 1: Wiederholung

- a) Sehen Sie sich die Syntax von
- **if-then-else**-Bedingungen
 - **loop**-Schleifen
 - **while**-Schleifen
- noch einmal im Skript an und schreiben Sie diese auf.
- b) Beschreiben Sie nun in eigenen Worten die drei oben genannten Programmierkonstrukte. Formulieren Sie dabei ganze Sätze!

Aufgabe 2: Polymorphie

Wichtig: Bitte bearbeiten Sie diese Aufgabe per Hand und nicht mit der KarelJIDE – in der Klausur haben Sie auch keinen Rechner!

Die vier Roboter **Karel**, **Adam**, **Ralf** und **Eva** wollen einen Spaziergang machen. Damit sie sich nicht verlaufen, bekommt jeder von ihnen den Weg mitgeteilt:

```
turnLeft();
move();
turnLeft();
move();
turnRight();
move();
```

Das Problem ist, dass jeder der vier Roboter ein wenig anders auf die Wegbeschreibung reagiert.

karel ist vom Typ **RightTurner** (vgl. Aufgabe 3 auf Übungsblatt 6):

```
class RightTurner extends Robot {
    void turnRight() {
        loop(3) turnLeft();
    }
}
```

adam ist vom Typ **FunnyTurner** (vgl. Aufgabe 3 auf Übungsblatt 6):

```
class FunnyTurner extends RightTurner {
    void turnLeft() {
```

```

        move();
        super.turnLeft();
    }
}

```

ralf ist vom Typ **Backwards** (vgl. Aufgabe 3 auf Übungsblatt 6):

```

class Backwards extends RightTurner {
    void move() {
        turnLeft();
        turnLeft();
        super.move();
        turnLeft();
        turnLeft();
    }
}

```

eva ist vom Typ **Runner** (vgl. Aufgabe 3 auf Übungsblatt 6):

```

class Runner extends RightTurner {
    void move() {
        super.move();
        super.move();
    }
}

```

Alle Roboter werden am Anfang auf der Kreuzung (5, 5) abgesetzt und schauen nach Norden. An welcher Kreuzung müssen die Roboter am Ende ihres Spaziergangs jeweils aufgesammelt werden? Bricht ein Roboter mit einem Fehler ab? Bitte füllen Sie die folgende Tabelle aus!

NAME	KLASSE	STREET	AVENUE	RICHTUNG	FEHLER
karel	RightTurner				
adam	FunnyTurner				
ralf	Backwards				
eva	Runner				

Aufgabe 3: Maximum berechnen

Schreiben Sie eine Roboterklasse **Maximizer**, die folgende Methoden enthält:

- **void putAllBeepers() {...}**
Legt alle Beeper, die sich im Korb des Roboters befinden, an der aktuellen Position ab.
- **int numBeepers() {...}**
Soll die Anzahl der Beeper zurückgeben, die an der aktuellen Position liegen. Dazu sammeln Sie alle vorhandenen Beeper ein und zählen dabei in einer Variable mit, wie viele Beeper eingesammelt wurden. Danach legen Sie alle Beeper wieder hin (Sie dürfen voraussetzen, dass der Roboter anfangs keine Beeper hat) und geben dann die Anzahl der gezählten Beeper zurück.
Informationen über Methoden mit Rückgabotyp finden Sie auf Folie 62 im KarelJ-Skript.

Außerdem erweitern Sie bitte die Klasse **Maximizer** um die Methode

- **void max() {...}**
Bei Aufruf dieser Methode soll der Roboter überprüfen auf welcher der Positionen (1, 2) und (1, 3) mehr Beeper liegen und sich auf die Kreuzung stellen, auf der sich mehr Beeper befinden. Dazu sollen Sie die beiden obigen Methoden sinnvoll benutzen. Sie dürfen voraussetzen, dass sich der Roboter bei Aufruf der Methode an Position (1, 1) befindet, keine Beeper hat und nach Osten schaut. Die Welt, in der sich der Roboter bewegt, enthält keine zusätzlichen Wände!

Zum Überprüfen der Korrektheit Ihrer Klasse schreiben Sie einen **task**, in dem Sie mittels **World.placeBeepers(...)** Beeper auf die Kreuzungen (1, 2) und (1, 3) legen, einen nach Osten schauenden Roboter der Klasse **Maximizer** ohne Beeper an Position (1, 1) erzeugen und die Methode **max()** dieses Roboters aufrufen. Versuchen Sie das natürlich mit verschiedenen Anzahlen von Beepern auf den beiden Kreuzungen!

*Hinweis: Wenn es Ihnen sinnvoll erscheint oder Sie möchten, dass Ihre Lösung auch dann noch ohne weitere Änderung funktioniert, wenn der Roboter nicht bei (1, 1) startet, können Sie auch die Methoden der Klasse **EnhancedRobot** aus Aufgabe 2 von Übungsblatt 5 verwenden.*