

SE1, Aufgabenblatt 1

Softwareentwicklung I – Wintersemester 22/23

Karel The Robot: Zusammengesetzte Befehle, Zählschleifen, Fallunterscheidung

Moodle-URL:

<https://uhh.de/min-inf-se1>

Ausgabedatum:

19.10.2022

Kernbegriffe

Karel The Robot ist eine auf die Lehre optimierte Programmiersprache. Sie ist bewusst einfach gehalten und leicht zu erlernen. Der Roboter Karel kennt nur sechs *elementare Befehle*:

```
moveForward();    // Karel geht einen Schritt nach vorne
turnLeft();       // Karel dreht sich um 90 Grad gegen den Uhrzeigersinn
turnAround();     // Karel dreht sich um 180 Grad
turnRight();      // Karel dreht sich um 90 Grad mit dem Uhrzeigersinn
pickBeeper();     // Karel hebt einen Diamanten auf
dropBeeper();     // Karel legt einen Diamanten ab
```

Es ist möglich, beliebig viele Befehle zu einem *zusammengesetzten Befehl* zu kombinieren. Wenn es den elementaren Befehl `turnRight` nicht gäbe, könnte man z.B. einen eigenen Befehl dafür schreiben:

```
void turnright() // Nach rechts drehen soll bedeuten:
{
    turnLeft();  // Zuerst nach links drehen
    turnLeft();  // Dann noch einmal nach links drehen
    turnLeft();  // Und zuletzt nochmal nach links drehen
}
```

Um Befehle mehrfach auszuführen, stehen Karel *Zählschleifen* zur Verfügung:

```
repeat (3)
{
    turnLeft();
}
```

Mit der *Fallunterscheidung* kann Karel Befehle abhängig von einer bestimmten *Bedingung* ausführen:

```
if (frontIsClear()) // Falls momentan vorne frei ist
{
    moveForward();   // Gehe einen Schritt nach vorne
}
```

Karel kann alternativ auch andere Befehle ausführen, wenn die Bedingung *nicht* erfüllt ist:

```
if (onBeeper())      // Falls Karel momentan auf einem Diamanten steht
{
    pickBeeper();     // Dann soll Karel den Diamanten aufheben
}
else                 // Ansonsten
{
    dropBeeper();     // Soll Karel einen Diamanten ablegen
}
```

Karel kennt fünf *elementare Bedingungen*:

```
onBeeper()          // Steht Karel momentan auf einem Diamanten?
beeperAhead()        // Befindet sich direkt vor Karel momentan ein Diamant?
leftIsClear()        // Ist links von Karel momentan keine Wand?
frontIsClear()       // Ist direkt vor Karel momentan keine Wand?
rightIsClear()       // Ist rechts von Karel momentan keine Wand?
```




Bedingungen können mit `&&` (*und*) bzw. `||` (*oder*) verknüpft sowie mit `!` (*nicht*) negiert werden:

```
if (onBeeper() || frontIsClear() && !beeperAhead()) { ... }
```

Dabei bindet `&&` stärker als `||`, ähnlich wie die Punkt-vor-Strich-Regel.

Informationen zur Übung

Lest euch den Text *Übungsablauf* unter dem Abschnitt *Übung* in Moodle durch.

- Aufgaben, die mit dem  Icon gekennzeichnet sind, bearbeitet ihr eigenständig in Moodle. Die Frist zur Bearbeitung der Moodle-Aufgaben ist der Mittwoch nach dem Ausgabedatum um 23:59 Uhr.
- Aufgaben, die mit dem  Icon gekennzeichnet sind, bearbeitet ihr in der Präsenzübung.
- Aufgaben, die mit dem  Icon gekennzeichnet sind, müsst ihr nicht bearbeiten. Dies sind Extraaufgaben, die ihr freiwillig bearbeiten könnt.



1.1 Aufgaben im Moodle

- 1.1.1 Lest euch die Textseite *Kriterien zum Bestehen der Übung* im Abschnitt *Übungen* durch und beantwortet die Moodle-Aufgabe *SE1 Moodle-Aufgabe 1.1.1 - Übungskriterien* in Semesterwoche 1. Um eine Moodle-Aufgabe zu bestehen, müsst ihr alle Fragen richtig beantworten (100% erreichen)!
- 1.1.2 Bearbeitet die Aufgaben in *SE1 Moodle-Aufgabe 1.1.2*.



1.2 Hollywood und Sprengstoff

- 1.2.1 Bevor es losgeht, solltest du Karel bereits installiert haben. Die Anleitung dazu findest du im Moodle unter *SE1 Karel Installationsanleitung* (Semesterwoche 0). Falls es Probleme bei der Installation gab, wende dich bitte an die Tutor*innen.
- 1.2.2 *saveBat* — Karel soll eine verletzte Fledermaus (grüner Diamant) aus einer Höhle retten. Wählt aus der Liste das Problem *saveBat* und klickt auf *goal*. Wenn du den Schieberegler nach *rechts* schiebst (oder ganz oft *step into* anklickst bzw. *F12* drückst), kannst du sehen, wie Karel sich (um die Wand herum) zur Fledermaus bewegt und diese zurück zur Startposition bringt. Wenn du den Schieberegler ganz nach *links* schiebst, kannst du den Einzelschrittmodus mit *F12* bedienen. Deine Aufgabe ist es nun, diesen Ablauf zu programmieren:

```
void saveBat ()
{
    // Befehle, die Karel veranlassen, die Fledermaus zu retten
}
```

Hinweis: Achtet darauf, dass der Name des Programmes im Editor (hier also *saveBat*) mit dem Namen in dem Auswahlménü übereinstimmt.

- 1.2.3 *defuseOneBomb* — Ihr solltet euren Code von Aufgabe 1.2.2 nicht löschen, schreibt den Code der nächsten Aufgabe einfach direkt darunter. Karel macht eine Ausbildung zum Sprengstoffexperten und muss dafür eine Bombe an der gegenüberliegenden Wand entschärfen. Anschließend kehrt er zu seiner Ausgangsposition zurück und betrachtet voller Stolz die Wand, die er soeben vor einer heftigen Detonation bewahrt hat. Öffne das Problem *defuseOneBomb* in der Liste. Bei der Lösung dieser Aufgabe wirst du wahrscheinlich merken, dass ein- und dieselbe Sequenz von Befehlen zweimal in deiner Lösung auftaucht. Diese Sequenz von Befehlen löst ein Teilproblem. Führe einen neuen Befehl dafür ein und wähle einen sinnvollen Namen.
- 1.2.4 Lasst euch jetzt Aufgabe 1.2 abnehmen. Geht hierzu im Moodle auf die *Übungsraumtafel* und tragt euch dort ein. Erklärt eurem*r Tutor*in die Oberfläche von Karel (Bereiche und Knöpfe) und führt Eure Lösungen im Einzelschrittmodus vor.



1.3 Sport und Gartenarbeit

Wieder solltet Ihr den Code der vorigen Aufgabe nicht löschen, sondern einfach im Editor unter den bisherigen Code schreiben.

- 1.3.1 *climbTheStairs* — Karel fragt sich auf dem Heimweg, ob er eine Sportkarriere starten sollte. Dass ausgerechnet jetzt der Fahrstuhl in seinem Appartement den Geist aufgibt, scheint göttliche Fügung zu sein. Von nun an heißt es: Treppen steigen.
- 1.3.2 *fillTheHoles* — Karel macht eine Ausbildung zum Zahnarzt und füllt vier kariöse Zähne mit Amalgam.
- 1.3.3 *mowTheLawn* — Karel hat seinem Opa versprochen, im Garten auszuhelfen. Zum Glück hat er schon das ganze Unkraut gejätet, und er muss sich nur noch um den Rasen kümmern.



1.4 Landwirtschaft und Schlaglöcher

- 1.4.1** *harvestTheField* — Karels Tante ist Landwirtin und hat ein Problem: Der Weizen muss geerntet werden. Nachdem Karel seinem Opa geholfen hat, kann er den Wunsch seiner Tante nicht ausschlagen.
- 1.4.2** *repairTheStreet* — Karel probiert eine neue Jogging-Strecke aus und ist entsetzt: Das Pflaster befindet sich in einem desaströsen Zustand! Zum Glück hat er genügend Zement dabei, sodass er alle Löcher stopfen kann.
Achtung: Das Problem *repairTheStreet* wird zufällig aufgebaut, die Löcher sind bei jedem Durchgang woanders platziert. Hier benötigst du zum ersten Mal die Fallunterscheidung mit `if` bzw. `if/else`.
- 1.4.3** *cleanTheRoom* — Heute Abend steht Besuch an, doch in Karels Wohnung sieht es aus als hätte er eine Bombe im Wohnzimmer explodieren lassen. Da hilft nur eins: Weg mit dem Müll — und zwar systematisch und effizient.
Auch dieses Problem wird zufällig aufgebaut. Der Müll ist bei jedem Durchgang anders platziert.

Herzlichen Glückwunsch, du hast für dieses Aufgabenblatt alle Aufgaben erledigt! Die jeweils letzte Aufgabe eines Aufgabenblatts muss nicht mehr abgenommen werden, sollte jedoch bearbeitet werden. Sofern genügend Tutor*innen zur Verfügung stehen, kannst du sie auch gerne abnehmen lassen. Du solltest dich mit der letzten Aufgabe beschäftigen. Zum einen werden hier noch einmal Inhalte wiederholt, zum anderem ist diese Aufgabe auch klausurrelevant.



1.5 Olympia und Labyrinth und andere Aufgaben

- 1.5.1** *defuseTwoBombs* — Eine Bombe ist offenbar keine Herausforderung für Karel. Aber schafft er auch zwei Bomben?
- 1.5.2** *practiceHomeRun* — Karel ist begeisterter Baseball-Fan. Aber hat er selbst auch das Zeug zum Star? Eines nachts bricht er heimlich in das Stadion seines Lieblingsvereins ein und probiert einen Homerun aus.
- 1.5.3** *saveTheFlower* — Beim Urlaub in den Alpen findet Karel eine seltene Blume, die nicht so richtig aufblühen mag. Der Luftdruck scheint zu hoch zu sein. Auf der Bergspitze herrscht hoffentlich ein günstigeres Klima.
- 1.5.4** *tileTheFloor* — Beim Bummeln durch den Baumarkt findet Karel tolle Fliesen. Voller Vorfreude läuft er nach Hause und legt sie vorerst zum Trocknen im Badezimmer aus. Ob Karel langfristig damit glücklich wird? Er beschließt, erst Mal eine Nacht drüber zu schlafen...
Hier bietet sich eine Zählschleife an, um alle Fliesen (wie viele?) abzulegen.
- 1.5.5** *stealOlympicFire* — Karel ist vom Olympia-Fieber besessen und stiehlt in seinem Fieberwahn das olympische Feuer.
- 1.5.6** *removeTheTiles* — Karel mag die Fliesen in seinem Badezimmer doch nicht mehr und entfernt sie deshalb wieder.
- 1.5.7** *walkTheLabyrinth* — Karel bahnt sich seinen Weg durch ein Labyrinth und findet am Ende einen Diamanten. Schau dir das Labyrinth genau an. Es enthält weder Kreuzungen noch Sackgassen. Karel muss sich deshalb nicht zwischen mehreren möglichen Wegen entscheiden oder beschriftete Wege zurückgehen. Das vereinfacht die Lösung.