

# E1b-PRP1, Aufgabenblatt 1

Programmieren 1 – Wintersemester 2015/16

## Karel The Robot: Zusammengesetzte Befehle, Zählschleifen, Fallunterscheidung

Ausgabewoche: ..... 28. September 2015

### Kernbegriffe

*Karel The Robot* ist eine auf die Lehre ausgerichtete Programmiersprache, die bewusst einfach gehalten und leicht zu erlernen ist. 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 bei der Punkt-vor-Strich-Rechnung aus der Grundschule.

## Aufgabe 1.0 Karel starten

Diese Aufgabe muss nicht abgenommen werden. Bei Problemen stehen die Betreuer immer bereit.

- 1.0.1 Setze dich zusammen mit einem Partner an einen Labor-Rechner und logge dich in Windows ein.
- 1.0.2 Starte einen Internet-Browser und gehe auf die Seite <https://github.com/frectures/pr1>  
Gehe in den Ordner `termin1`, lade die Datei `karel2015.zip` herunter und entpacke sie.
- 1.0.3 Starte die Datei `runme-windows.bat` per Doppelklick. Falls du auf deinem eigenen Rechner Linux verwendest oder einen Macintosh, verwende stattdessen die Datei `runme-linux.sh`

## Aufgabe 1.1 Hollywood und Sprengstoff

- 1.1.1 Karel wurde als potenzieller Nachfolger von Harrison Ford für den nächsten Indiana-Jones-Film ausgewählt. Um sein schauspielerisches Talent zu beweisen, stellt er die klassische Szene nach, in welcher Indy das wertvolle Artefakt aus der Indianerhöhle rettet.

Wähle aus der Liste der Probleme das Problem 1.1.1 *obtainArtefact* und klicke 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) zum Artefakt bewegt und dieses zurück zur Startposition bringt. Deine Aufgabe ist es nun, dieses Verhalten nachzuprogrammieren:

```
void obtainArtefact()
{
    // Befehle, die Karel veranlassen, das Artefakt zurückzubringen
}
```

- 1.1.2 Karel macht eine Ausbildung zum Sprengstoffmeister 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.

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 kleineres Problem. Führe einen neuen Befehl dafür ein und wähle einen sinnvollen Namen.

- 1.1.3 Eine Bombe ist offenbar keine Herausforderung für Karel. Aber schafft er auch zwei Bomben?
- 1.1.4 Karel ist begeisterter Baseball-Fan. Aber hat er selber auch das Zeug zum Star? Eines Nachts bricht er heimlich in das Stadion seines Lieblingsvereins ein und probiert einen Homerun aus.  
(Hier lohnen sich wahrscheinlich keine neuen Befehle für Teilprobleme; kann man aber machen.)
- 1.1.5 Hebe eine Hand und sage einem Betreuer „Wir möchten uns Aufgabe 1.1 abnehmen lassen“. Erkläre ihm die Oberfläche von Karel (was bedeuten die verschiedenen Bereiche und Knöpfe) und führe ihm deine Lösungen im Einzelschrittmodus vor.

Falls momentan kein Betreuer verfügbar ist, schreibe deinen Namen und den Namen deines Partners gut lesbar links oben an die Tafel unter ggf. bereits dort vermerkte Namen. Es wird sich dann so schnell wie möglich jemand um dich kümmern. In der Zwischenzeit kannst du gerne schon mit der Bearbeitung der Aufgabe 1.2 anfangen.

Verfahre analog am Ende von Aufgabe 1.2.

## Aufgabe 1.2 Sport und Gartenarbeit

- 1.2.1 Vom Homerun noch voller Glücksgefühle fragt sich Karel auf dem Heimweg, ob er seine Sportkarriere weiter vorantreiben sollte. Dass ausgerechnet jetzt der Fahrstuhl in seinem Appartement den Geist aufgibt, scheint göttliche Fügung zu sein. Von nun an heißt es: Treppen steigen!
- 1.2.2 Karel macht eine Ausbildung zum Zahnarzt und füllt vier kariöse Zähne mit Amalgam.
- 1.2.3 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!
- 1.2.4 Karel hat seiner Tante versprochen, im Garten auszuhelfen. Zum Glück hat sie schon das ganze Unkraut gejätet, und er muss sich nur noch um den Rasen kümmern.

## Aufgabe 1.3 Landwirtschaft und Schlaglöcher

- 1.3.1 Karels Onkel ist Landwirt und hat ein Problem: Der Weizen ist reif und muss geerntet werden! Nachdem Karel seiner Tante geholfen hat, kann er seines Onkels Wunsch nicht ausschlagen...

- 1.3.2 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, um alle Löcher zu stopfen.  
**Achtung:** Das Problem `repairTheStreet` wird zufällig aufgebaut, die Löcher sind also jedes Mal woanders. Hier benötigst du zum ersten Mal die Fallunterscheidung mit `if` bzw. `if/else`.
- 1.3.3 Heute Abend steht Besuch an, doch in Karels Wohnung sieht es aus wie bei Hempels unterm Sofa! Da hilft nur eins: Weg mit dem Müll, und zwar systematisch und effizient.  
Auch dieses Problem wird zufällig aufgebaut, der Müll ist also jedes Mal woanders.
- 1.3.4 Beim Bummeln durch den Baumarkt findet Karel einen Satz von Fliesen, die er unwiderstehlich findet. Voller Vorfreude läuft er nach Hause und legt sie vorerst trocken 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.

#### Aufgabe 1.4 Olympia und Labyrinth

Herzlichen Glückwunsch, du hast alle scheinrelevanten Aufgaben erledigt! Die jeweils letzte Aufgabe eines Aufgabenblatts muss nicht mehr abgenommen werden. Sofern genügend Betreuer zur Verfügung stehen, kannst du sie aber gerne abnehmen lassen. So oder so solltest du dich mit der letzten Aufgabe beschäftigen, denn hier werden noch einmal Inhalte wiederholt, und klausurrelevant ist sie auch.

- 1.4.1 Karel ist vom Olympia-Fieber besessen und stibitzt in seinem Fieberwahn das olympische Feuer.
- 1.4.2 Karel mag die Fliesen in seinem Badezimmer doch nicht mehr und entfernt sie deshalb wieder.
- 1.4.3 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 niemals zwischen mehreren möglichen Wegen entscheiden oder beschrittene Wege zurückgehen. Das vereinfacht die Lösung enorm.  
(Wir werden uns später noch mit interessanteren Labyrinthen befassen.)