



Technische Universität Darmstadt
 Fachbereich Informatik
 Prof. Dr. Johannes Fürnkranz

Allgemeine Informatik 1 im WS 2006/07

Übungsblatt 2

Bearbeitungszeit: 06.11. bis 12.11.2006

Aufgabe 1: Bedienung von KarelJIDE

Laden Sie sich, falls noch nicht geschehen, von der Webseite auch die Datei **uebung02.task** herunter (<http://www.ke.informatik.tu-darmstadt.de/lehre/ws0607/ai1/uebungen.php>).

Starten Sie dann **KarelJIDE** und laden Sie mittels des Buttons **Open** diese Datei in den Editor.

Klicken Sie einmal links in das Editorfenster. Es sollte ein kleiner, senkrechter, blinkender Strich zu sehen sein. Dieser nennt sich **Cursor** und markiert die Stelle, an der Sie sich in der Datei befinden (und wo Texteingaben eingefügt werden).

Wir wollen nun grundlegende Editortechniken kennen lernen, die auch beim Programmieren häufig sehr hilfreich sind. Im Code stehen zwei Kommentarzeilen, erkennbar durch `//` am Anfang der Zeile.

Markieren Sie die erste dieser Zeilen, indem Sie mit der Maus am Anfang der Zeile klicken, den Mauszeiger zum Ende der Zeile bewegen und die Maustaste dann loslassen. Sie können auch den Cursor mit der Maus oder den Pfeiltasten der Tastatur an den Anfang der Zeile bewegen und dann die **SHIFT**-Taste¹ gedrückt halten, während Sie den Cursor ans Ende der Zeile bewegen.

Die Zeile selbst gibt Ihnen die nächste Anweisung: sie will nach Zeile 13 kopiert werden. Zum **Kopieren** des markierten Texts in die Zwischenablage drücken Sie **CTRL+C**². Bewegen Sie dann den Cursor an den Anfang von Zeile 13 (rechts unterhalb des Weltfensters werden Zeile und Spalte als „Line: y, Column: x“ angezeigt). Zum **Einfügen** des Texts aus der Zwischenablage drücken Sie nun **CTRL+V**.

Die zweite Kommentarzeile möchte gerne an einer anderen Stelle stehen. Dazu markieren Sie die Zeile und drücken dann **CTRL+X** zum **Ausschneiden**. Der Text ist nun zwar nicht mehr zu sehen, aber in der Zwischenablage noch vorhanden. Daher können Sie ihn nun in Zeile 2 wie gehabt mit **CTRL+V** wieder einfügen.

Aufgabe 2: Fehler in Programmen

Um die Bedeutung der häufigsten Fehlermeldungen kennen zu lernen, werden in dieser Aufgabe absichtlich Fehler in ein (zunächst fehlerfreies) Programm eingefügt. Sie sollten die entstandenen Fehlermeldungen genau studieren und versuchen, diese mit den vorgenommenen Änderungen in Zusammenhang zu bringen.

Geben Sie das folgende Programm ein und speichern Sie es unter einem beliebigen Namen. Stellen Sie sicher, dass das Programm fehlerfrei ist, indem Sie das Programm ausführen. Falls Sie alles richtig eingegeben haben, sollte ein kleiner Roboter erzeugt werden, der zwei Schritte nach rechts macht.

Hinweis: Sie müssen nach jeder Änderung das Programm speichern. Ansonsten wird die Änderung nicht aktiv und es wird keine Fehlermeldung erzeugt. Die Fehlermeldungen werden im unteren

¹ Der Pfeil nach oben auf der linken Seite der Tastatur

² Dazu halten Sie erst die Taste **CTRL** (auf deutschen Tastaturen **STRG**) und drücken dann dazu die Taste C.

Bereich von KarelJIDE angezeigt. Nach jeder Teilaufgabe müssen Sie die Änderungen der Teilaufgabe wieder rückgängig machen, bevor Sie die nächste Teilaufgabe angehen!

```
task {  
    Robot karel = new Robot(1, 1, 0, East);  
    karel.move();  
    karel.move();  
}
```

- a) Entfernen Sie bei einer der **move**-Anweisungen das Semikolon.
- b) Ändern Sie bei einer der **move**-Anweisungen **karel** nach **karl**.
- c) Entfernen Sie die öffnende geschweifte Klammer.
- d) Entfernen Sie die schließende geschweifte Klammer.
- e) Ersetzen Sie bei einer der **move**-Anweisungen den Punkt durch ein Leerzeichen.
- f) Ersetzen Sie bei einer der **move**-Anweisungen den Punkt durch ein Komma.
- g) Ersetzen Sie bei einer der **move**-Anweisungen den Punkt durch ein Semikolon.
- h) Entfernen Sie in einer der **move**-Anweisungen das Klammerpaar.
- i) Ändern Sie in einer der **move**-Anweisungen das **move** in **mowe**.
- j) Entfernen Sie beim Erzeugen von **karel** das letzte Komma und die Himmelsrichtung East.

Versuchen Sie eigene Fehlermeldungen zu produzieren. Führen Sie insbesondere mehrere der obigen Änderungen gleichzeitig ein und kontrollieren Sie die produzierten Fehlermeldungen.

Sie sollten nach dieser Aufgabe in der Lage sein, Fehler wie z.B. fehlende Semikolons selbst zu erkennen und zu korrigieren (ohne Ihren Tutor fragen zu müssen)!

Aufgabe 3: Formatierung von Programmen

Das folgende Programm enthält einen Syntaxfehler. Können Sie den Fehler finden?
Ändern Sie die Formatierung so wie in der Vorlesung beschrieben. Können Sie jetzt den Syntaxfehler finden?

```
task { Robot karel = new Robot(3, 2, 1, West)  
;karel.turnLeft();karel.turnLeft();karel.move()  
karel.move();karel.move();karel.move();karel  
.putBeeper(); }
```

Aufgabe 4: Quadrat

Schreiben Sie ein Programm, welches zwei Roboter **karel** und **karl** erzeugt.
karel soll mit Hilfe von Beepern ein Quadrat mit der Kantenlänge 4 erzeugen (d.h., jede Kante soll aus 4 Beepern bestehen) und **karl** soll, nachdem das Quadrat erzeugt wurde, alle Beeper wieder einsammeln.

*Hinweis: Überlegen Sie sich zuerst, wie viele Beeper **karel** benötigt um das Quadrat zu erzeugen.*

Sollten Sie mit den Aufgaben 2-4 auf diesem Übungsblatt nicht weiterkommen, lesen Sie noch mal das „Karel J. Robot“-Skript bis Folie 30. Dieses bekommen Sie auf unserer Webseite:

<http://www.ke.informatik.tu-darmstadt.de/lehre/ws0607/ai1/material.php>