

# 1 Karol stellt sich vor

## 1.1 Algorithmus

Fritz hat zum Geburtstag einen CD-Player als Geschenk erhalten. Natürlich will er sofort das Geschenk ausprobieren und legt seine Lieblings-CD ein. Allerdings gefallen ihm nicht alle Lieder auf dieser CD. Glücklicherweise findet er in der Gebrauchsanweisung folgenden Abschnitt:

### Programmierung einer Titelfolge

Es können bis zu 20 Titel in beliebiger Reihenfolge abgespeichert werden. Die Eingabe einer Programmierung kann nur in der Stopp-Position vorgenommen werden.

1. Legen Sie eine CD ein.
2. Drücken Sie den Programmknopf.
3. Wählen Sie mit den Skip-Knöpfen den ersten zu speichernden Titel aus.
4. Die Speicherung wird durch Drücken des Programmknopfes vorgenommen.
5. Wiederholen Sie nun die Punkte 3 und 4 solange, bis alle gewünschten Titel abgespeichert sind.
6. Ist der Speicher voll, erscheint "----" im Display.
7. Um die Wiedergabe zu starten, drücken Sie den Wiedergabe/Pause-Knopf. Die Nummer des jeweiligen Titels erscheint im Display, die Programm- und Wiedergabeanzeigen leuchten.

### Übung 1.1

Beurteile diese Gebrauchsanweisung.

- a) Beschreibe die Merkmale einer guten Gebrauchsanweisung.
- b) Wie ist eine Gebrauchsanweisung gegliedert?
- c) Finde immer wieder auftauchende Schlüsselbegriffe.
- d) Suche zu Hause Gebrauchsanweisungen, die deiner Meinung nach gut bzw. schlecht sind.

### Übung 1.2

Falte aus Papier ein Flugzeug, ein Schiff, eine Blume o.ä. und formuliere den Faltvorgang schriftlich. Tausche nur die Beschreibung, nicht dein Modell mit deinem Nachbargruppe. Versuche nun das Modell deines Nachbarn nur an Hand der Beschreibung zu falten.

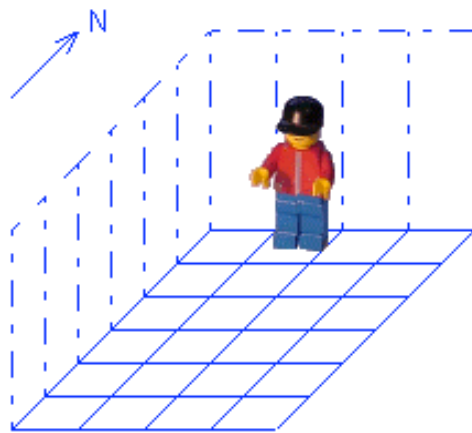
Die Übung 1.2 hat dir gezeigt, dass es gar nicht so leicht ist, den Faltvorgang so zu beschreiben, dass dein Nachbar beispielsweise das Flugzeug an Hand deiner Gebrauchsanweisung basteln kann. Eine gute Gebrauchsanweisung erkennst du daran, dass sie

1. in geeignete Einzelschritte gegliedert ist und
2. exakte und verständliche Formulierungen verwendet.

Auch die Entwickler von Software müssen beispielsweise den Robotern in einer geeigneten Sprache mitteilen, wie sie in einer Fabrik einen Autositz in eine Karosserie einbauen oder Teile zusammenschweißen sollen. Roboter und Automaten setzt man ein, wenn dieselben Anweisungen wiederholt und präzise nach einem vorgegebenen Programm ausgeführt werden sollen. Solche Folgen von Anweisungen nennt man Algorithmen:

Ein **Algorithmus** ist eine endliche Folge aus eindeutigen und ausführbaren Anweisungen zur Lösung eines allgemeinen Problems.

## 1.2 Der Roboter Karol



**Abbildung 1.1:** Der Roboter *Karol*

Nun möchte ich dir den kleinen Roboter *Karol* vorstellen. Karol trägt eine blaue Hose und eine rote Jacke. Da er mit Ziegelsteinen arbeitet, schützt er seinen Kopf mit einem schwarzen Helm. Karol lebt in einer kleinen Welt. In der Abbildung 1.1 ist diese Welt 4 Zellen breit und 6 Zellen lang. Wenn du dich mit Karol angefreundet hast, darfst du später seine Welt vergrößern.

### Übung 1.3;

Setze in Gedanken Karol auf eine beliebige Zelle in seiner Welt. Welche Angaben musst du deinem Nachbarn mitteilen, damit er diese Position erkennt?

Aus objektorientierter Sicht ist Karol ein Objekt der Klasse *Roboter* mit den Eigenschaften *PositionX*, *PositionY* und *Blickrichtung*:

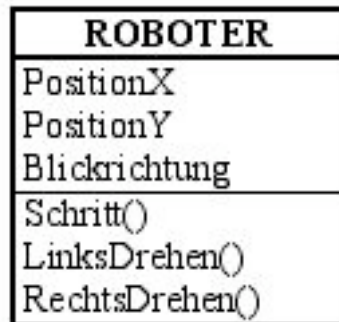


Abbildung 1.2: Die Klasse *ROBOTER*

Nun kann ein Objekt dieser Klasse erzeugt werden. Zuerst wird (automatisch) diesem Objekt der Namen Karol gegeben und in eine bestimmte Zelle gesetzt. Abbildung 1.3 zeigt, dass Karol in der Zelle mit den Koordinaten (2/1) mit Blickrichtung Süden steht.

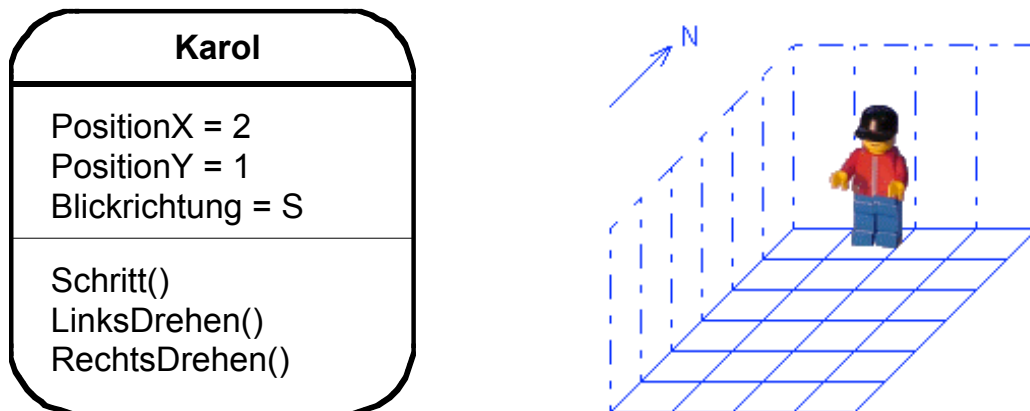
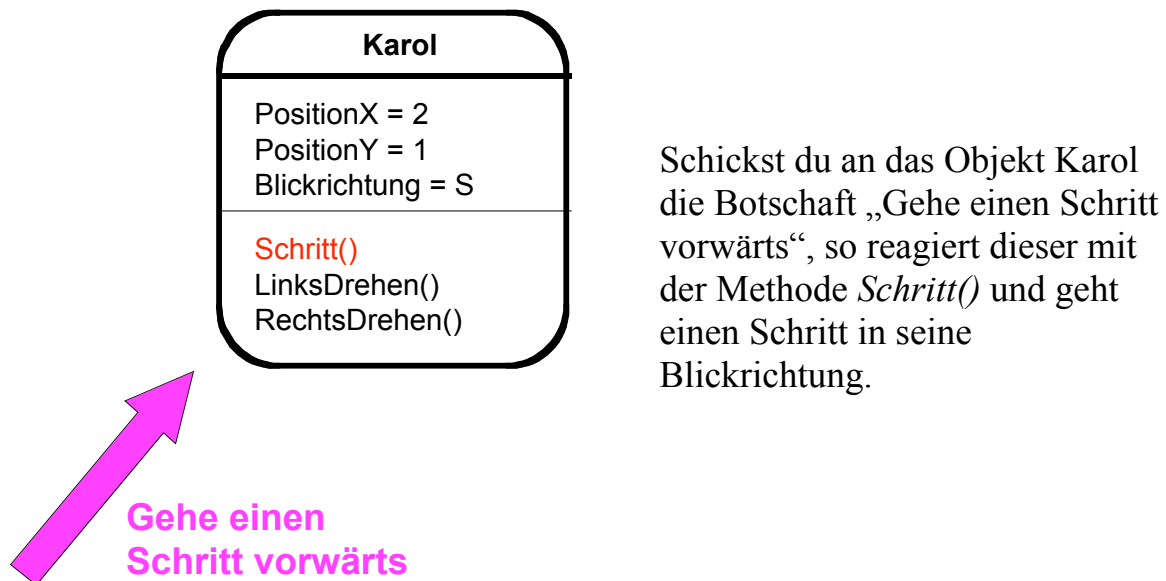


Abbildung 1.3: Karol steht in Zelle (2/1) mit Blickrichtung Süden

Neben diesen Eigenschaften hat Karol auch bestimmte Fähigkeiten. Er besitzt Methoden, um sich bewegen zu können. Beispielsweise kann er einen Schritt nach vorne machen oder sich nach links oder nach rechts drehen.

Wenn Karol nun einen Schritt nach vorne machen soll, rufst du ihm die Anweisung „Gehe eine Schritt vorwärts“ zu. Du sendest ihm also eine Botschaft und Karol reagiert darauf (hoffentlich, falls du keinen Fehler gemacht hast).

Karol führt die entsprechende Methode *Schritt()* aus, dabei ändern sich seine Attributwerte, in diesem Fall gilt anschließend  $PositionY = 2$ .



Schickst du an das Objekt Karol die Botschaft „Gehe einen Schritt vorwärts“, so reagiert dieser mit der Methode *Schritt()* und geht einen Schritt in seine Blickrichtung.

**Abbildung 1.4:** Botschaft “Gehe einen Schritt vorwärts“ wird gesendet

**Bemerkung:**

Die korrekte Anweisung lautet *Karol.Schritt()*.

1. Da jedoch nur ein einziges Objekt erzeugt werden kann, kannst du auf den Bezeichner *Karol* verzichten, obwohl Karol nichts dagegen hat, wenn du ihn mit dem korrekten Namen eine Botschaft sendest.
2. In Programmiersprachen wie Java erhalten die Methoden eine Klammer, in der alle Parameter angegeben werden können. Karol benötigt diese Klammern nicht, obwohl er auch diese akzeptiert.

Es reicht also aus, wenn sdu Karol befehlst *Schritt*, damit er einen Schritt vorwärts geht.

### 1.3 Die Programmierumgebung von Robot Karol

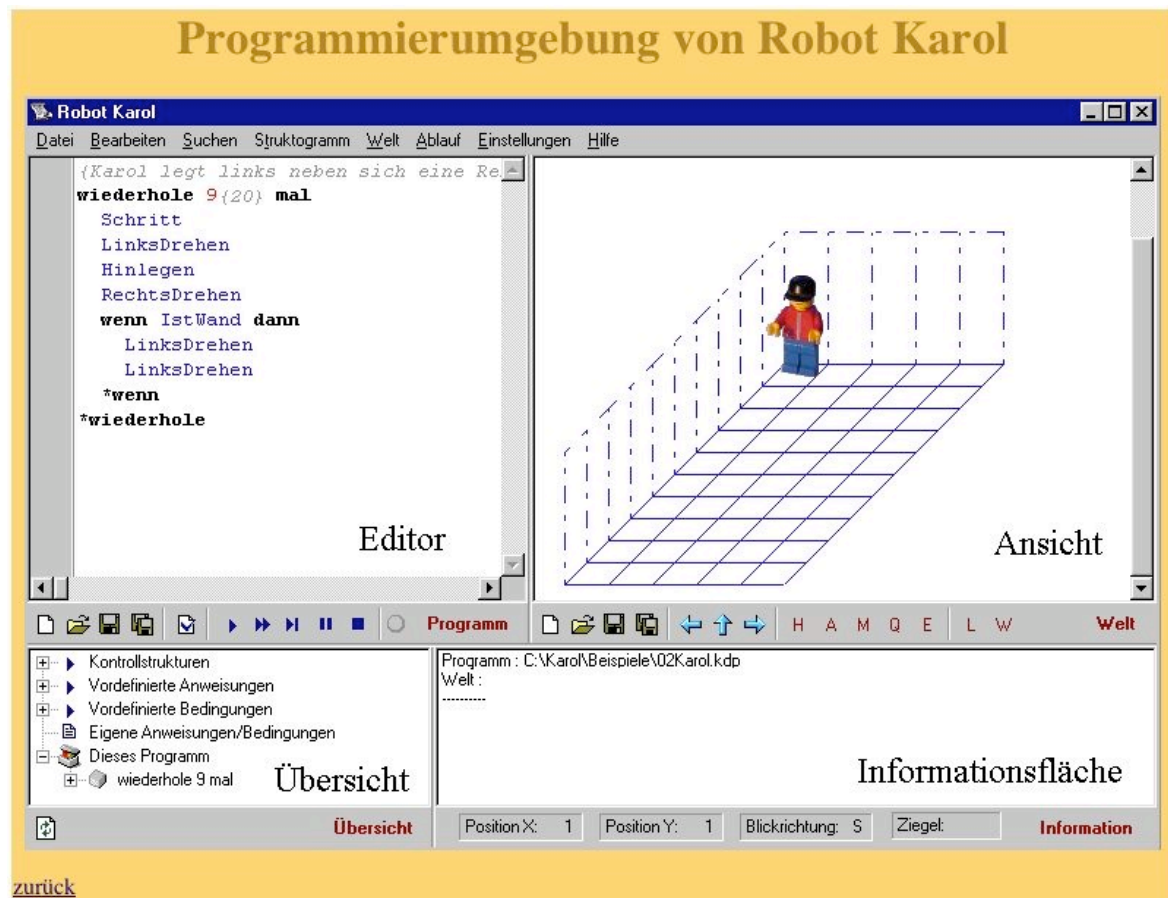



Abbildung 1.5: Programmierumgebung von Robot Karol

Die Programmierumgebung von Robot Karol gliedert sich in vier Bereiche:

1. Ansicht: Zeigt die Welt des Roboters Karol.
2. Informationsfläche: Liefert Informationen zum Programmablauf, beispielsweise Fehlermeldungen und Anzeige der aktuellen Attributwerte von Karol.
3. Übersicht: Zeigt Anweisungen, die Karol schon kennt, und die, die du ihm später beibringst, angezeigt.
4. Editor: Hier wirst du den Anweisungstext – das Programm – eingeben.




Übung 1.4:



Klicke auf die blaue Pfeiltaste . Was macht Karol. Probiere die anderen Tasten aus.



Wenn du auf die blaue Pfeiltaste  geklickt hast, schickst du an Karol die Botschaft „Gehe einen Schritt vor“, dieser reagiert mit seiner Methode *Schritt* und führt, soweit möglich, den Schritt aus.

### Übung 1.5:

Fülle die Tabelle aus.

Schaltfläche	Was macht Karol?
	
	
	
H	
A	
M	
Q	
E	

Mit der Schaltfläche  kann man die Welt löschen und mit  wieder herstellen.

Durch Klick auf  sieht man die Welt und Karol im Grundriss;  schaltet in die Schrägansicht zurück.

### Übung 1.6:

- Karol legt in jede Ecke zwei Ziegelsteine aufeinander.
- Karol legt mit Ziegelsteinen den Anfangsbuchstaben deines Vornamens.
- Karol legt ein Schachbrettmuster.
- Karol baut einen Würfel oder eine Pyramide.

## 1.4 Das Objekt Karol

### Übung 1.7:

Fülle die Leerzeilen aus:

Karol	Was macht Karol?
PositionX = 2 PositionY = 3 Blickrichtung = N	
Schritt()	_____
LinksDrehen()	_____
RechtsDrehen()	_____
Hinlegen()	_____
Aufheben()	_____
MarkeSetzen()	_____
MarkeLöschen()	_____
	_____
IstWand()	_____
NichtIstWand()	_____
IstZiegel()	_____
NichtIstZiegel()	_____
IstMarke()	_____
NichtIstMarke()	_____
IstNorden()	_____

Durch eine *Anweisung* sendet man eine \_\_\_\_\_ an Karol, der darauf mit der entsprechenden \_\_\_\_\_ reagiert.

### 1.5 Weitere Aufgaben für Karol im Direktmodus

Übung 1.8;

- Lege eine Reihe Ziegelsteine bis zur nächsten Wand.
- Lege den Rand eines Quadrates der Seitenlänge 4 mit Ziegeln aus.
- Lege ein Kreuz aus Ziegeln.
- Baue eine 3 Ziegel hohe und 5 Ziegel lange Mauer.
- Belege in einer quadratischen Welt beide Diagonalen mit Ziegeln.
- Baue eine 5 Stufen hohe Treppe.

- g) Baue die in Aufgabe a bis Aufgabe f errichteten Gebäude wieder ab.
- h) Denke dir selbst interessante Gebäude aus und baue sie. Teile deinem Nachbarn die benötigten Anweisungen mit.