

Die Sprache Karol

Die Sprache Karol umfasst:

- vordefinierte Anweisungen; eigene, selbstdefinierte Anweisungen;
- vordefinierte Bedingungen; eigene, selbstdefinierte Bedingungen;
- Kontrollstrukturen: bedingte Wiederholung; Wiederholung mit fester Anzahl; bedingte Anweisung;
- schnelle, langsame Ausführung; diverse Erweiterungen

Roboter Karol und seine Welt

Der Roboter Karol kann aus **objektorientierter Sicht** als ein **Objekt** der Klasse **ROBOTER** betrachtet werden, mit folgender Struktur:

Eigenschaften: Position (mit X und Y); Blickrichtung; Geschwindigkeit; maximale Sprunghöhe; ZiegelImRucksack; RucksackGröße;

Die **Methoden** (Fähigkeiten) des Roboters Karol werden, aufgrund ihrer Verwendung in zwei Gruppen unterteilt.

Karol hat vordefinierte Methoden mit denen er bestimmte Vorgänge ausführen kann.

Ohne äußere Aktionen geschieht noch gar nichts. Erst wenn man eine Botschaft an Karol schickt reagiert dieser mit der entsprechenden Methode. Man erteilt also Karol einen Befehl etwas zu tun, in dem man die jeweilige Methode aufruft (zum Beispiel „einen Schritt nach vorne gehen“). Bei dieser Sichtweise bedeutet eine **Anweisungen** im Programm das Senden einer Botschaft an das Objekt Karol, der mit der zugehörigen Methode darauf reagiert.

Eine neue Anweisung in der Sprache Karol festlegen (definieren) heißt, dem Roboter Karol eine neue Methode beibringen, mit der er auf diese Anweisung reagieren kann.

Karol hat aber auch Methoden, mit denen er auf eine Anfrage mit WAHR oder FALSCH antwortet (zum Beispiel „stehst du vor der Wand?“). Der Aufruf einer Methode dieser Art heißt **Bedingung**. Neue Bedingungen in der Sprache Karol festlegen heißt, dem Roboter Karol neue Methoden beibringen, mit denen er auf eine bestimmte Anfrage mit WAHR oder FALSCH antworten kann.

Um diese Vorstellung zu unterstützen können in Karol-Programmen alle Methoden (Anweisungen und Bedingungen) von Karol auch mit der Angabe des Objekts und Parameterklammern aufgerufen werden. Das heißt, es ist zum Beispiel neben der Schreibweise `schritt()` auch die Schreibweise `karol.schritt()` möglich.

Der Roboter Karol kann durch ein Programm in einer Welt mit Quadratmuster bewegt werden. Es gibt ein **Objekt der Klasse WELT** und dieses hat die Eigenschaften Breite, Länge und Höhe.

Eingerahmt ist die Welt an allen vier Seiten von Wänden in der entsprechenden Höhe.

Die Welt kann verschiedene Objekte aus den folgenden Klassen enthalten.

ZIEGEL: Diese Objekte kann Karol vor sich hinlegen und später wieder aufheben. An einer Stelle sind mehrere Ziegel aufeinander möglich, jedoch maximal bis zur Höhe der Welt. Karol ist ein kräftiger Bursche und kann beliebig viele (oder begrenzte Anzahl, je nach Einstellung der RucksackGröße) Ziegel mit sich herumschleppen. Er kann auch auf Ziegelstapel hinaufspringen und herabspringen, jedoch maximal so hoch wie seine Eigenschaft „maximaleSprunghöhe“ zulässt.

MARKE: Diese Objekte kann Karol an der Stelle anbringen, an der er sich gerade befindet. An jeder Stelle ist höchstens eine Marke möglich. Eine Stelle kann markiert sein oder nicht.

QUADER: Mit Quader kann die Welt zusätzlich gestaltet werden. An einer Stelle kann höchstens ein Quader stehen und Karol kann nicht auf einen Quader springen. Quader können nur im Direktmodus aufgestellt und entfernt werden, nicht im Programm. Im Programm verhalten sich Quader wie eine Wand.

Anweisungen

Die Ausführung einer Anweisung in einem Karol-Programm bedeutet eine Botschaft an Karol zu schicken, der mit der entsprechenden Methode reagiert und dabei eine gewisse Aktion ausführt.

Vordefinierte Anweisungen

Anweisung	Aktion die Karol ausführt	mögliche Ablauffehler
Schritt()	macht einen Schritt in die Blickrichtung	steht vor der Wand; steht vor einem Quader; kann nicht so hoch springen;
Schritt(Anzahl)	macht „Anzahl“-viele Schritte	wie bei Schritt
LinksDrehen()	dreht sich nach links (um 90°)	
RechtsDrehen()	dreht sich nach rechts (um 90°)	
Hinlegen()	legt vor sich einen Ziegel hin	steht vor der Wand; steht vor einem Quader; maximale Stapelhöhe erreicht; hat nichts zum Hinlegen ^(*) ;
Hinlegen(Anzahl)	legt „Anzahl“-viele Ziegel vor sich hin	wie bei Hinlegen
Aufheben()	hebt einen Ziegel auf, der vor ihm liegt	steht vor der Wand; steht vor einem Quader; kein Ziegel vor Karol; maximale Tragfähigkeit erreicht ^(*) ;
Aufheben(Anzahl)	hebt „Anzahl“-viele Ziegel auf, die vor ihm liegen	wie bei Aufheben
MarkeSetzen()	setzt an seiner Position eine Marke	
MarkeLöschen()	löscht an seiner Position eine Marke	
Warten()	wartet eine Sekunde	
Warten(Anzahl)	wartet „Anzahl“-viele Millisekunden	
Ton()	gibt einen Ton von sich	
Beenden()	stoppt den Programmablauf	

^(*) nur bei eingeschalteter Kontrolle der Tragfähigkeit

Bedingungen

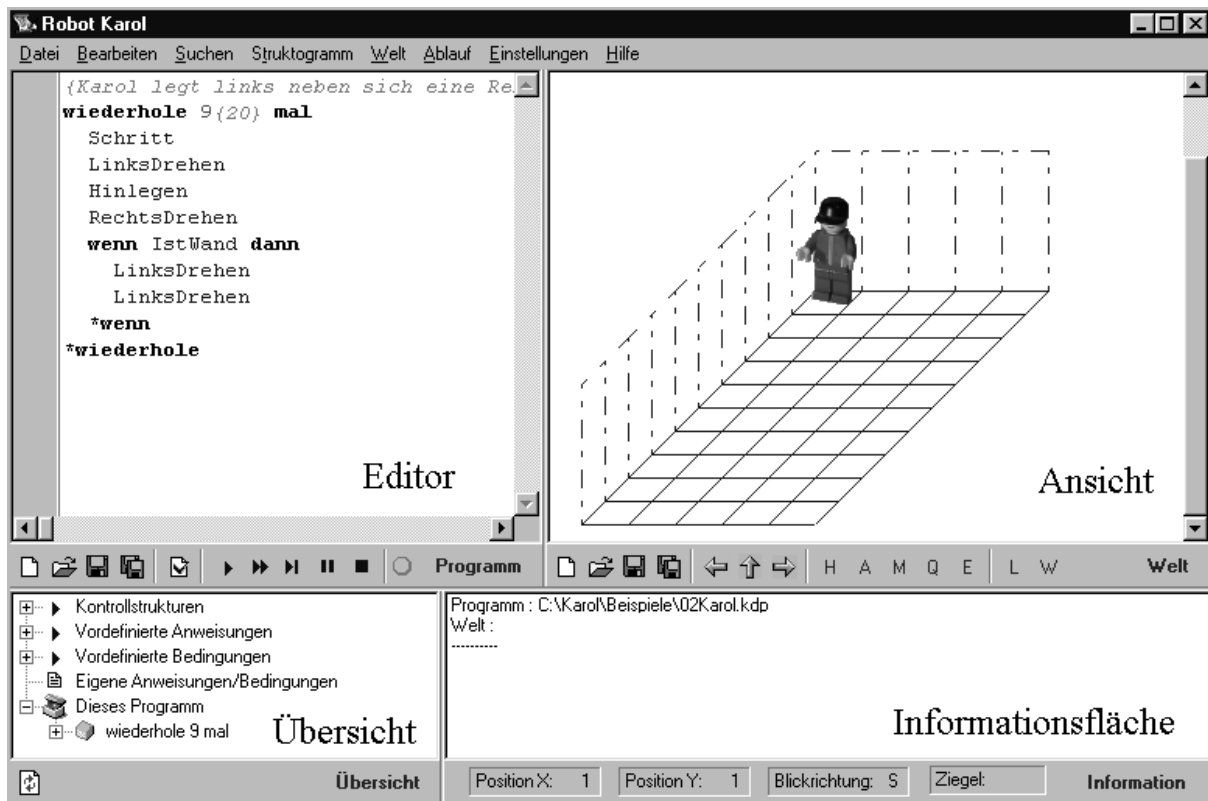
Die Ausführung einer Bedingung in einem Karol-Programm bedeutet eine Anfrage an Karol zu schicken, der mit der entsprechenden Methode reagiert, die Situation seiner Umgebung begutachtet und mit WAHR oder FALSCH antwortet.

Vordefinierte Bedingungen

Bedingung	Karol meldet WAHR,
IstWand	wenn er vor der Wand oder vor einem Quader steht und in diese Richtung schaut
NichtIstWand	wenn IstWand nicht zutrifft
IstZiegel	wenn er vor einem Ziegel oder Ziegelstapel steht und zu diesem schaut
IstZiegel(Anzahl)	wenn er vor einem Ziegelstapel mit „Anzahl“-vielen Ziegeln steht und zu diesem schaut
NichtIstZiegel	wenn IstZiegel nicht zutrifft
NichtIstZiegel(Anzahl)	wenn IstZiegel(Anzahl) nicht zutrifft
IstMarke	wenn er auf einer Marke steht
NichtIstMarke	wenn IstMarke nicht zutrifft
IstSüden, IstNorden, IstWesten, IstOsten	wenn Karol in diese Richtung schaut

Programmierungsumgebung

Die Programmieroberfläche „Robot Karol“ läuft unter dem Betriebssystem Windows (in der 32-Bit Version), d.h. Windows95, Windows98, WindowsME, WindowsNT, Windows2000 und WindowsXP. Sie wird in der üblichen Art und Weise wie andere Windows-Programme bedient.



Das Hauptfenster von Robot Karol umfasst die vier Teile Editor, Ansicht, Übersicht und Informationsfläche. Die Größe der Bereiche kann durch Ziehen mit der Maus an den Trennlinien eingestellt werden.

Editor

Texterfassung

Im Editor wird der Programmtext erfasst. Der Editor hebt den Programmtext, entsprechend der Syntax, farblich hervor. Durch Einrückungen mit <Tab> kann der Programmtext sinnvoll strukturiert werden. Die Einrückbreite von <Tab> wird über den Menüpunkt „Einstellungen - Editor“ festgelegt. Der Editor verfügt über die üblichen Fähigkeiten zum Kopieren, Ausschneiden, Einfügen, Suchen und Ersetzen von Text (siehe Menüpunkt „Bearbeiten“).

Für die vordefinierten Anweisungen und Bedingungen besteht, neben der Eingabe durch Eintippen, eine Eingabe durch Auswahl aus einer Aufklappliste, die man mit <Strg>+<Leer> öffnen kann. Zusätzlich kann man eingegebene Anfangsteile der reservierten Wörter durch <Shift>+<Leer> automatisch vervollständigen lassen (z.B. aus „wie“ wird „wiederhole mal“). Mit RechteMausKlick öffnet sich ein Aufklappenmenü aus dem man ein reserviertes Wort auswählen kann, das dann an der Klickstelle eingefügt wird.

Mit „Bearbeiten - Formatieren“ kann eine Formatierung des Textes vorgenommen werden (Der Umfang der Formatierung wird über „Einstellungen - Editor“ festgelegt).

Eine Einblendung von Zeilennummern am Rand hilft bei der Besprechung der Programme (ein- und ausschalten über den Menüpunkt „Einstellungen - Editor“)

Wichtige Steuertasten im Editor:

Tasten bzw. Tastenkombination	
<Pfeil rechts>	Cursor ein Zeichen rechts
<Strg>+<Pfeil rechts>	Cursor ein Wort rechts
<Pfeil links>	Cursor ein Zeichen links
<Strg>+<Pfeil links>	Cursor ein Wort links
<Pfeil oben>	Cursor eine Zeile nach oben
<Strg>+<Pfeil oben>	rollt Text um eine Zeile nach oben
<Pfeil unten>	Cursor eine Zeile nach unten
<Strg>+<Pfeil unten>	rollt Text um eine Zeile nach unten
<Bild auf>	rollt Text um eine Seite nach oben
<Bild ab>	rollt Text um eine Seite nach unten
<Pos1>	Cursor zum Zeilenanfang
<Strg>+<Pos1>	Cursor zum Textanfang
<Ende>	Cursor zum Zeilenende
<Strg>+<Ende>	Cursor zum Textende
<Umschalt>+<Steuertaste(n)>	markiert Text ab Cursorposition bis zur neuen Cursorposition; diese ergibt sich aus der obigen Übersicht
<Strg>+<a>	markiert gesamten Text
<Rück>	löscht Zeichen links vom Cursor
<Alt>+<Rück>	widerruft die letzte Aktion (Undo); mehrere Stufen möglich
<Umschalt>+<Rück>	führt die letzte widerrufenen Aktion wieder durch (Redo)
<Strg>+<y>	löscht die ganze Zeile in der sich der Cursor befindet
<Strg>+<Umschalt>+<ziffer>	setzt/löscht am Rand eine Markierung mit der Ziffer
<Strg>+<Ziffer>	setzt den Cursor an die markierte Position
<Umschalt>+<Leertaste>	ergänzt das Wort an der Cursorposition, sofern es sich um ein definiertes Schlüsselwort handelt
<Strg>+<Leertaste>	zeigt in einem Aufklappfeld alle zu dem Wortanfang passenden Schlüsselwörter
rechte Mausklick	öffnet AufklappMenü mit allen vordefinierten Kontrollstrukturen, Anweisungen und Bedingungen; fügt ausgewählte an der Cursorposition ein

Der Programmtext kann gespeichert, später wieder geöffnet und ausgedruckt werden (siehe Menü „Datei“). Die übliche Endung für Programmtexte ist *.kdp („Karol deutsch Programm“).

Programmablauf

Mit den Schaltflächen unterhalb des Editors oder über das Menü „Ablauf“ wird die Programmausführung gesteuert. Karol kann das Programm schrittweise (normaler „Programmstart“) und schnell ausführen („Schnelllauf“). Bei der schrittweisen Abarbeitung hält Karol nach jeder Anweisung eine bestimmte Zeit inne, deren Dauer über den Menüpunkt „Einstellungen - Karol“ festgelegt werden kann. Dies ermöglicht eine bessere Betrachtung des Programmablaufes. Zusätzlich ist auch „Einzelschritt“ möglich. Bei jedem Klick auf diese Schaltfläche führt Karol eine einzelne Anweisung aus und wartet bei der nächsten auf einen weiteren Klick. Durch „Pause“ kann das Programm angehalten und durch „Abbruch“ abgebrochen werden. Die „Lampe“ neben den Schaltflächen zeigt den Programmstatus (grün Programmablauf, gelb Einzelschritt).

Über den Menüpunkt „Ablauf - Stoppunkt“ bzw. <Strg>+ wird an der aktuellen Cursorstelle ein Stoppunkt gesetzt. Der Programmablauf stoppt bei Erreichen dieser Zeile. Über „Programmstart“, „Schnelllauf“ oder „Einzelschritt“ kann das Programm fortgesetzt werden.

Vor jedem Programmablauf wird der Programmtext einer Syntaxprüfung unterzogen. Liegt ein Fehler vor, so wird dieser in der Informationsfläche beschrieben und die betroffene Zeile durch ein Warnsymbol im Editorrand markiert. Ein Programm kann erst ablaufen wenn die Syntaxprüfung keine Fehler ergab.

Die Syntaxprüfung kann auch unabhängig von einem Programmstart über den Menüpunkt „Ablauf - Syntaxprüfung“ bzw. die Schaltfläche mit „Häckchen“ aufgerufen werden.

Ansicht / Karol-Welt

Karol-Welt

Im Bereich der Ansicht sieht man Roboter Karol in seiner Welt. Eine Welt umfasst neben ihrer Ausdehnung (Breite, Länge, Höhe) auch die Ziegel, Quader und Marken die in ihr liegen und die Startposition von Karol.

Über die Schaltflächen unterhalb der Ansicht bzw. dem Menü „Welt“ kann man eine neue Welt anlegen, eine Welt speichern und eine gespeicherte Welt öffnen. Die übliche Endung für Karol-Welten ist *.kdw (Karol deutsch Welt).

Die Welt kann auch als Grafik gespeichert werden und steht damit als Bild zur Verwendung in anderen Programmen zur Verfügung. Die 3D-Darstellung wird als Bitmap mit der Endung *.bmp (Windows Bitmap), die 2D-Darstellung als Grafik im Windows-Metafile-Format (*.emf) abgespeichert.

Für die Welt gibt es zwei Darstellungsmodi. Standard ist die 3D-Darstellung, in der Karol, die Ziegel und die Welt räumlich in einem Schrägbild gezeichnet werden. Zusätzlich gibt es eine 2D-Darstellung (Menü „Welt - 2D Darstellung“ oder Schaltfläche „2D“) in der die Welt als Grundriss dargestellt wird. Karol wird durch ein Dreieck repräsentiert und die Ziegel durch rote Quadrate. Die Anzahl der Ziegel in einem Stapel wird durch eine entsprechende Zahl wiedergegeben.

Festlegung der Karol-Welt

Die Karol-Welt kann auf zwei Arten festgelegt werden. Entweder durch Steuerung von Karol im Direktmodus und damit Anbringung von Ziegel, Quader oder Marken an bestimmten Stellen. Oder durch direktes Setzen/Löschen der Objekte mit Mausclick (nur in der 2D-Darstellung möglich).

Festlegung durch Karol im Direktmodus

Mit den Schaltflächen „Pfeile“ und „H“, „A“, „M“, „Q“, „E“ kann Roboter Karol direkt gesteuert und zur Arbeit (Ziegel hinlegen, aufheben; Marken setzen, löschen; Quader aufstellen, entfernen) aufgefordert werden.

„H“: Hinlegen - Karol legt auf das Feld vor sich einen Ziegel

„A“: Aufheben - Karol hebt einen Ziegel vom Feld vor sich auf

„M“: Marke - Karol setzt bzw. löscht eine Marke auf dem Feld auf dem er steht

„Q“: Quader - Karol stellt in dem Feld vor sich einen Quader auf

„E“: Entfernen - Karol entfernt einen Quader, der vor ihm steht

Damit kann eine Welt sowohl in der 3D-Darstellung als auch in der 2D-Darstellung aufgebaut werden.

Durch Tastendruck kann der Roboter Karol in seiner Welt ebenfalls direkt gesteuert werden. Die Tastensteuerung wird durch einen Klick auf die Welt aktiviert. Als Tasten sind möglich: <Pfeil links> für Linksdrehen, <Pfeil rechts> für Rechtsdrehen, <Pfeil oben> für Schritt vorwärts, <Pfeil unten> für Schritt rückwärts, <h> für Ziegel Hinlegen, <a> für Aufheben, <m> für Marke setzen/löschen, <q> für Quader aufstellen und <e> für Quader entfernen.



Festlegung durch direktes Setzen/Löschen

In der 2D-Darstellung können die Objekte der Welt durch Mausklick direkt gesetzt bzw. gelöscht werden. Hierzu ist das Werkzeugfenster über das Menü „Welt - Welt direkt festlegen“ aufzurufen.

Im Werkzeugfenster wird durch Klick der entsprechende Arbeitsmodus ausgewählt (von links nach rechts, von oben nach unten: Ziegel setzen, Ziegel entfernen, Marke setzen, Marke entfernen, Quader setzen, Quader entfernen, Karol setzen, Werkzeugfenster schließen). Ein Klick mit der Maus auf die entsprechende Stelle in der Karol-Welt führt die Aktion aus.

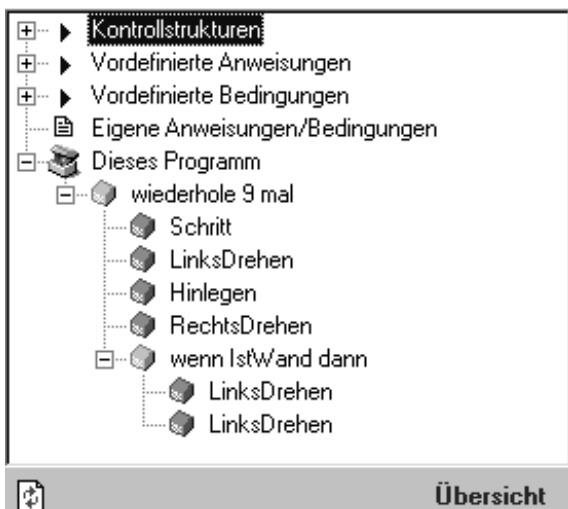
Wiederherstellen/Löschen

Während des Programmablaufs (siehe oben) kann man die Bewegungen und Arbeiten von Karol direkt in der Welt betrachten.

Durch die Schaltfläche „W“ kann die Welt wieder hergestellt werden. Wurde die Welt schon gespeichert, so wird sie entsprechend des gespeicherten Zustands wieder hergestellt, andernfalls nimmt die Welt den Zustand vor dem letzten Programmstart an.

„L“ löscht die ganze Welt und stellt Karol in die Ecke ganz hinten links (Ursprung).

Übersicht



Die Übersicht stellt in hierarchischen Listen sowohl vordefinierte als auch eigene Anweisungen und Bedingungen dar. Auf Wunsch kann das ganze Programm in Form einer Baumstruktur angezeigt werden (hierzu ist vorher ein Klick auf die Schaltfläche „Übersicht aktualisieren“ nötig). Dabei sind die einzelnen Kontrollstrukturen und die eingebundenen Blöcke hierarchisch dargestellt. Auch selbstdefinierte Anweisungen/Bedingungen aus Bibliotheken, die in das Programm eingefügt wurden werden dargestellt.

Informationsfläche

In der Informationsfläche wird ständig die aktuelle Position und Blickrichtung von Karol eingeblendet. Auf Wunsch kann die Anzahl Ziegel angezeigt werden, die Karol momentan trägt (siehe „Einstellungen - Karol“).

Im Textfeld werden die Namen und Pfade für die Dateien der aktuellen Welt und des Programms aufgeführt. Bei der Syntaxprüfung erfolgt die Ausgabe der Fehlermeldung in diesem Bereich.

Während des Programmlaufs werden hier Laufzeitfehler gemeldet. Diese treten auf wenn Karol einen Bewegungsfehler (z.B. läuft gegen die Wand) macht und im Dialog „Einstellungen - Karol“ die Anzeige dieser Fehler gewählt wurde.