

NIKI für Windows

von

Walter Hupfeld

Inhalt

- 1 Einführung
 - 1.1 Was ist NIKI für Windows?
 - 1.2 Informationen zum Handbuch
 - 1.3 Hinweise zum Kopierschutz
- 2 Start des Programms
 - 2.1 Aus der Windows-Programmgruppe
 - 2.2 Aus dem Dateimanager
- 3 Programmbedienung
 - 3.1 Dateimenü
 - 3.2 Programmenü
 - 3.3 Arbeitsfeld
 - 3.4 Teach-in-Menü
- 4 Die Programmiersprache
 - 4.1 Programmaufbau
 - 4.2 Kontrollstrukturen
 - 4.3 Prozeduren
 - 4.4 Sensoren
 - 4.5 Compileroptionen
 - 4.6 Include-Dateien
- 5 Anwendungsbeispiele
 - 5.1 Die Grundaufgabe
 - 5.2 Handsteuerung
 - 5.3 Teachin-Modus
 - 5.4 Programmsteuerung
 - 5.5 Prozeduren
 - 5.6 Entscheidungen
 - 5.7 Schleifen
 - 5.8 Rekursion
 - 5.9 Weitere Aufgaben
- 6 Installation
 - 6.1 Systemkonfiguration
 - 6.2 Installation
- 7 Anhang
 - 7.1 Literatur
 - 7.2 Syntaxdiagramme

Einführung

1.1 Was ist NIKI für Windows?

NIKI - der Roboter ist eine geschlossene Programmierumgebung mit eigenem Compiler und Editor für den Anfangsunterricht in *PASCAL*.

Die Idee zu diesem Programm entstammt Materialien zur Lehrerfortbildung in Nordrhein-Westfalen und geht ursprünglich auf das amerikanische Vorbild **Karel the Robot** von Richard E. Pattis zurück.

NIKI ist das Modell eines programmierbaren Roboters, der sich auf einem Arbeitsfeld der Größe 10 x 15 bewegen und dort Gegenstände aufnehmen und ablegen kann. Außerdem verfügt er über Sensoren, mit denen er den Zustand seiner Umgebung wahrnehmen kann. Roboterprogramme sind gewöhnliche Pascal-Programme mit allen Kontrollstrukturen (Verzweigungen und Schleifen), aber *ohne* Variablen.

Es gibt verschiedene Möglichkeiten den Roboter in Aktion zu setzen:

Handbetrieb

NIKI kann im Handbetrieb gesteuert werden. Er kann hierbei die folgenden vier Tätigkeiten ausführen.

- ? vor NIKI geht einen Schritt in Richtung des Pfeiles. Läuft er dabei gegen ein Hindernis, erscheint ein entsprechender Hinweis.
- ? drehe_links NIKI verändert die Blickrichtung um 90 Grad nach links.
- ? nimm_auf NIKI nimmt einen Gegenstand auf. Ist kein Gegenstand vorhanden, erscheint wieder ein entsprechender Hinweis.
- ? gib_ab NIKI legt einen Gegenstand auf dem Arbeitsfeld ab. Hat er keinen Gegenstand zum Ablegen oder belegt das Feld schon neun Gegenstände, erscheint eine Fehlermeldung.

Die vier NIKI-Befehle können entweder durch Auswahl der Menüpunkte im Menü *Teach-in* oder durch Drücken der entsprechenden Anfangsbuchstaben des Befehls auf der Tastatur eingegeben werden.

Teach-in-Modus

Im Teach-in-Modus erfolgt das 'Erlernen' von Arbeitsvorgängen, indem NIKI im Handbetrieb einmal exemplarisch gesteuert wird.

Dieser Vorgang läßt sich anschließend als Programm beliebig oft wiederholen. Tritt im Programmablauf allerdings ein Fehler auf, so bricht das Programm ab.

Teach-in-Programme können auf Diskette gespeichert und von dort geladen werden. Sie werden dort als lesbare Texte in Form von Prozeduren abgelegt und können daher nachträglich mit einem Editor verändert oder als Includedateien zu Programmen hinzugeladen werden.

Programmsteuerung

Erst mit Hilfe der Programmsteuerung kann NIKI komplexere Aufgaben bewältigen und flexibel auf seine Umgebung reagieren. Hierzu werden Programme in PASCAL-Syntax mit einem Texteditor geschrieben, anschließend mit dem Compiler übersetzt und sofort gestartet. Dabei ist NIKI in der Lage, flexibel auf seine Umgebung zu reagieren. Der Roboter verfügt hierzu über Sensoren, mit denen er seine Umgebung erfassen kann:

- ? **Platz_belegt** gibt an, ob auf einem Feld ein Gegenstand liegt;
- ? **vorne_frei** weist auf ein Hindernis in Blickrichtung hin;
- ? **rechts_frei** weist auf ein Hindernis rechts von NIKI hin;
- ? **links_frei** weist auf ein Hindernis links von NIKI hin;
- ? **hat_Vorrat** gibt an, ob NIKI einen Gegenstand bei sich trägt.

So kann man NIKI z.B. durch den Befehl "WHILE vorne_frei DO vor;" bis zu einem Hindernis bewegen.

Diese Programmanleitung will und kann kein Lehrbuch sein. Für den Unterricht sei hier auf die bereits erschienene Literatur (s. Anhang) verwiesen.

1.2 Informationen zum Handbuch

Vor dem ersten Einsatz von NIKI für Windows lesen Sie bitte den Abschnitt 6.2 des Handbuchs. Dort finden Sie Informationen zum Kopierschutz und zur Installation, denn NIKI für Windows ist von der gelieferten Diskette aus nicht lauffähig.

Im Handbuch erwarten Sie sechs Kapitel und ein Anhang.

Das 1. Kapitel gibt Ihnen einen kurzen Überblick über das Programm, den Aufbau des Handbuchs und den Kopierschutz.

Kapitel 2 zeigt Ihnen, wie Sie NIKI für Windows starten.

In Kapitel 3 wird der Aufbau des Bildschirms und die allgemeine Bedienung des Programms beschrieben.

Die Beschreibung sämtlicher Menüpunkte finden Sie in Kapitel 4.

Kapitel 5 stellt Ihnen Anwendungsbeispiele und Einsatzmöglichkeiten für den Unterricht vor.

Und Kapitel 6 gibt einige technische Hinweise zur Installation des Programms.

Start des Programms

Sie können NIKI für Windows auf zwei verschiedene Art und Weisen starten.

- Aus einer Windows-Programmgruppe,
- mit dem Dateimanager

2.1 Aus der Windows-Programmgruppe

Legen Sie mit Hilfe von Windows eine Programmgruppe an. Wie dies geht, finden Sie im Windows Handbuch beschrieben. Öffnen Sie die Windows Programmgruppe und klicken Sie bitte das NIKI-Symbol an. NIKI für Windows startet.

2.2 Mit dem Dateimanager

Öffnen Sie bitte die Hauptgruppe auf der Windows-Oberfläche. Dort finden Sie den Dateimanager. Klicken Sie den Dateimanager an. Er öffnet sich. Wählen Sie nun das Laufwerk und den Pfad, indem sich NIKI für Windows befindet. Klicken Sie nun "niki.exe" an und NIKI für Windows startet.

Programmbedienung und Benutzerführung

Im folgenden finden Sie die Beschreibung des NIKI-Menüs. Alle Menü-Punkte lassen sich auch mit der Tastatur bedienen, wenn man den unterstrichenen Buchstaben zusammen mit der ALT-Taste drückt.

3.1. Datei-Menü

Neu

Eine Programmdatei kann in einer neuen Programmdatei eingegeben werden.

Öffnen

Eine Programmdatei wird in den Speicher geladen. Der Text kann jetzt vom Compiler übersetzt werden oder mit dem Editor bearbeitet werden.

Speichern

Der im Speicher befindliche Text wird auf Diskette oder Festplatte geschrieben.

Speichern unter

Eine im Speicher befindlicher Programmtext kann unter einem anderen Namen gesichert werden. Dies ist sinnvoll, wenn Sie ein Programm für ein neues Arbeitsfeld weiterentwickeln wollen.

Beenden

Das Programm wird beendet. Eine Warnbox weist auf einen nicht gesicherten Programmtext oder ein verändertes Arbeitsfeld hin.

3.2 Programm-Menü

In diesem Menüpunkt findet der Programmablauf und die Programmentwicklung statt.

Programm starten

Der Programmtext wird zunächst übersetzt. Tritt während der Übersetzung ein Fehler auf, wird nach einer entsprechenden Warnung das Editorfenster geöffnet und der Text-Cursor hinter die Fehlerstelle gesetzt.

Bei einer erfolgreichen Übersetzung wird das NIKI-Programm sofort gestartet. Der Menüeintrag ändert sich während des Programmablaufs in **Anhalten**, durch erneute Anwahl kann jetzt NIKI gestoppt werden.

Tritt während des Programmablaufs ein Fehler in der Ausführung des Programms auf, wird das NIKI-Programm mit einem entsprechenden Hinweis beendet.

Fortfahren

Sie können ein angehaltenes NIKI-Programm wieder fortsetzen.

Einzelritt

Zur Fehlersuche ist es sehr hilfreich, wenn Sie gleichzeitig den Programmablauf und den gerade abgearbeiteten Quelltext beobachten können. Nach Anwahl des Menüpunktes *Einzelritt* öffnet sich das Editorfenster und zeigt durch einen Balken die Programmanweisung, die als nächste abgearbeitet wird. Durch Drücken einer beliebigen Taste wird das Programm Schritt für Schritt durchlaufen. Es ist hierbei sinnvoll, das Textfenster so zu verschieben, daß es die Sicht auf das Arbeitsfeld nicht stört.

Den Programmablauf im Einzelschrittmodus können Sie mit der Escape-Taste abbrechen.

Editor

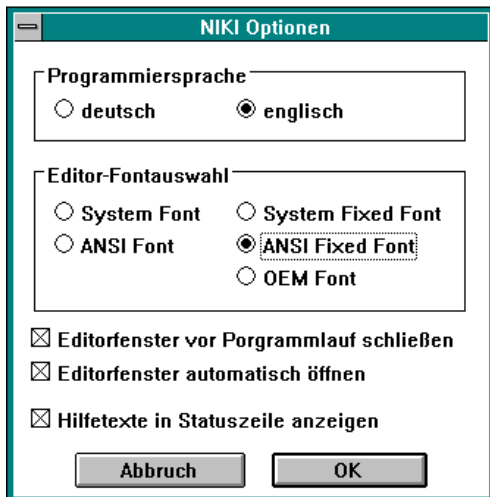
Das Editorfenster wird geöffnet bzw. geschlossen.

Der Editor unterstützt die Windows-üblichen Funktionen wie Kopieren, Ausschneiden und Einfügen. Durch Drücken der rechten Maustaste erhalten Sie ein Popup-Menü zur Textbearbeitung. Beim gleichzeitigen Drücken der Shift-Taste und der rechten Maustaste erscheint in Popup zur Fontauswahl.

Auch vom Editor aus können Sie NIKI-Programme durch Anwahl des entsprechenden lokalen Menüs starten bzw. anhalten.

Optionen

Nach Auswahl dieses Menüeintrags erscheint eine Dialogbox, mit der einige Einstellungen der NIKI-



Programmierumgebung vorgenommen werden können. Unter Programmiersprache können Sie zwischen der Verwendung englischer oder deutscher Schlüsselwörter wählen.

Mit der Editor-Fontauswahl haben sie die Wahl zwischen verschiedenen Systemfonts.

Beim Ankreuzen von *Editorfenster vor Programmlauf schließen* wird beim Starten eines NIKI-Programms das Editorfenster automatisch geschlossen.

Mit *Editorfenster automatisch öffnen* wird beim Laden einer neuen Textdatei das Editorfenster geöffnet. Beim ersten Öffnen einer Datei wird das Editorfenster immer geöffnet.

Mit *Hilfetexte in Statuszeile anzeigen* können Sie die Anzeige der Hilfetexte abschalten.

3.3 Arbeitsfeld

Dieser Menüpunkt dient zum Erstellen neuer Arbeitsfelder und zum Verändern bereits vorhandener Arbeitsfelder. Das Arbeitsfeld besteht aus 10 x 15 Quadraten, die an den Seiten durch Hindernisse begrenzt sein können. Auf jedem Quadrat haben maximal 9 Gegenstände Platz. Diese werden durch kleine Kreise dargestellt.

Verändern

Hinter dieser Funktion verbirgt sich ein komfortable Möglichkeit, das Arbeitsfeld zu verändern. Sobald der Menüpunkt angewählt wird, verändert der Mauszeiger seine Form in ein Rechteck. Fährt man nun mit dem Mauszeiger auf das Arbeitsfeld, verändert der Mauszeiger ständig seine Form, je nachdem in welchem Bereich eines Feldes er sich befindet.

Es gibt vier aktive Mauszeiger:

- Oval mit Pfeil

In dieser Stellung können Sie durch Drücken der linken Maustaste die Position von NIKI verändern. Mehrmaliges Drücken an der Stelle, an der sich der Roboter bereits befindet, ändert die Richtung von NIKI.

- Kleiner Kreis

Durch Drücken der linken Maustaste können Sie Gegenstände im entsprechenden Quadrat ablegen. Es ist Platz für maximal neun Gegenstände. Mit Hilfe der rechten Maustaste werden die Gegenstände wieder entfernt.

- Waagerechter und senkrechter Balken

Befindet sich der Mauszeiger auf den Grenzflächen zwischen den einzelnen Quadraten, verwandelt er sich in einen waagerechten bzw. senkrechten Balken. Jetzt können Sie Hindernisse durch Drücken der linken Maustaste hinlegen und mit der rechten Maustaste wieder entfernen.

Durch nochmaliges Anwählen des Menüpunktes wird das Verändern des Arbeitsfeldes beendet.

Neu

Es erscheint ein neues leeres Arbeitsfeld und NIKI wird in die Ausgangsposition auf das linke untere Quadrat gesetzt.

Zurücksetzen

Das Arbeitsfeld wird in den Zustand gebracht, den es beim Laden oder beim letzten Verändern hatte. Ist dieser Zustand noch vorhanden, findet keine Veränderung statt. Beim Starten eines NIKI -Programms wird das Arbeitsfeld automatisch zurückgesetzt.

Speichern und Laden

Sie können mit diesen Menüpunkten Arbeitsfelder auf Diskette oder Festplatte speichern, bzw. von dort laden.

3.4 Teach-in-Menü

In diesem Menü sind die Einträge zum Steuern von NIKI im Handbetrieb. Gleichzeitig sind die NIKI-Befehle auch über die Tastatur erreichbar.

Im Teach-in-Modus können Sie eine exemplarische Befehlssequenz ausführen und anschließend als Programm beliebig oft wiederholen.

Nach Aufrufen des Menüpunktes **Sequenz aufzeichnen** werden alle NIKI-Anweisungen, mit denen NIKI entweder durch Anwählen der entsprechende Menüpunkte oder durch die Tastatur gesteuert wird, aufgezeichnet

Der Roboter muß dabei einmal den vollständigen Arbeitsauftrag durchlaufen. Beendet wird diese Befehlssequenz im Teach-in-Modus durch wiederholte Anwahl des Menüpunktes.

Mit **Sequenz starten** wird ein solches Teach-in-Programm gestartet. Vorher wird das Feld zurückgesetzt. Tritt während des Programmablaufs ein Fehler auf, wird die Programmausführung abgebrochen.

Laden und Speichern

Sie können Teach-in-Programme mit **Speichern** auf Diskette oder Festplatte sichern oder ein gespeichertes Programm mit **Laden** von dort wieder in den Rechner laden.

Die Teach-in-Programme werden als Texte in Form von NIKI -Prozeduren abgespeichert. Der Prozedurname entspricht dabei dem Dateinamen (ohne Extension), der in der Dateiauswahlbox benannt wurde.

Man kann solche Teach-in-Programme als Includes zu Niki-Programmen hinzuladen und hat damit eine einfache, aber wirksame Methode, lange Befehlssequenzen ohne viel Tipparbeit zu erzeugen. Durch die Compileranweisung `{ $I Treppe.tip }` kann die Teach-in-Sequenz zu einer Programmdatei hinzugeladen werden. Die Sequenz wird dann durch den Befehl *Treppe* gestartet.

Die Programmiersprache

Die NIKI-Sprache entspricht einer Untermenge von PASCAL. Als Elementarstatements kommen nur die NIKI-Anweisungen in Betracht. Die Formulierung der Schlüsselwörter und der Programmanweisungen hängen von der gewählten Programmiersprache ab. Sie können je nach Einstellung im Optionen-Dialog englische oder deutsche Schlüsselwörter benutzen. Im folgenden werden die englischen Bezeichnungen verwendet.

4.1 Programmaufbau

Jedes NIKI-Programm besteht aus einem Programmkopf, der das Schlüsselwort PROGRAM und den Namen des Programms enthält, und Programmanweisungen, die mit BEGIN und END geklammert und durch ein Semikolon getrennt werden. Das Programmende wird mit einem Punkt markiert. Die entsprechenden Syntaxdiagramme befinden sich im Anhang.

Nach dem Programmkopf können Prozeduren folgen, die dem Aufbau von Programmen entsprechen. Der Prozedurkopf besteht aus dem Wort PROCEDURE und dem Prozedurnamen, gefolgt von einem Anweisungsblock. Prozeduren können sich dabei selbst aufrufen (**Rekursion**)

An jeder beliebigen Stelle können im Programm **Kommentare** eingefügt werden. Diese Kommentare werden beim Übersetzen eines Programmtextes ignoriert. Zwei Kommentarklammern sind möglich:

```
{ dies ist ein Kommentar mit geschweiften Klammern }  
(* dies ist eine Klammer mit einem Stern *)
```

Beispiele für ein NIKI-Programm in englisch:

```
Program Gehe_auf_das_Podest;
```

```
Procedure drehe_rechts;
```

```
Begin
```

```
drehe_links;
```

```
drehe_links;
```

```
drehe_links
```

```
End;
```

```
Begin
```

```
vor;
```

```
vor;
```

```
drehe_links;
```

```
vor;
```

```
drehe_rechts;
```

```
vor
```

```
End.
```

Beispiele für ein NIKI-Programm in deutsch:

```
Programm Gehe_auf_das_Podest;
```

```
Prozedur drehe_rechts;
```

```
Beginn
```

```
drehe_links;
```

```
drehe_links;
```

```
drehe_links;
```

```
Ende;
```

```
Beginn
```

```
vor;
```

```
vor;
```

```
drehe_links;
```

```
vor;
```

```
drehe_rechts;
```

```
vor
```

```
Ende.
```


4.2 Kontrollstrukturen

Die Formulierung der Kontrollstrukturen hängt von der gewählten Programmiersprache ab. Es können je nach Einstellung im Optionen-Dialog englische oder deutsche Schlüsselwörter verwendet werden. Die Syntax entspricht der von PASCAL.

Folgende Kontrollstrukturen sind in der NIKI-Programmiersprache enthalten:

- Einseitige und zweiseitige Entscheidungsanweisungen
- Wiederholungsanweisungen mit Eingangs- und Ausgangsbedingungen

Kontrollstrukturen in englisch:

Verbund-Anweisung:	BEGIN <i>Anweisungen</i> END;
Schleife mit Eingangsbedingung:	WHILE <i>Bedingung</i> DO <i>Anweisung</i> ;
Schleife mit Ausgangsbedingung:	REPEAT <i>Anweisungen</i> UNTIL <i>Bedingung</i> ;
einseitige Entscheidungsanweisung:	IF <i>Bedingung</i> THEN <i>Anweisung</i> ;
zweiseitige Entscheidungsanweisung:	IF <i>Bedingung</i> THEN <i>Anweisung</i> ELSE <i>Anweisung</i> ;

Kontrollstrukturen in deutsch:

Verbund-Anweisung:	BEGINN <i>Anweisungen</i> ENDE;
Schleife mit Eingangsbedingung:	SOLANGE <i>Bedingung</i> TUE <i>Anweisung</i> ;
Schleife mit Ausgangsbedingung:	WIEDERHOLE <i>Anweisungen</i> BIS <i>Bedingung</i> ;
einseitige Entscheidungsanweisung:	WENN <i>Bedingung</i> DANN <i>Anweisung</i> ;
zweiseitige Entscheidungsanweisung:	WENN <i>Bedingung</i> DANN <i>Anweisung</i> SONST <i>Anweisung</i> ;

4.3 NIKI-Prozeduren

Die folgenden Elementarstatements bilden den Grundwortschatz.

vor	NIKI geht eine Stelle in Blickrichtung weiter
drehe_links	NIKI dreht sich um 90° links
nimm_auf	NIKI nimmt einen Gegenstand von seiner Position auf
gib_ab	NIKI legt einen Gegenstand auf seiner Position ab
Arbeitsfeld(<i>Feld</i>)	<i>Feld</i> ist ein Arbeitsfeldname
abschalten	Das Programm wird beendet

Der Befehl *Arbeitsfeld* dient zum Laden eines Arbeitsfeldes von Diskette oder Festplatte. Er darf allerdings nur als erster Befehl in einem *Block* stehen, d.h. er muß entweder der erste Befehl im Hauptprogramm oder in einer Prozedur sein. Es wäre also denkbar, NIKI in einem Programm auf mehreren Arbeitsfeldern agieren zu lassen.

Die Anweisung *abschalten* kann an jeder beliebigen Stelle des Programms aufgerufen werden und schaltet den Roboter ab.

Sowohl *Arbeitsfeld* als auch *abschalten* sind optional und müssen nicht verwendet werden.

4.3 NIKI-Sensoren

Seine 'Intelligenz' erhält NIKI erst dadurch, daß er über *Sensoren* verfügt und mit diesen Sensoren flexibel auf seine Umgebung reagieren kann. Die Sensoren können als Bedingungen in die Entscheidungsanweisungen und Wiederholungsanweisungen eingesetzt werden.

Außerdem lassen sich die Sensoren mit NOT negieren, bzw. mit AND und OR verknüpfen. Dabei gelten die in PASCAL üblichen Vorrangregeln. Klammern binden stärker als NOT, NOT geht vor AND und dieses vor OR.

Es gelten die folgenden Verknüpfungstabellen:

Bedingung 1	Bedingung 2	AND	OR
falsch	falsch	falsch	falsch
falsch	wahr	falsch	wahr
wahr	falsch	falsch	wahr
wahr	wahr	wahr	wahr

(Beispiel: *WHILE vorne_frei AND NOT Platz_belegt DO vor;*)

	<i>Wahr, wenn</i>
vorne_frei	vor NIKI in Blickrichtung kein Hindernis steht
rechts_frei	rechts neben NIKI kein Hindernis ist
links_frei	links neben NIKI kein Hindernis ist
Platz_belegt	sich auf Nikis Position mindestens ein Gegenstand befindet
hat_Vorrat	NIKI mindestens einen Gegenstand als Vorrat besitzt
nordwaerts	NIKI in Richtung Nord blickt
suedwaerts	NIKI in Richtung Süd blickt
westwaerts	NIKI in Richtung West blickt
ostwaerts	NIKI in Richtung Ost blickt

4.4 Compileroptionen

In der vorliegenden Version ist lediglich eine Compileroption zum Umschalten der NIKI-Programmprache vorgesehen. Man kann so innerhalb eines Programms zwischen der Syntax für deutsche und englische Schlüsselwörtern wählen.

Compileroptionen stehen in Kommentarklammern, denen ein \$-Zeichen folgt. Die folgenden Optionen sind möglich:

D+	schaltet die deutsche Version ein
D-	schaltet die englische Version ein
D=	schaltet zu der vor der letzten Compiler-Option gültigen Version

Gültige Ausdrücke sind z.B. { \$ D+ } oder (*\$ D= *).

4.5 Include-Dateien

Es ist möglich, Include-Dateien in NIKI-Programme einzufügen.

Der Compileroption **I** folgt sofort der Dateiname der Include-Datei. Die Include-Datei wird dann auf dem aktuellem Pfad gesucht. Dadurch ist es möglich, den Quelltext auf mehrere Dateien zu verteilen, was die Projektarbeit erheblich vereinfacht.

Beispiel:

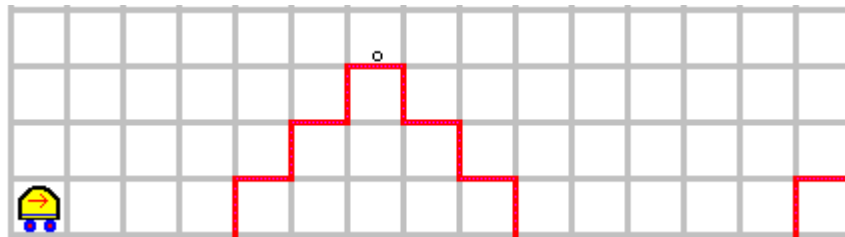
Durch { \$ IBASIS.PAS } wird die Datei BASIS.PAS zum Quelltext hinzugeladen. Auf diese Weise lassen sich auch Teachin-Sequenzen zu einem Programmtext laden. Die Sequenz wird durch Aufruf des entsprechenden Dateinamens gestartet.

Anwendungsbeispiele

Mit Hilfe einer kleinen Einführungssequenz lernen Sie die Möglichkeiten der NIKI-Programmierungsumgebung kennen. Im folgenden werden bei allen Beispielen die englischen Schlüsselwörter verwendet.

5.1 Die Grundaufgabe

NIKI befindet sich auf einem Arbeitsfeld mit einer Treppe und einem Podest. NIKI soll den Gegenstand von der obersten Stufe der Treppe auflesen und zum Podest transportieren und anschließend vor dem Podest stehenbleiben.



Nikis Möglichkeiten sind sehr eingeschränkt. Der Roboter kann Gegenstände aufnehmen und abgeben, er kann sich in Pfeilrichtung nach vorne bewegen und er hat die Möglichkeit, sich nach links zu drehen.

5.2 Handsteuerung

Zunächst soll NIKI mit der Hand gesteuert werden. Hierzu können Sie den Roboter mit Hilfe der Tastatur bewegen. Folgende Tasten können benutzt werden:

- v - gehe einen Schritt nach vorne
- d - Drehung um 90° nach links
- n - nimm einen Gegenstand auf
- g - gib einen Gegenstand ab

Eine entsprechende Eingabesequenz könnte dann folgendermaßen aussehen:

v v v v d v d d d v d v d d d v ...

☞ Führen Sie jetzt die oben gestellte Aufgabe mit Hilfe der Handsteuerung durch.

5.3 Teachin-Modus

Die Handsteuerung benötigt immer noch einen Menschen, der den Roboter steuert. Sinnvoller wäre es, NIKI könnte die Aufgabe selbständig wiederholen und einen einmal durchgeführten Arbeitsauftrag erlernen. Dies ist mit Hilfe des Teachin-Modus möglich.

Wenn man im Menü **Teach-in** den Eintrag **Sequenz aufzeichnen** auswählt, werden alle nachfolgenden Arbeitsanweisungen aufgezeichnet. Die Aufzeichnung wird durch Anwahl des gleichen Menüeintrags beendet. Mit **Sequenz starten** kann NIKI jeweils den gleichen Arbeitsauftrag beliebig oft wiederholen. Mit **Speichern** kann man Sequenzen auf Festplatte speichern.

☞ Erstellen Sie eine Teachin-Sequenz für die Anfangsaufgabe und speichern Sie diese unter dem Namen *Treppe0.tip* auf Diskette oder Festplatte.

5.4 Programmsteuerung

Das Aufzeichnen von Arbeitssequenzen ist schon ein bedeutender Schritt zur Automatisierung. Allerdings kann NIKI hierbei noch nicht auf Veränderungen seiner Umgebung reagieren. Ein verändertes Arbeitsfeld, z.B. ein fehlender Gegenstand würde einen Fehler auslösen.

Erst mit Hilfe der Programmsteuerung kann NIKI flexibel reagieren.

NIKI-Programme werden mit Hilfe eines Texteditors geschrieben, von einem speziellen Programm, dem Compiler, in NIKI-Anweisungen übersetzt.

Die Programmentwicklung findet im Programm-Menü statt. Mit **Editor** wird der Programmeditor geöffnet, mit **Starten** wird das Übersetzungsprogramm gestartet und, falls kein Syntaxfehler aufgetreten ist, das NIKI-Programm gestartet. Beide Aktionen können Sie mit der Tastenkombination **Alt+E** und **Alt+S** auch über die Tastatur auslösen.

NIKI-Programme haben einen festgelegten Aufbau. So muß jedes Programm mit dem Schlüsselwort **Program** und dem Namen des Programms beginnen. Danach folgt ein Semikolon. Die Niki-Anweisungen werden mit **Begin** und **End** geklammert, einzelne Anweisungen durch ein Semikolon getrennt. Das Programmende markiert ein Punkt.

Wir wollen jetzt für die obige Aufgabe ein Programm entwerfen:

```
Program Treppe0;  
Begin  
  vor;  
  vor;  
  vor;  
  drehe_links;  
  vor;  
  drehe_links;  
  drehe_links;  
  drehe_links;  
  vor;  
  ....  
End.
```

☞ Vervollständigen Sie das Programm mit den fehlenden Anweisungen.

5.5 Prozeduren

Wenn man die Aufgabe als Programmsequenz eingibt, hat man relativ viel Schreibarbeit. Betrachtet man den Programmtext, fällt auf, daß häufig gleiche Programmsequenzen auftauchen. So muß sich NIKI, um sich einmal nach rechts zu drehen, dreimal nach links drehen.

Man kann solche wiederkehrenden Befehlssequenzen zu neuen NIKI-Befehlen zusammenfassen - NIKI kann also neue Befehle erlernen.

Das Formulieren neuer Befehle geschieht mit Hilfe von Prozeduren. In einer Prozedur werden NIKI-Befehle zu einer neuen Anweisung zusammengefaßt.

Beispiel für eine Prozedur:

```
Procedure drehe_rechts;  
Begin  
  drehe_links;  
  drehe_links;  
  drehe_links  
End;
```

Beliebig viele Prozeduren können direkt hinter dem Programmkopf eingefügt werden:

Program Treppe1;

Procedure drehe_rechts;

Begin

drehe_links;

drehe_links;

drehe_links

End;

Begin

vor;vor;vor;

drehe_links;

vor;

drehe_rechts;

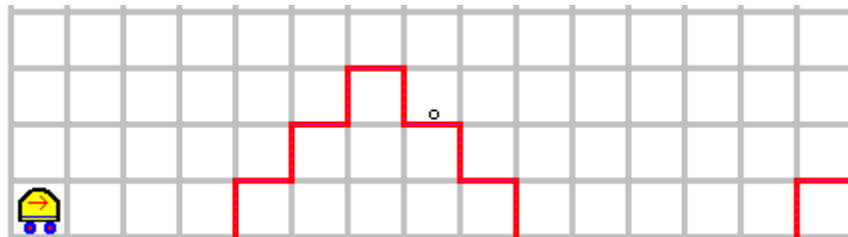
....

End.

✎ Ändern Sie das Programm aus der letzten Aufgabe entsprechend.

5.6 Entscheidungen

NIKI bekommt eine neue Aufgabe. Er soll von der Treppe einen Gegenstand auflesen. Es ist aber nicht bekannt, an welcher Stelle auf der Treppe dieser Gegenstand liegt. Ein Arbeitsfeld könnte folgendermaßen aussehen:



Um entscheiden zu können, wo sich ein Gegenstand befindet, benötigt NIKI Sensoren, mit denen er seine Umgebung abtasten kann. NIKI verfügt über mehrere Sensoren. Der Sensor für Gegenstände heißt **Platz_belegt**. Niki kann diesen Sensor abfragen und dann entscheiden, ob er einen Gegenstand aufnehmen muß. Dies geschieht mit der Anweisung

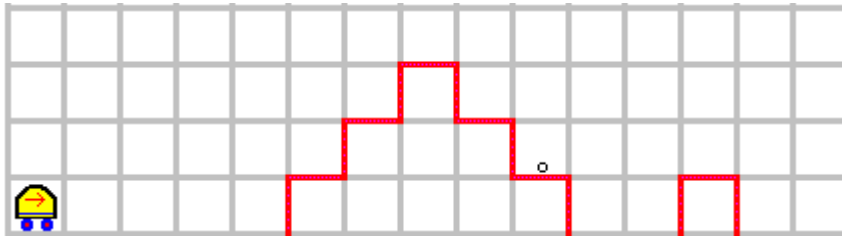
If Platz_belegt **then** nimm_auf;

Eine solche Anweisung wird *Entscheidungsanweisung* genannt.

✎ Verändern Sie das Programm für die oben gestellte Aufgabe so, daß sich der Gegenstand an einer beliebigen Stelle auf der Treppe befinden kann.

5.7 Schleifen

Das Arbeitsfeld wird jetzt so verändert, daß NIKI die Lage der Treppe und des Podestes nicht kennt. Treppe und Podest stehen hintereinander, aber weder der Abstand zur Treppe noch der Abstand zwischen Treppe und Podest sind bekannt.



NIKI verfügt allerdings über einen Sensor, mit dem er Hindernisse wahrnehmen kann. Dieser Sensor heißt **vorne_frei**.

Mit der folgenden Anweisung kann NIKI bis zu einem Hindernis laufen;

```
While vorne_frei do vor;
```

Eine solche Anweisung heißt *Wiederholungsanweisung* oder *Schleife*.

✍️ Verändern Sie das Programm für die oben gestellte Aufgabe so, daß sich Treppe und Podest an einer beliebigen Stelle befinden können.

5.8 Rekursion

Da NIKI keine Variablen kennt, läßt sich die folgende Aufgabe nur schwer realisieren:



NIKI soll bis zum Hindernis gehen, sich umdrehen und dann exakt die gleiche Anzahl von Schritten wieder zurückgehen.

Da in der NIKI-Programmiersprache die Möglichkeit besteht, daß sich Prozeduren selbst aufrufen, ist eine Lösung mit Hilfe der **Rekursion** möglich. Eine Prozedur heißt rekursiv, wenn sie sich innerhalb eines Anweisungsblockes selbst aufruft. Der Aufruf muß ein bedingter sein und folgende Form haben:

```
Procedure rekursiv_arbeiten;
```

```
Begin
```

```
If <Bedingung> Then  
  <Anweisung1>
```

```
Else
```

```
  Begin
```

```
    <Anweisung2>;  
    rekursiv_arbeiten;  
    <Anweisung3>
```

```
  End;
```

```
End;
```

Für das oben aufgeführte Beispiel ergibt sich die folgende Lösung:

Program Gehe_zum_Hindernis_und_zurueck;

Procedure drehe_um;

Begin

drehe_links;

drehe_links

End;

Procedure Zum_Hindernis_und_zurueck;

Begin

IF not vorne_frei **then**

drehe_um

Else

Begin

vor;

Zum_Hindernis_und_zurueck; {Prozedur ruft sich selbst auf}

vor;

End;

End;

Begin

Zum_Hindernis_und_zurueck

End.

Wir wollen nun die Grundaufgabe dahingehend erweitern, daß NIKI über eine beliebig hohe Treppe geht. Man kann den Lösungsweg für eine rekursive Prozedur **gehe_über_die_Treppe** so beschreiben:

Falls noch eine Stufe kommt, dann

gehe eine Stufe hoch

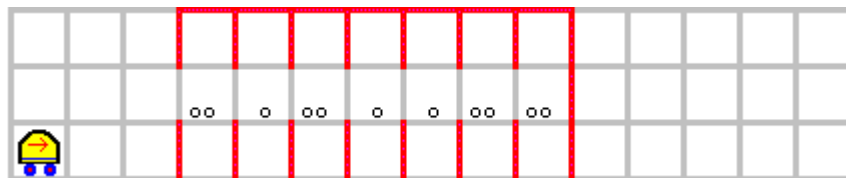
gehe über die Treppe

gehe eine Stufe runter

☞ Erweitern Sie die Grundaufgabe entsprechend, so daß NIKI über eine beliebig hohe Treppe gehen kann.

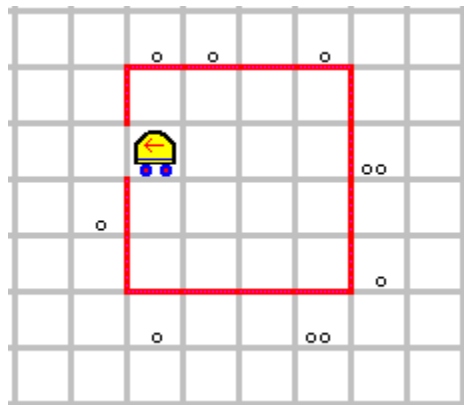
5.9 Weitere Aufgaben

Aufgabe Lagerhalle



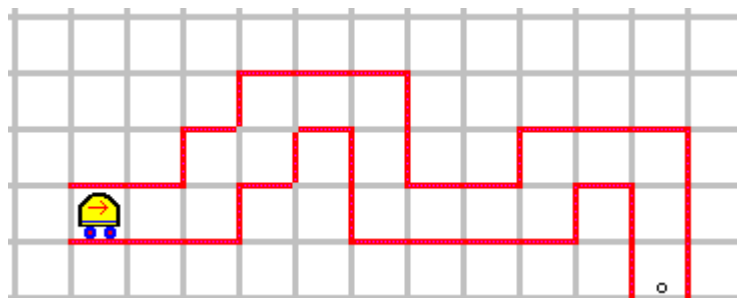
Im Gang einer Lagerhalle sind Paletten abgestellt. NIKI soll so programmiert werden, daß die hohen Paletten (zwei Gegenstände) in die rechte Regalreihe, die kleinen Paletten in die linke Regalreihe einsortiert werden.

Aufgabe Müll



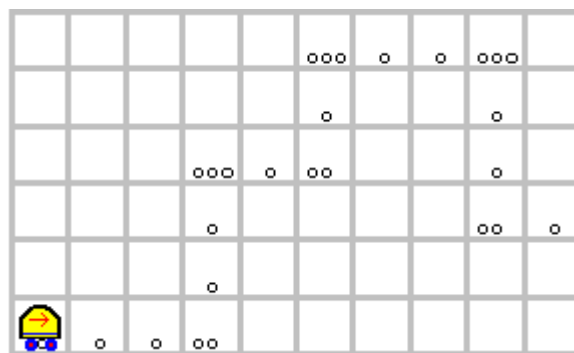
Um die Fabrikhalle herum hat sich Müll angesammelt. NIKI erhält den Auftrag, den Müll einzusammeln und in die Halle auf seine Ausgangsposition zurückzukehren. Er kennt allerdings die Ausmaße der Halle nicht und weiß auch nicht, wieviel Müll sich auf den einzelnen Feldern befindet. Programmieren Sie ein entsprechendes NIKI-Programm.

Aufgabe Tunnel



NIKI soll ein Arbeitswerkzeug am Ende des Tunnels holen. Der Tunnel ist jeweils ein Feld breit, die genaue Form des Tunnels ist allerdings nicht bekannt. Programmieren Sie ein entsprechendes Programm.

Aufgabe Schnitzeljagd



NIKI soll einer Spur aus Scheiben folgen. Die Spur besteht aus eins, zwei oder drei Scheiben. Dabei gibt die Anzahl der Scheiben den Weg vor. Sind keine Scheiben mehr vorhanden, ist die Spur zu Ende und NIKI bleibt stehen. Es bedeutet:

- 1 Scheibe : NIKI geht einen Schritt vor
- 2 Scheiben: NIKI biegt nach links ab und geht einen Schritt vor
- 3 Scheiben: NIKI biegt nach rechts ab und geht einen Schritt vor

Installation

6.1 Systemkonfiguration

Anhang

7.1. Literaturverzeichnis

Lehrerfortbildung in NRW,
Neue Technologien, Informatik in de gymnasialen Oberstufe, Heft 1.2., Landesinstitut für Schule und
Weiterbildung, Soest

Am Hermes, D. Stobbe: *Informatik eins*, Ernst Klett Verlag, Stuttgart

Wolfgang Ambros: *Der Hamster*, Programmieren lernen in einer Modellwelt, J.B. Metzler, Stuttgart

D. Stobbe: *Niki der Roboter*, J.B. Metzler, Stuttgart (noch nicht erschienen)

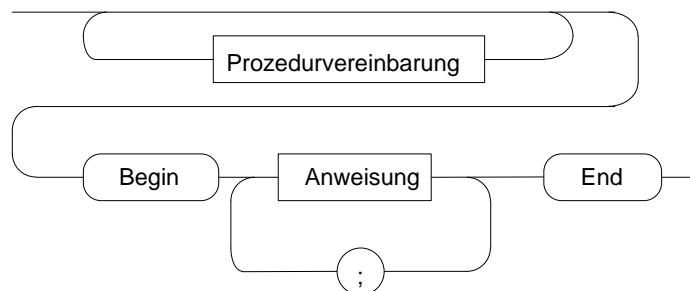
A. Knülle-Wenzel, H.-J. Himmeröder: *ROBI der Roboter*, Westermann Verlag, Braunschweig

7.2. Niki-Syntaxdiagramme

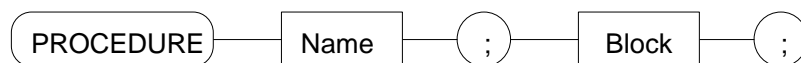
Program



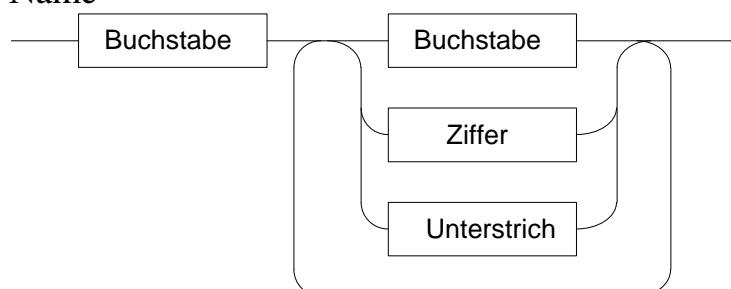
Block



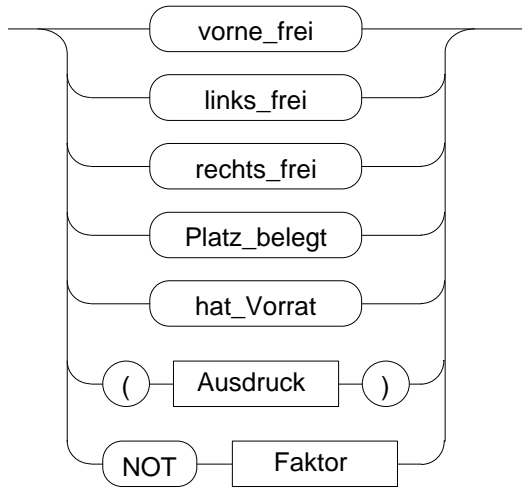
Prozedurvereinbarung



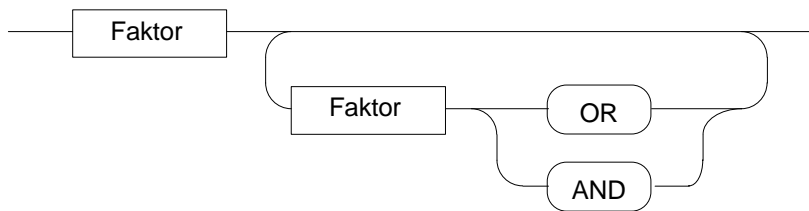
Name



Faktor



Ausdruck



Anweisung

