

Robot Karol - Bau eines Schwimmbads

Stand: 22.03.2017

Jahrgangsstufen	Lernbereich 2: Modul 2.6.1
Fach/Fächer	Informationstechnologie
Übergreifende Bildungs- und Erziehungsziele	Technische Bildung
Zeitraumen	2 Unterrichtsstunden
Benötigtes Material	Programmierungsumgebung Robot Karol

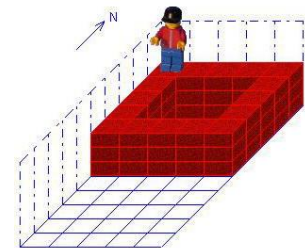
Kompetenzerwartungen

Die Schülerinnen und Schüler

- analysieren Abläufe (z. B. bedingte Bewegung eines Roboters) und gliedern diese in sinnvolle Teilschritte, um dazu eindeutige Handlungsvorschriften zu formulieren.
- verwenden algorithmische Grundstrukturen, um Abläufe zu modellieren und stellen sie mit geeigneten Notationsformen dar.
- testen und optimieren ihre einfachen Programme.

Aufgabe

Roboter Karol will sich ein quadratisches Schwimmbad bauen. Die Seiten sollen fünf Ziegel breit und das Schwimmbad drei Reihen hoch sein. Karol steht am Anfang und am Ende des Programms in seiner Ausgangsposition in der Nord-West-Ecke seiner Welt mit Blickrichtung nach Süden.



Anforderungen an deine Lösung:

1. Dein Programmcode soll möglichst kurz sein.
2. Nutze die Programmstruktur „Wiederholung“ für Arbeitsschritte bzw. Abläufe, die Karol beim Bau des Schwimmbads mehrmals ausführen muss.

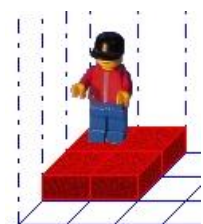
Hinführende Aufgaben als Hilfestellung (zu bearbeiten in Partnerarbeit)

Die Bearbeitung der folgenden zwei Teilaufgaben führt euch zur schrittweisen Entwicklung der Lösung für den Schwimmbadbau. Auch hier gilt, dass der Programmcode durch die Verwendung von Wiederholungsstrukturen möglichst kurz sein soll.

Teilaufgabe 1 „Quadrat legen“

Karol soll mit vier Ziegelsteinen ein Quadrat am Boden legen. Am Anfang und am Ende des Programms steht er in seiner Ausgangsposition in der Nord-West-Ecke seiner Welt mit Blickrichtung nach Süden.

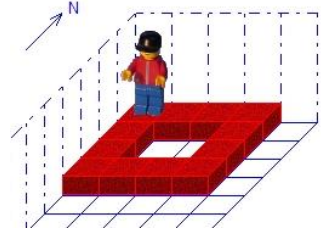
Tipp: Spiele mit einem Partner die nötigen Aktionen des Roboters Karol nach: Einer von euch spielt Karol und verwendet z. B. als Ziegel DIN-A4-Blätter, die er auf An-



weisung am Boden ablegt. Der andere steuert Karol mit den bekannten Befehlen. Spielt die Aktionen auf verschiedene Arten solange durch, bis ihr eine Folge von Befehlen verwendet, die sich in der gleichen Reihenfolge mehrfach wiederholen!

Teilaufgabe 2 „Sandkasten bauen“

Karol baut einen quadratischen Sandkasten, der eine Reihe hoch ist. Die Seiten sind vier Ziegel lang. Am Anfang und am Ende des Programms steht Karol in seiner Ausgangsposition in der Nord-West-Ecke seiner Welt mit Blickrichtung nach Süden.



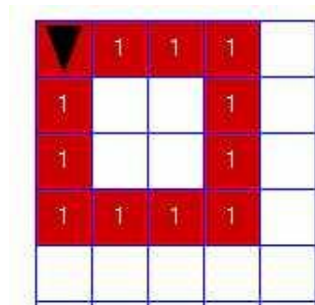
Tipp: Spiele mit einem Partner die nötigen Aktionen des Roboters Karol nach: Spielt solange, bis ihr entdeckt habt, welche Aktionen sich in zwei ineinander geschachtelte Wiederholungen packen lassen.

Hilfestellung zur Teilaufgabe 2

Diese Sicht von oben auf den fertigen Sandkasten, hilft dir, um zu erkennen, wie viele Ziegel Karol pro Seite legen muss und wie oft sich dies wiederholt.

Tipp:

Nummeriere die Ziegel durch, die pro Seite gelegt werden müssen und markiere die Wiederholungen!



Quellen- und Literaturangaben

- Download der Programmierumgebung Robot Karol unter <https://www.mebis.bayern.de/infportal/faecher/mint/inf/robot-karol/> (Stand: 12/2016)
- Handreichung des Arbeitskreises am ISB „Der Schwerpunkt Informatik im Fach Natur und Technik“ (6. und 7. Jgst.), Kapitel 3.3: Beschreibung von Abläufen durch Algorithmen, München, 2005, Seite 118, Download unter: <https://www.isb.bayern.de/gymnasium/materialien/d/der-schwerpunkt-informatik-im-fach-natur-und-techn/> (Stand: 12/2016)
- Illustrierende Aufgabe zum LehrplanPLUS des Gymnasiums für Natur und Technik, 7. Jgst.: 7.2.4 Beschreibung von Abläufen durch Algorithmen – Gartenmauer, Download der PDF-Datei unter: https://www.lehrplanplus.bayern.de/zusatzinformationen/aufgabe/lernbereich/56589/fachlehrplaene/gymnasium/7/nt_gym (Stand: 12/2016)
- Bildquellen: Screenshots (Autorin C. Hirtl-Baur)
- Elektronisches Schulbuch inf-schule (Pädagogisches Landesinstitut Rheinland-Pfalz): Algorithmusbegriff, <http://www.inf-schule.de/algorithmen/algorithmen/algorithmusbegriff> (Stand: 12/2016)

Hinweise zum Unterricht

Folgende Inhalte aus dem Modul 2.6.1 sollten bereits bekannt sein:

- Algorithmus: Begriff und Beispiele
- Algorithmische Grundstrukturen: Anweisung, Sequenz, Wiederholung mit Zähler

Im Elektronischen Schulbuch inf-schule (vgl. Literaturhinweise) wird z. B. der Algorithmusbegriff anhand der Steuerung von Robot Karol eingeführt. Zudem werden grundlegende Eigenschaften von Algorithmen (Ausführbarkeit, Eindeutigkeit, Endlichkeit und Allgemeinheit) mit Beispielen erklärt.

Die algorithmische Struktur der „Wiederholung mit Zähler“ sollte ebenfalls vor dieser Aufgabe eingeführt werden, z. B. anhand der Bauanleitung für einen Turm: Karol geht zwei Schritte nach Osten, dann fünf Schritte nach Süden, dreht sich nach Osten und baut einen Turm aus vier Ziegelsteinen.

Hauptprogramm
LinksDrehen
Schritt
Schritt
RechtsDrehen
Schritt
Schritt
Schritt
Schritt
Schritt
LinksDrehen
Hinlegen
Hinlegen
Hinlegen
Hinlegen

Diese Aufgabe wird zuerst als Sequenz (vgl. Struktogramm links) gelöst. Anschließend wird die Struktur der „Wiederholung mit Zähler“ eingeführt und zur Strukturierung im Programmcode werden so oft wie möglich Wiederholungen mit fester Anzahl (vgl. Struktogramm rechts) eingebaut.

Bei der Arbeit mit der Programmierumgebung Robot Karol ist das spielerische Erfassen der Aufgabe und eine gemeinsame, interaktive Suche nach der Lösung stets hilfreich: Ein Schüler spielt Robot Karol und verwendet z. B. als Ziegel DIN-A4-Blätter, die er in der Hand hält und auf Anweisung am Boden ablegt bzw. aufhebt. Ein anderer Schüler steuert ihn mit den bekannten Befehlen. Verschiedene Versuche werden gestartet bis eine optimale Lösung entsteht. Erst danach programmieren die Schüler an ihren Rechnern.

Hauptprogramm
LinksDrehen
wiederhole 2 mal
Schritt
RechtsDrehen
wiederhole 5 mal
Schritt
LinksDrehen
wiederhole 4 mal
Hinlegen

Die Aufgabe „Schwimmbadbau“ kann je nach Vorwissen der Schüler direkt gelöst oder durch die schrittweise Programmierung der beiden Teilaufgaben erarbeitet werden. Erfahrungsgemäß benötigen die Schüler mehrere Entwürfe, um für den Bau eines Sandkastens (Teilaufgabe 2) den kürzesten Programmcode mit zwei verschachtelten Wiederholungen zu erkennen und umzusetzen (vgl. Beispiele für mögliche Lösungen). Bei Bedarf sollte den Schülern die Hilfestellung (Sicht des Sandkastens von oben) zur Verfügung gestellt werden. Die Übertragung des Ergebnisses der Teilaufgabe 2 auf den Bau eines Schwimmbads fällt den Schülern in der Regel relativ leicht.

Beispiele für mögliche Lösungen der Schülerinnen und Schüler

Lösung zur Teilaufgabe 1 „Quadrat legen“

Struktogramm	Programmcode				
<p>Hauptprogramm</p> <table border="1"> <tr><td>wiederhole 4 mal</td></tr> <tr><td> Hinlegen</td></tr> <tr><td> Schritt</td></tr> <tr><td> LinksDrehen</td></tr> </table>	wiederhole 4 mal	Hinlegen	Schritt	LinksDrehen	<pre> wiederhole 4 mal Hinlegen Schritt LinksDrehen *wiederhole </pre>
wiederhole 4 mal					
Hinlegen					
Schritt					
LinksDrehen					

Hinweis: Mithilfe des Menüpunkts „Struktogramm“ wird grafisch die Struktur des Programmcodes angezeigt.

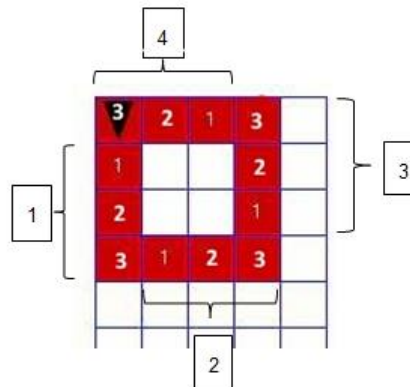
Lösung zur Teilaufgabe 2 „Sandkasten bauen“ mit zwei ineinander geschachtelten Wiederholungen

Struktogramm	Programmcode
<p>Hauptprogramm</p> <pre> wiederhole 4 mal wiederhole 3 mal Hinlegen Schritt Linksdrehen </pre>	<pre> wiederhole 4 mal {4 Seiten des Quadrats} wiederhole 3 mal {je 4 Ziegel pro Seite} Hinlegen Schritt *wiederhole Linksdrehen *wiederhole </pre>

Hinweis: Häufig entwickeln Schüler zuerst Lösungen mit einer bzw. vier Wiederholungen hintereinander.

<pre> wiederhole 4 mal Hinlegen Schritt Hinlegen Schritt Hinlegen Schritt Linksdrehen *wiederhole </pre>	<pre> wiederhole 4 mal hinlegen Schritt *wiederhole linksdrehen wiederhole 3 mal hinlegen Schritt *wiederhole linksdrehen wiederhole 3 mal hinlegen Schritt *wiederhole linksdrehen wiederhole 2 mal hinlegen Schritt *wiederhole </pre>
--	--

Die **Hilfestellung** (Sicht von oben) mit Beschriftung bietet einen Impuls, um die Verwendung zweier ineinander geschachtelter Wiederholungen zu erkennen:



Lösung zum „Bau eines Schwimmbads“

Struktogramm	Programmcode
<p>Hauptprogramm</p> <pre> wiederhole 3 mal wiederhole 4 mal wiederhole 4 mal Hinlegen Schritt Linksdrehen </pre>	<pre> wiederhole 3 mal {3 Reihen hoch} wiederhole 4 mal {4 Seiten des Quadrats} wiederhole 4 mal {5 Ziegel pro Seite} Hinlegen Schritt *wiederhole Linksdrehen *wiederhole *wiederhole </pre>

Einige Schüler fassen in ihrer Lösung die beiden äußeren Wiederholungen mit Zähler zu einer Wiederholung zusammen:

Struktogramm	Programmcode
<p>Hauptprogramm</p> <pre> graph TD A[wiederhole 12 mal] --> B[wiederhole 4 mal] A --> C[LinksDrehen] B --> D[Hinlegen] B --> E[Schritt] </pre>	<pre> wiederhole 12 mal {3 Reihen + 4 Seiten} wiederhole 4 mal {5 Ziegel pro Seite} Hinlegen Schritt *wiederhole LinksDrehen *wiederhole </pre>

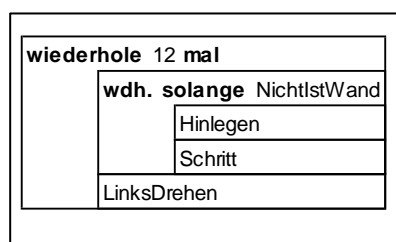
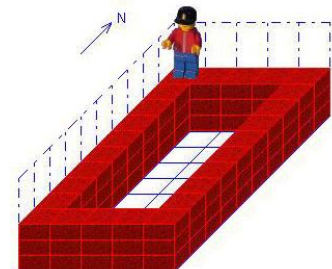
Anregung zum weiteren Lernen

Eine weitere Bearbeitung der Lösung zum Schwimmbadbau mit ineinander geschachtelten Wiederholungen mit Zähler bietet die illustrierende Aufgabe zum LehrplanPLUS des Gymnasiums für das Fach Natur und Technik: NT 7.2.4 Beschreibung von Abläufen durch Algorithmen – Gartenmauer (vgl. Literaturhinweise).

Einführung weiterer algorithmischer Grundelemente

Wiederholung mit Anfangsprüfung

Karol soll in einer **beliebig großen Welt** ein Schwimmbad bauen. Er läuft dabei an den vier Wänden entlang und legt Ziegel ab. Die Seiten des Schwimmbads bestehen aus so vielen Ziegeln wie die Welt lang bzw. breit ist. Das Schwimmbad soll drei Reihen hoch werden.



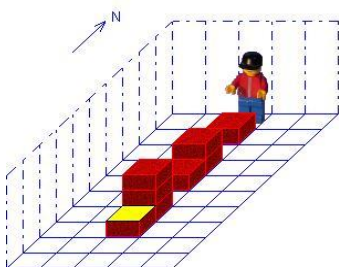
Für das erfolgreiche Legen einer Seite ohne dass Karol an der Wand anstößt, ist bei dieser Aufgabenstellung eine Wiederholung mit Anfangsbedingung sinnvoll:

Die Anweisungen im Rumpf der Schleife werden so oft wiederholt, solange die Bedingung den booleschen Wert „wahr“ zurückgibt. Die Bedingung wird stets am Anfang jeder Wiederholung überprüft.

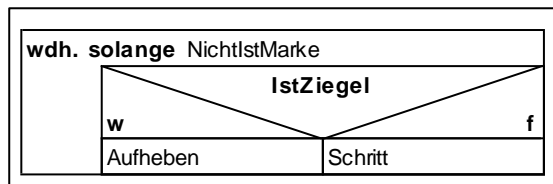
Bevor Karol vorwärts geht und einen Ziegel ablegt, muss er prüfen, ob keine Wand vor ihm ist. Die Bedingung „NichtIstWand“ entspricht der Frage an Karol: „Ist vor dir keine Wand?“ Solange Karol mit „Ja“ antwortet, wird die Wiederholung ausgeführt.

Zweiseitige Auswahl

Für die Einführung der zweiseitigen Auswahl bieten sich Aufgaben an, bei denen Karol in einer bereits vorgefertigten Welt reagieren muss. Selbst gestaltete Welten können als Datei im Format *.kdw abgespeichert werden.



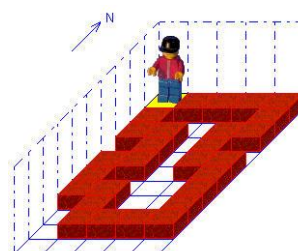
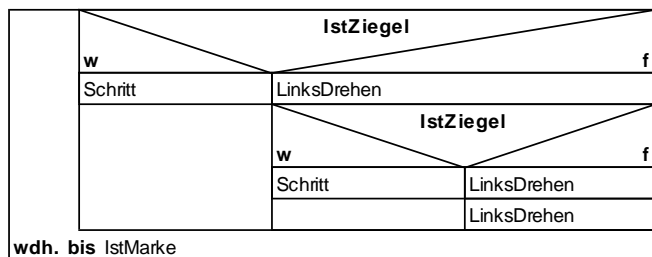
Karol soll **einzelne oder mehrere Ziegel aufheben**, die in einer Reihe vor ihm liegen. Dabei bewegt er sich vorwärts bis er eine Marke erreicht.



Komplexere Programmieraufgaben

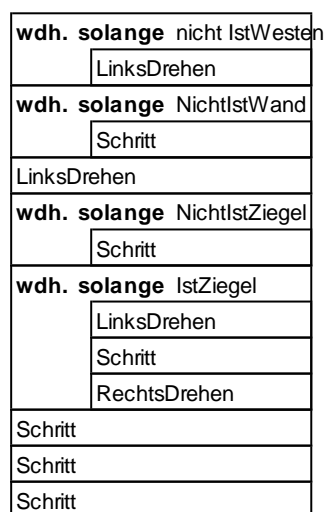
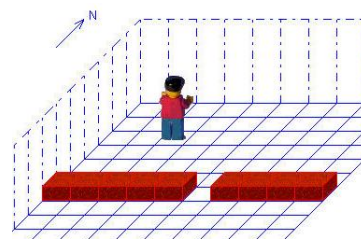
Zur Vertiefung und Verknüpfung der erlernten algorithmischen Grundelemente werden komplexere Programmieraufgaben in einer vorgegebenen Welt bearbeitet:

Beispiel 1: „Rundweg“: Karol läuft einen mit Ziegeln gelegten Weg entlang, auf dem eine Marke Start und Ziel kennzeichnet.



In einer Wiederholung mit Endeprüfung ist eine mehrstufige Auswahl eingebettet, mit deren Hilfe Karol prüft, ob vor ihm, rechts von ihm oder links von ihm ein Ziegel liegt, auf dem er seinen Weg fortsetzen kann.

Beispiel 2: „Türe finden“: In Karols Welt befindet sich in West-Ost-Ausrichtung eine Wand aus Ziegeln mit einer Lücke an einer beliebigen Position innerhalb der Mauer. Karol steht mit beliebiger Blickrichtung nördlich der Mauer. Er soll zu dieser Mauer gehen, die Türe (Lücke) suchen und hindurchgehen.



Da Karol irgendwo rechts bzw. links von der Türe nördlich der Mauer mit beliebiger Blickrichtung steht, muss er sich zunächst an einen fest definierten Ausgangspunkt begeben. Deshalb dreht er sich zuerst nach Westen, läuft bis zur westlichen Wand und dreht sich dann nach Süden. Von dort aus läuft er bis zur Mauer und dann schrittweise an ihr entlang bis er die Lücke gefunden hat und hindurch gehen kann.

Denkbar wäre auch ein kreatives Arbeiten, wobei sich Schüler selbst Aufgaben und Lösungen für Karol ausdenken und diese vor der Klasse vorstellen.

Hinweis: Der Inhalt „Variablenkonzept: Bezeichner, Datentyp, Wertzuweisung“ des Moduls 2.6.1 kann mit „Robot Karol“ nicht umgesetzt werden. Dazu muss im Anschluss auf ein anderes Programmierwerkzeug umgestiegen werden, z. B. eine Programmiersprache, die im Modul 1.2 oder 2.6.2 verwendet wird.