

Niveaubestimmende Aufgabe zum Fachlehrplan

Informationstechnik

Fachgymnasium

„Füllalgorithmus“

Schuljahrgang 11

Arbeitsstand: 28.04.2017

Niveaubestimmende Aufgaben sind Bestandteil des Lehrplankonzeptes für das Gymnasium und das Fachgymnasium. Die nachfolgende Aufgabe soll Grundlage unterrichtlicher Erprobung sein. Rückmeldungen, Hinweise, Anregungen und Vorschläge zur Weiterentwicklung der Aufgabe senden Sie bitte über die Eingabemaske (Bildungsserver) oder direkt an andrea.neubauer@lisa.mb.sachsen-anhalt.de

An der Erarbeitung der niveaubestimmenden Aufgabe haben mitgewirkt:

Dr. Preuschoff, Frank

Halle

Schneider, Jörg

Magdeburg

Schreiber, Sikor

Halle

Schulze, Holger

Halle (Leitung der Fachgruppe)

Herausgeber im Auftrag des Ministeriums für Bildung des Landes Sachsen-Anhalt:

Landesinstitut für Schulqualität und Lehrerbildung Sachsen-Anhalt

Riebeckplatz 09

06110 Halle



Die vorliegende Publikation, mit Ausnahme der Quellen Dritter, ist unter der „Creative Commons“-Lizenz veröffentlicht.

 CC BY-SA 3.0 DE

<http://creativecommons.org/licenses/by-sa/3.0/de/>

Sie dürfen das Material weiterverbreiten, bearbeiten, verändern und erweitern. Wenn Sie das Material oder Teile davon veröffentlichen, müssen Sie den Urheber nennen und kennzeichnen, welche Veränderungen Sie vorgenommen haben. Sie müssen das Material und Veränderungen unter den gleichen Lizenzbedingungen weitergeben.

Die Rechte für Fotos, Abbildungen und Zitate für Quellen Dritter bleiben bei den jeweiligen Rechteinhabern, diese Angaben können Sie den Quellen entnehmen. Der Herausgeber hat sich intensiv bemüht, alle Inhaber von Rechten zu benennen. Falls Sie uns weitere Urheber und Rechteinhaber benennen können, würden wir uns über Ihren Hinweis freuen.

Füllalgorithmus

Einführungsphase

Entwickeln Sie einen Programmalgorithmus zur Lösung der folgenden Problemstellung und stellen Sie diesen visuell dar.

Gegeben sei eine zweidimensionale Anordnung von rechteckigen Zellen (vgl. Abbildung). Einige der Zellen sind besetzt, d. h. mit einem Element besetzt, andere hingegen sind leer. Die Verteilung sei rein zufällig. Die beliebig große zweidimensionale Anordnung von Zellen soll mindestens zwei Spalten und eine Zeile besitzen.

Der Automat hat die Aufgabe, alle leeren Zellen der zweidimensionalen Anordnung mit je einem Element aufzufüllen, damit sich am Ende auf jedem Feld genau 1 Element befindet.

x				
	x			
x				
			x	
x	x		x	x
	x			x
		x		

Abb.1: mögliche Anordnung der Felder

Ein Automat kann sich schrittweise von Zelle zu Zelle bewegen (Kommando **vor**) und hat die Fähigkeit bei Bedarf eine Linksdrehung um 90° (Kommando **linksUm**) zu vollführen. Als weitere Kommandos stehen dem Automaten das Kommando **vornFrei** zum Testen der Wegfreiheit in Laufrichtung und die Kommandos **ElementDa** sowie **VorratLeer** zur Verfügung. Das Kommando **ElementDa** testet, ob auf der aktuellen Zellposition des Automaten bereits ein Element liegt. Mit dem Kommando **ElementGib** kann der Automat ein Element in das aktuelle Feld setzen. Das Kommando **VorratLeer** testet, ob der Vorratsspeicher an Elementen beim Automaten leer ist.

1. Analysieren Sie die möglichen Fallkonstellationen und zerlegen Sie die Gesamtaufgabe systematisch in überschaubare Teilaufgaben. Stellen Sie diese als Programmgrobstruktur in Form einer hierarchischen Abhängigkeitsübersicht dar.
2. Realisieren Sie die Feinentwürfe für die Lösung der Teilaufgaben in Form von Struktogrammen.
3. Testen Sie Ihren Entwurf mithilfe einer Modellumgebung am Computer.

Einordnung in den Fachlehrplan

Kompetenzschwerpunkt:

- Daten und Informationen analysieren und zur Verarbeitung aufbereiten

zu entwickelnde Schlüsselkompetenzen:

- Strukturen und Funktionen technischer Systeme und Prozesse erkennen
- selbstständige Handlungsweise weiterentwickeln
- mathematischen Strukturen in Problemen erkennen
- logisches und räumliches Denken verbessern
- digitale Werkzeuge nutzen

zu entwickelnde fachspezifische Kompetenzen:

- Informationen aus verschiedenen Quellen ermitteln und validieren, mittels informationstechnischer Systeme aufarbeiten und interpretieren
- Algorithmen mittels Tabellenkalkulation entwerfen

Bezug zu grundlegenden Wissensbeständen:

- Grundstrukturen von Programmabläufen (z. B. Iteration, Selektion, Sequenz, Funktionsblock)
- Darstellungsformen von Algorithmen (z. B. Struktogramm, Pseudotext)
- logische Verknüpfungen (z. B. AND, NOT, OR)

Anregungen und Hinweise zum unterrichtlichen Einsatz

Im Vordergrund steht das Erkennen der Zweckmäßigkeit einer funktionalen Verfeinerung/Strukturierung bei der effektiven algorithmischen Problembewältigung. Lösungsansätze, die ohne granulare Zerlegung auskommen, sollten daher möglichst frühzeitig verhindert werden. Die Bearbeitung sollte unter Nutzung einer Mikroprogrammierwelt erfolgen, welche eine visuelle Programmierung unterstützt, da eine formale, nicht jedoch textuelle Programmbeschreibung erfolgen soll. Hierfür eignen sich insbesondere Systeme wie der Hamstersimulator, Scratch, Byob oder Blopp.

Die Schülerinnen und Schüler haben bereits eine ähnliche Aufgabenstellung bearbeitet. Die Aufgabenbewältigung kann sowohl in Einzelarbeit als auch im Team erfolgen. Auf die fachgerechte und prozessbegleitende elektronische Dokumentation sollte besonderes Augenmerk gelegt werden. Die Arbeitszeit für die Lösung der niveaubestimmenden Aufgabe beträgt ca. zwei Unterrichtsstunden.

Variationsmöglichkeiten

Anstelle einer Visualisierung mit Struktogrammen kann auch eine andere Form vorgegeben oder diese den Schülerinnen und Schülern als freie Wahl überlassen werden. Eine Aufgabenvereinfachung ist erreichbar, wenn die Anzahl der Zeilen als geradzahlig vorgegeben wird. Weiterhin könnte ein Lösungsansatz in einer visuellen Darstellungsform mit der Zielstellung der Überführung in eine andere visuelle Form vorgegeben werden, z. B. Scratchdarstellung nach Struktogramm.

ERPROBUNG

Erwarteter Stand der Kompetenzentwicklung

Aufgabe	erwartete Schülerleistung	AFB prozent. Anteil
1	<p>Die Schülerinnen und Schüler erstellen eine Übersicht folgender Fälle:</p> <ul style="list-style-type: none"> – Fall 1: aktuelle Position auf leerem Feld, – Fall 2: aktuelle Position auf besetztem Feld, – Fall 3: Ende einer Reihe in Richtung links nach rechts, – Fall 4: Ende einer Reihe in Richtung rechts nach links, – Fall 5: Ende der letzten Reihe erreicht. <p>Darstellung der funktionalen Hierarchie. Es sind mehrere weitere sinnvolle funktionale Zerlegungen denkbar.</p> <div style="text-align: center;"> <pre> graph TD main[main] --> fueH[fuelleHin] main --> fueR[fuelleRueck] fueH --> rechts[rechtsUm] fueH --> fueRe[fuelleReihe] fueR --> fueRe fueRe --> teste[testeGib] </pre> <p>The diagram is a hierarchical dependency chart. At the top is a box labeled 'main'. A line from 'main' branches into two boxes: 'fuelleHin' on the left and 'fuelleRueck' on the right. From 'fuelleHin', a line branches into two boxes: 'rechtsUm' on the left and 'fuelleReihe' in the center. From 'fuelleRueck', a line branches into one box: 'fuelleReihe' in the center. From 'fuelleReihe', a line leads down to a box labeled 'testeGib'.</p> </div> <p>Abb. 1: Hierarchische Abhängigkeitsübersicht</p>	<p>I 10 %</p> <p>II 20 %</p>

<p>2</p>	<p>Die Schülerinnen und Schüler stellen alle Funktionseinheiten in einer visuellen/ formalen Darstellung wie folgt dar:</p> <p>Abbildung 2: Struktogrammform mit Bezug zur Microwelt HaSE</p>	<p>II/III 50 %</p>
<p>3</p>	<p>Die Schülerinnen und Schüler testen am PC mit jeweils 2 verschiedenen Verteilungen der Elemente:</p> <ul style="list-style-type: none"> – ein Szenario mit 1 Zeile, – ein Szenario mit ungeradzahliger Anzahl von Zeilen, – ein Szenario mit geradzahliger Anzahl von Zeilen, – Speichern aller Testszenarien. 	<p>I 20 %</p>