ALGORITHMEN

UNTERRICHTSGANG

MIT

HINTERGRUNDINFORMATIONEN

Thomas Schaller – E-Mail: schaller@mlg-bad.de
Monika Eisenmann – E-Mail: eisenmann.schule@email.de. – April 2018

Inhaltsverzeichnis

Hintergrund....3
Programmiersprache / Programmierumgebung....3
Unterrichtsgang....5
Wiederholung Klasse 8....5
Erstes Kennenlernen der Entwicklungsumgebung....6
Zählschleifen in Java – Figur C mit Schleife....7
Kaffeehausillusion – die erste optische Täuschung....7
Wellen – weitere optische Täuschungen....8
Highscorediagramm Teil1: Anzeige eines Balkendiagramms....9
Highscorediagramm Teil 2: Füllen des Arrays mit Zufallszahlen....9
Highscorediagramm Teil 3: Einlesen der CSV-Datei....9
Highscorediagramm Teil 4: Maximumsuche....10
Highscoretabelle Teil 5: Zeitungsmeldung....12
Highscoretabelle Teil 6: Bubblesort....12
XY-Diagramm einer Schallpegelmessung....13
Literaturliste und Internetseiten:....15

Hintergrund

Ziel dieser Einheit ist es, dass die Schülerinnen und Schülern eine textuelle Programmiersprache erlernen. Dabei werden die in der blockbasierten Programmiersprache eingeführten Konzepte auf die textuelle Programmiersprache übertragen.

Programmiersprache / Programmierumgebung

Grundsätzlich schreibt der Bildungsplan weder eine bestimmte Programmiersprache noch eine bestimmte Programmierumgebung vor. Die Wahl bleibt dem Lehrer überlassen und sollte nach didaktischen Gesichtspunkten gefällt werden:

- Ist ein einfacher Einstieg ohne viel Overhead möglich?
- Wird die Suche nach Fehlern gut durch Anzeige von Syntaxfehlern und einen guten Debugger unterstützt? Wird der Quelltext strukturiert dargestellt?
- Trägt die Programmiersprache/Programmierumgebung langfristig, d.h. kann sie auch in den kommenden Schuljahren eingesetzt werden?
- Gibt es andere didaktische Tools für die eingeführte Programmiersprache, die im weiteren Unterrichtsverlauf hilfreich sein können?

In diesem Unterrichtsvorschlag kommt die **Programmiersprache Java** zum Einsatz, da die Zahl der didaktischen Hilfsmittel für Java zur Zeit am größten ist. Java kann als objektorientierte Programmiersprache bis zum Abitur fortgeführt werden.

Einen Überblick über die schon bekannten oder jetzt hinzukommenden Kontrollstrukturen in Java im Vergleich zum MIT App Inventor 2 oder zu Scratch bieten die beiden Dokumente im Ordner Hintergrund: *03_alg_vergleich_ai_java.odt* bzw. *03_alg_vergleich_scratch_java.odt*. Sie sind für Sie als Lehrkraft gedacht, können aber auch am Ende der Einheit den Schülerinnen und Schülern zur Verfügung gestellt werden.

Um den Einstieg zu erleichtern, schlägt dieser Unterrichtsgang vor, graphische Ausgaben (optische Täuschungen und Balken- und XY-Diagramme) zu erzeugen, da man dort die Wirkung der eingesetzten Befehle direkt sieht. Um graphische Ausgaben zu erzeugen, eignet sich die Java-Variante **Processing**[1], die eine Vielzahl von Zeichenbefehlen mitbringt. Ergänzt wird dies durch einfach zu verwendende Bibliotheken für den Zugriff auf CSV- und XML-Dateien. Es ist in Processing ohne Rahmenprogramm, ohne Deklaration von Klassen und ohne aufwändige Definition einer GUI möglich, auf einer Zeichenfläche zu zeichnen. Textergänzungen (Strg+Leerzeichen) lassen sich in den Einstellungen aktivieren. Das Animieren von Algorithmen ist sehr einfach. Der Einstieg fällt den Schülerinnen und Schülern daher sehr leicht[2]. Leider ist die Unterstützung der Schüler durch Syntaxhervorhebung von Schleifen und Entscheidungen und den einfachen gehaltenen Debugger nicht so effektiv wie bei anderen Programmierumgebungen. GUIs lassen sich zwar mit zusätzlichen Tools erzeugen, sind aber keine vollwertigen Swing oder JavaFX-Oberflächen.

Als zweite Variante ist es möglich, **Greenfoot**[3] einzusetzen. Durch eine spezielle Bibliothek können viele der in Processing zur Verfügung stehenden Befehle auch in Greenfoot eingesetzt werden. Dort wird eine von Actor abgeleitete Zeichenfläche eingesetzt, deren Bild durch die Zeichenbefehle verändert werden kann. Schön ist, dass man durch das Kontextmenü der Zeichenfläche direkt Zeichenbefehle geben und deren Wirkung beobachten kann. So kann man die Befehle direkt ausprobieren. Texteditor und Debugger unterstützen die Schülerinnen und Schüler effektiv bei der Arbeit. Nachteilig ist, dass das

1 Programmierumgebung Processing: *https://processing.org/* (Stand: 07.02.2019)
2 Processing lässt sich durch einen Android-Modus ergänzen. (Siehe *04_alg_processing_android.odt*)
3 Programmierumgebung Greenfoot: *https://www.greenfoot.org/door* (Stand: 07.02.2019)

Zeichnen sehr langsam ist, da jedes Mal das Bild des Actors verändert werden muss. Dies bremst die Ausführung der Algorithmen bei größeren Datenmengen (ca. 10000 Datensätze beim XY-Diagramm) erheblich aus. Greenfoot bietet keine Möglichkeit, GUIs zu erzeugen, da das dem Konzept von Greenfoot widerspricht. Für die Erzeugung einer graphischen Oberfläche (vgl. IMP Klasse 10) muss daher zu einem anderen Tool gewechselt werden.

Die dritte Möglichkeit ist der Einsatz des **Java-Editors**[4] von Gerhard Röhner. Der Java-Editor bietet eine gute Unterstützung der Schülerinnen und Schüler durch farbliche Hervorhebung von Schleifen und Entscheidungen. Auch der Debugger kann effektiv eingesetzt werden. Mit dem Java-Editor können sehr gut GUIs erzeugt werden (vgl. IMP Klasse 10). Auch die Darstellung von Klassen und Objekten und das direkte Arbeiten in der UML-Darstellung ist überzeugend (ähnlich wie bei BlueJ; vgl. Bildungsplan Kursstufe). Durch eine geeignete Bibliothek stehen auch hier viele der Processing-Befehle zur Verfügung. Der Nachteil ist, dass für die Anzeige eines graphischen Ausgabefensters eine GUI notwendig ist. Diese kann den Schülern zwar vorgegeben werden, so dass sie sie nicht selbst erstellen müssen, erhöht aber die Anzahl an Java-Klassen selbst für ein einfaches Programm. Die Arbeit der Schülerinnen und Schüler beschränkt sich aber auf eine einzige übersichtliche Klasse. Etwas komplizierter ist die Animation von Algorithmen, da hier die Programmierung von Threads notwendig ist, um eine Aktualisierung der Ausgabe während der Ausführung des selbst programmierten Algorithmus zu ermöglichen. Auch dieses muss den Schülerinnen und Schülern vorgegeben werden.

Da keines der drei Tools alle gewünschten Eigenschaften auf sich vereint, bleibt die Wahl Ihnen als Lehrkraft überlassen. Die vorliegenden Materialien sind so gestaltet, dass die Arbeitsblätter für alle drei Programmierumgebungen eingesetzt werden können. Die Vorlagen, mit denen die Schülerinnen und Schüler arbeiten sollen, stehen für alle drei Umgebungen bereit. Die wesentlichen Inhalte der Lösungen unterscheiden sich dabei nicht. Sie müssen den Schülern aber den Umgang mit der Programmierumgebung beibringen. Dies wird nicht auf den Arbeitsblättern erklärt.

Nutzung der Programmierumgebungen

Processing: Hier muss lediglich die entsprechende pde-Datei in Processing geöffnet werden. Globale Variablen/Attribute werden vor allen Methoden deklariert. Die Methode setup() wird automatisch ausgeführt und sollte die Methode zum Zeichnen aufrufen. Alternativ kann die Zeichenmethode aus der vordefinierten Methode draw() aufgerufen werden. Diese wird von Processing in regelmäßigen Abständen ausgeführt (vgl. Animationen).

Schriftarten müssen (vom Lehrer) vorab generiert werden (Menü Tools) und liegen danach als .vlw-Datei in dem Verzeichnis data. Sie werden am sinnvollsten im Setup mit dem Befehl `loadFont("KleineSchrift.vlw")` geladen.

Greenfoot: Öffnen Sie das Greenfoot-Projekt. Jede Aufgabe liegt in einem eigenen Verzeichnis. Die Schülerinnen und Schüler müssen jeweils eine Unterklasse von PictureViewer bearbeiten. Globale Variablen/Attribute werden innerhalb der Klasse vor allen Methoden deklariert. Der Konstruktor wird automatisch ausgeführt und ruft die Methode zum Zeichnen auf. Die Welt erzeugt automatisch ein Objekt der jeweiligen Klasse, so dass sofort die Zeichenfläche angezeigt wird.

Die Dokumentation der Klassen Picture, PictureViewer, Table, TableRow, XML erhält man in Greenfoot am einfachsten, indem man den Quelltext öffnet und dann im Editor rechts oben von "Quelltext" auf "Dokumentation" umschaltet. Alternativ steht die Dokumentation im Hintergrund-Ordner als HTML-Dateien bereit.

4 Programmierumgebung Java-Editor: *http://javaeditor.org/doku.php* (Stand 07.02.2019)

Java-Editor: Für die Nutzung der Klassen Table, TableRow und XML ist die Bibliothek `jdom-1.1.3.jar` erforderlich. Daher muss dem Java-Editor mitgeteilt werden, dass man diese Bibliothek benutzen möchte. Das macht man unter Menü Fenster → Konfiguration → Java → Interpreter: ClassPath-User. Dort wählt man die jdom-Bibliothek als neue Jar-Datei aus.

Danach öffnen Sie die zwei Dateien XYGUI.java und XY.java (XY steht für den Namen der Aufgabe: z.B. Balkendiagramm.java). Die GUI muss geöffnet sein, damit der Java-Editor das Projekt korrekt starten kann. An dieser Datei müssen/dürfen die Schülerinnen und Schüler nichts ändern. Sie bearbeiten lediglich die XY.java, die eine Unterklasse von Picture ist. Globale Variablen/Attribute werden innerhalb der Klasse vor allen Methoden deklariert. Der Konstruktor wird automatisch aufgerufen und ruft die Methode zum Zeichnen auf. Die GUI enthält einen Button mit dem ein Objekt der jeweiligen Klasse erzeugt und im Zeichenfenster angezeigt wird.

Die Dokumentation der Klassen Picture, PictureViewer, Table, TableRow, XML erhält man im Java-Editor, indem man die Quelltexte alle öffnet und dann im Editor Menü Start → JavaDoc wählt. Da die Dateien in UTF-8 Format vorliegen, muss man vorher noch unter Menü Fenster → Konfiguration → Java → Programme die Parameter "`-encoding UTF-8 -charset UTF-8 -docencoding UTF-8`" ergänzen. Der Java-Editor erzeugt dann ein Unterverzeichnis "doc", das die Dokumentation enthält. Alternativ steht die Dokumentation im Hintergrund-Ordner schon als HTML-Dateien bereit.

Unterrichtsgang

Die meisten der vorliegenden Materialien wurden in drei neunten Klassen getestet. Die Klassen waren vollständig, da es sich an der Testschule um eine Poolstunde für Informatik handelt, die alle Schülerinnen und Schüler verpflichtend in ihrem Plan haben. Das Interesse am Fach ging deshalb sehr weit auseinander und ständige Differenzierungsmaßnahmen waren unabdingbar.

Als Programmierumgebung wurde bei der Erprobung Processing verwendet. Die Materialien wurden im Nachhinein so angepasst, dass sie auch in den beiden anderen Programmierumgebungen Greenfoot und dem JavaEditor nutzbar sind.

An manchen Stellen macht es Sinn, spezifische Eigenschaften der Programmierumgebung noch zu thematisieren oder sogar in das Material einzufügen.

Wie schon erwähnt, sind die Vorlagen in allen drei Varianten vorhanden. Bei der Arbeit mit dem JavaEditor oder mit Greenfoot muss der Code, der in Processing direkt im Editor geschrieben werden kann, in der Datei *Zeichnung.java* ergänzt werden.

Die Lösungen zum ersten Teil liegen in der Processing-Variante vor. Wird eine der anderen Umgebungen im Unterricht verwendet, müssen die Inhalte der setup()-Methode in den Konstruktor der Zeichnung-Klasse und die selbstgeschriebenen Methoden darunter kopiert werden.

Der Unterrichtsgang ist in zwei Teile aufgeteilt. Im ersten werden nach einer Wiederholung der Kontrollstrukturen aus den Klassen 7 und 8 anhand von optischen Täuschungen die grundlegenden Strukturen (Schleifen, Verzweigungen), Variablen, Arrays und die Benutzung von Unterprogrammen eingeführt.

Im zweiten Teil steht der Umgang mit Arrays im Vordergrund. Einige Standardalgorithmen auf Arrays sollen eingeübt werden. Als Aufhänger wird die Darstellung einer Highscore-Tabelle herangezogen. Zur Vertiefung der Algorithmen wird dann noch zur Darstellung eines XY-Diagramms gewechselt.

Damit den Schülerinnen und Schüler spannende Daten zur Verfügung stehen, spielen sie

zunächst ein „Binärspiel“ (Umrechnung von Binärzahlen in Dezimalzahlen). Die Ergebnisse werden auf einen Server übertragen und in einer gemeinsamen CSV-Datei zusammen mit dem Spielernamen gespeichert. Ziel ist es, daraus ein Balkendiagramm zu erzeugen, das die Ergebnisse nach den Punkten absteigend anzeigt.

Für das XY-Diagramm können Lärmpegeldaten aus dem Unterricht herangezogen werden. Mit geeigneten Apps können diese Daten leicht aufgezeichnet und als CSV-Dateien abgespeichert werden.

Wiederholung Klasse 8

Inhalt: Erteilen und Umsetzen von Anweisungen, Wiederholung der Kontrollstrukturen aus Scratch und dem Begriff des Algorithmus

Einstieg in die Stunde ist eine Gruppenarbeit zu Algorithmen aus Computer Science Unplugged[5]. Für jede Gruppe liegen am Pult die gleichen Bilder bereit. Die Bilder und der Auftrag für die Gruppen ist in der Datei *01_alg_wdh_begriffe.odt* zu finden. Jeweils ein Gruppenmitglied holt sich ein Bild und erklärt es den anderen, die es zeichnen müssen. Nachdem alle fertig gezeichnet haben, werden die Bilder mit dem Original verglichen und es werden Schwierigkeiten besprochen und evtl. notiert. Es genügt, pro Gruppe zwei bis drei Figuren zeichnen zu lassen.

Die Lehrkraft beobachtet die unterschiedlich geschickten Erklärversuche und notiert sich ggf. elegante oder verbesserbare Methoden für die Auswertung der Gruppenarbeit.

Nach der Auswertung im Plenum werden die schon in Klasse 7 kennen gelernten Kontrollstrukturen und der Begriff des Algorithmus anhand eines Arbeitsblattes (*02_alg_kontrollstrukturen_wdh.odt*) wiederholt.

Falls noch Zeit bleibt, kann man das kahoot-Quiz[6] aus Klasse 8 zur Wiederholung der Algorithmen in Scratch spielen. Bei einer Doppelstunde bietet es sich an, das aufs Ende zu verschieben. Sollte keine Zeit bleiben, können Sie das Quiz als Challenge bis zur nächsten Stunde als Hausaufgabe geben.

Erstes Kennenlernen der Entwicklungsumgebung

Inhalt: Erste Begegnung mit der Entwicklungsumgebung, Kennenlernen erster Zeichenmethoden, Schreiben des ersten kleinen Programms

Zu Beginn dieser Stunde sollte man den Schülerinnen und Schülern bewusst machen, dass sie auch bisher schon selbständig Programme entwickelt haben (in Scratch oder dem MIT App Inventor), dass sie aber bisher noch keine Programmzeile selbst schreiben mussten.

Das war auf der einen Seite ein großer Vorteil, weil sie keine Schreibfehler machen konnten, auf der anderen Seite konnten sie aber nur das in ihren Programmen umsetzen, was durch die gegebenen Blöcke realisierbar war.

Um mehr Möglichkeiten zu haben, lernen die Schülerinnen und Schüler ab Klasse 9 in IMP eine Programmiersprache kennen. Die Materialien sind, wie oben schon beschrieben, für Java entwickelt.

Bevor die Schülerinnen und Schüler mit den ersten Aufträgen starten, zeigen Sie ihnen die von Ihnen gewählte Entwicklungsumgebung (Grundlagen wie Öffnen und Speichern einer Datei, wo schreibe ich meine Programmzeilen, wie teste ich das Programm, wie schreibe ich Kommentare, etc.).

5 *https://csunplugged.org/de/* (abgerufen am 21.03.2019)

6 Website: kahoot.com – Suche nach „Algorithmen in Scratch“, Autorin: eisenmann_ggk

Die Schülerinnen und Schüler bearbeiten dann das Arbeitsblatt aus der Datei *03_alg_keine_bloecke_mehr.odt*. Sie brauchen dazu die beiden Dateien *wasmacheich* und *figuren* im Tauschverzeichnis.

Die Schülerinnen und Schüler schauen sich die erste Datei an und versuchen herauszufinden, was die einzelnen Werte in den aufgerufenen Methoden bewirken. Dazu verändern sie die Werte, überlegen, was es sein könnte und testen ihre Vermutungen.

Es bietet sich an, die Ergebnisse kurz gemeinsam zu besprechen, bevor die weiteren Aufgaben gemacht werden. In der vorliegenden Datei ist noch eine Übersicht über die wichtigsten Zeichenmethoden enthalten, die hier den Schülerinnen und Schülern geeignet zur Verfügung gestellt werden sollte.

Bei der Programmierung einer Methode, die Figur C aus der ersten Stunde zeichnet, ist noch nicht an die Nutzung einer Schleife gedacht. Das folgt erst in der nächsten Doppelstunde. Hier geht es zunächst einmal darum, die Zeichenmethoden einzusetzen und vor allem durch Überlegung und Ausprobieren, die Positionen von Figuren auf dem Fenster bewusst vorzugeben. Das fällt einigen zunächst schwer. Machen Sie den großen Nutzen einer Skizze klar!

Für Schnelle bzw. als Hausaufgabe sind die letzten beiden Aufgaben des Blattes gedacht.

Zählschleifen in Java – Figur C mit Schleife

*Inhalt für eine **Doppelstunde**: Zählschleifen in Java*

Die Schülerinnen und Schüler, die in Klasse 8 in IMP mit dem MIT App Inventor gearbeitet haben, kennen die Zählschleife schon mit einer Zählvariable. Vielleicht haben sie auch schon Programme entwickelt, bei denen sie diese Zählvariable in der Schleife genutzt haben.

Bei dem Thema ist es wichtig, dass das Grundverständnis bei allen Schülerinnen und Schülern da ist. Es bietet sich hier an, gemeinsam für die aus der vergangenen Stunde programmierte Figur C, eine Version mit Zählschleife zu entwickeln.

Wer möchte, kann hier die Kopiervorlage (letzte Seite der Datei *04_alg_zaehlschleife.odt*) als Folie oder digital nutzen. Im Video *zaehlschleife_aktiv.mp4* (im Ordner 05_Präsentationen) wird eine Möglichkeit gezeigt, wie das Ganze mit den Schülerinnen und Schülern entwickelt werden kann.

Den zweiten Teil von Figur C probieren die Schülerinnen und Schüler dann eigenständig. Die Aufgaben dazu sind auf dem Blatt in der Datei von oben. Zur Unterstützung sind Hilfekarten vorhanden, die Sie geeignet zur Verfügung stellen sollten.

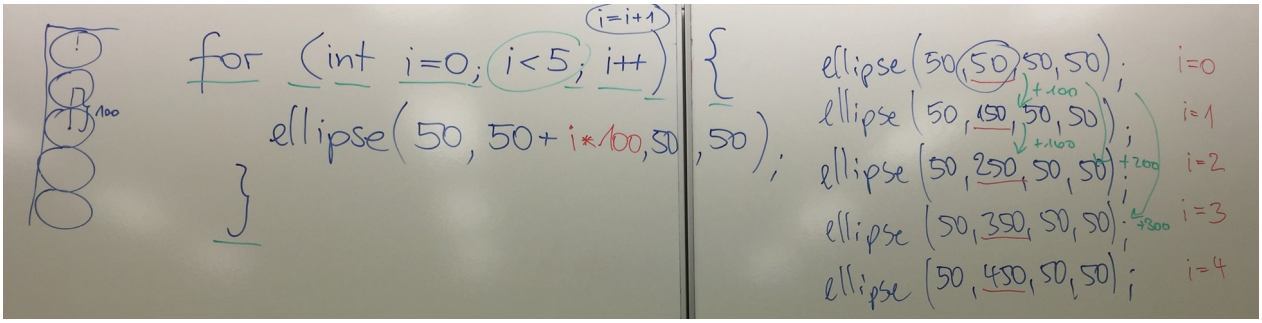
Die weiteren Übungen mit Differenzierungsmöglichkeiten sollen den Umgang mit Zählschleifen trainieren und vertiefen und können auch als Hausaufgabe aufgegeben werden, wenn die Zeit im Unterricht nicht mehr ausreicht.

Für die Differenzierungsaufgaben braucht man schon die Information, wie Methoden mit Übergabeparametern in Java geschrieben werden. Hier kann das Informationsblatt (*04_alg_infos_methoden.odt*) ausgegeben werden, das dann im späteren Verlauf der Unterrichtseinheit auch alle bekommen sollen.

Kaffeehausillusion – die erste optische Täuschung

*Inhalt für eine **Doppelstunde**: Kurze Wdh. Zählschleife, Kommentare, verschachtelte Zählschleifen, Felder (Arrays)*

Zum Einstieg zeichnen Sie ein Bild an die Tafel (z.B. 5 Kreise untereinander (siehe Tafelbild)), entwickeln dann gemeinsam mit den Schülerinnen und Schülern die zugehörigen Programmzeilen – zunächst ohne, dann mit Zählschleife.



Im Anschluss daran ermöglicht ein Wettpuzzeln zweier Gruppen gegeneinander die Wiederholung der Zählschleifen in Java. Sieben Bilder und dafür alle nötigen Schleifenteile müssen jeweils zusammengefunden werden (*04_alg_zaehlschleife_puzzle.odt*).

Nach dieser Wiederholung starten Sie mit ihren Schülerinnen und Schülern mit der ersten optischen Täuschung.

Sämtliche Materialien (Aufgaben und Lösungen, die ausgelegt werden sollten) sind in der Datei *05_alg_optische_taeuschungen_kaffeehaus.odt* zu finden.

Die Schülerinnen und Schüler arbeiten in ihrem eigenen Tempo und werden von der Lehrkraft individuell unterstützt.

Es bietet sich an, bei Schwierigkeiten eine gemeinsame Phase einzuschieben, in der der Aufbau der verschachtelten Zählschleife kurz besprochen wird, damit alle das Prinzip verstanden haben und es nutzen können.

Es hat sich gezeigt, dass Schülerinnen und Schüler am besten mit verschachtelten Zählschleifen klarkommen, wenn sie sich zunächst die innere Schleife überlegen. Im vorliegenden Fall schreiben sie also zunächst die Schleife für eine Reihe von acht Quadraten.

Erst in einem zweiten Schritt wird dann überlegt, was sich jetzt von Reihe zu Reihe ändert und wie diese Änderung durch die zweite Zählvariable der äußeren Zählschleife realisiert werden kann.

Auch bei der Interpretation von Programmzeilen macht es Sinn, zunächst die innere Schleife anzuschauen und erst dann zu überlegen, was bei der Wiederholung des Inneren noch passiert.

Als Übung und Vertiefung gibt es zwei weitere optische Täuschungen (*06_alg_optische_taeuschungen_uebungen.odt*), die mit einer verschachtelten Zählschleife programmiert werden können. In der Erprobung haben starke Schülerinnen und Schüler damit schon im Laufe der Doppelstunde begonnen. Für beide Aufgaben stehen Hilfekarten zur Verfügung, so dass die Schülerinnen und Schüler wieder eigenständig in ihrem Tempo arbeiten können.

Die beiden Übungen eignen sich auch als Hausaufgabe.

Wellen – weitere optische Täuschungen

*Inhalt für eine **Doppelstunde**: Methoden, Verzweigungen, bedingte Schleife (while-Schleife)*

Zum Einstieg bietet sich wieder eine kurze Wiederholung – dieses Mal der verschachtelten

Zählschleife – an. Dazu schreiben Sie z.B. den Java-Code für das Einmaleins an die Tafel und lassen ihre Schülerinnen und Schüler überlegen, welche Werte berechnet werden. Besprechen Sie hier nochmal das Vorgehen: zuerst innere Schleife betrachten, dann überlegen, was durch die äußere Schleife noch passiert.

Im Material (*07_alg_optische_taeuschungen_up_verzw.odt*) werden anhand einer weiteren optischen Täuschung Methoden genauer betrachtet (das Infoblatt hierzu konnte von einzelnen Schülerinnen und Schülern schon bei den Aufgaben zu Zählschleifen geholt und genutzt werden – alle anderen bekommen es hier: *04_alg_infos_methoden.odt)* und Verzweigungen genutzt.

Die Blätter sind wieder so gestaltet, dass die Schülerinnen und Schüler eigenständig die optische Täuschung entwickeln können. Es ist aber auch ein zunächst gemeinsames Vorgehen möglich. Im Anschluss daran können dann die restlichen Aufgaben bearbeitet werden. Entscheiden Sie hier so, wie es zu ihrer IMP-Gruppe am besten passt.

Zur Übung und Vertiefung und zur Einführung der „while“-Schleife ist die nächste optische Täuschung gedacht. Eine weitere ist als Differenzierungsmöglichkeit angehängt.

Für das Beispiel zur Einführung der bedingten Schleife finden sich in der Materialdatei zur Doppelstunde Hilfekarten und eine Seite mit Informationen zur neuen Schleifenart.

An dieser Stelle oder auch zu einem späteren Zeitpunkt können sie ein weiteres kahoot-Quiz[7] spielen oder als Challenge herausgeben.

Im Material finden Sie noch weitere Übungen (zwei davon nutzen spezielle Processing-Möglichkeiten) in der Datei *08_alg_optische_taeuschungen_uebungen.odt*.

Highscorediagramm Teil1: Anzeige eines Balkendiagramms

Inhalt: Umgang mit Arrays, Unterscheidung Index/Wert an einer Position, Abarbeitung von Arrays mit Hilfe von for-Schleifen, gezieltes Testen

Starten Sie mit der Präsentation *10_arrays.odp*. Dort werden verschiedene Einsatzbereiche für Arrays vorgestellt. Einen davon sollen die Schüler bearbeiten: die Highscoretabelle. Dazu werden zunächst mit Hilfe des Highscore-Spiels (Ordner *06_software/01_highscore-spiel*) Daten generiert. Starten Sie das Serverprogramm auf dem Lehrerrechner und aktivieren Sie den Server. Teilen Sie den Schülerinnen und Schülern den Namen oder die IP-Adresse des Lehrerrechners mit, damit sie sich mit dem Server verbinden können. Sie sehen im linken Fenster, wer mit dem Server verbunden ist. Lassen Sie das Spiel einige Minuten spielen, damit genügend Datensätze zusammenkommen. Speichern Sie die Daten als CSV-Datei (Trennung mit Kommas) ab. Stellen Sie die Datei den Schülern für die Aufgaben ab "Einlesen der CSV-Datei" zur Verfügung.

Um den Schülerinnen und Schülern den Zugang zu erleichtern, wird nicht sofort mit der CSV-Datei gearbeitet. Zunächst wird der Umgang mit Arrays an einem vorgegebenen mit Zahlen gefüllten Array wiederholt und eine for-Schleife zur Abarbeitung eines ganzen Arrays eingesetzt. Die Schülerinnen und Schüler müssen erkennen, wie man eine sequentielle Ausführung der Befehle für die einzelnen Array-Elemente in eine Schleife umwandeln kann. Es müssen die sich verändernden Koordinaten bei der Ausgabe identifiziert werden und durch Terme zur Berechnung der Koordinaten in Abhängigkeit von der Zählvariable der Schleife ersetzt werden. Abschließend werden die Schülerinnen und Schüler dazu angehalten, ihr Programm gezielt auf Fehler zu testen, die in speziellen Fällen (z.B. negative Punkte, 0 Punkte) auftreten könnten.

7 Website: kahoot.com – Suche nach „IMP 9 – Java-Grundlagen“ - Autorin: eisenmann_ggk

Highscorediagramm Teil 2: Füllen des Arrays mit Zufallszahlen

Inhalt: Zufallszahlen, gezieltes Testen

Das automatische Füllen des Punktearrays mit Zufallszahlen lässt vielfältigere Werte zu. Die Schülerinnen und Schüler üben dabei den Umgang mit Zufallszahlen. Diese werden von einer geeigneten Methode erzeugt[8]. Das Array lässt sich nun leicht in der Größe variieren. Es zeigt sich, ob die Implementierung der Darstellung des Balkendiagramms auch für andere Zahlen und unterschiedliche Arraygrößen trägt.

Highscorediagramm Teil 3: Einlesen der CSV-Datei

Inhalt: Umgang mit Textdateien, Arrays verschiedener Typen

Die angezeigten Daten sollen schlussendlich aus den gespeicherten Spielergebnissen bestehen. Dazu wird an dieser Stelle der Umgang mit Textdateien behandelt. Processing stellt eine Bibliothek Table / TableRow zur Verfügung, die einen einfachen Zugriff auf CSV-Dateien ermöglicht. Diese Bibliothek wurde für den JavaEditor und Greenfoot übertragen und kann dort ähnlich eingesetzt werden. Nur das Erzeugen eines Table-Objekts (incl. Laden der Datei) unterscheidet sich. Daher wird dieser Befehl vorgegeben. Um im Java-Editor die Bibliothek nutzen zu können, ist es erforderlich, die JAR-Datei „jdom-1.1.3.jar“ bei Classpath-User unter Fenster>Konfiguration>Java>Interpreter auszuwählen.

Neben den Punkten, die in einem Integer-Array gespeichert werden, müssen hier zusätzlich die Spielernamen in einem String-Array gespeichert werden. Das übt den Umgang mit Arrays von verschiedenen Typen.

Highscorediagramm Teil 4: Maximumsuche

Inhalt: Methoden mit Rückgabewert, Array-Algorithmus Maximumsuche, automatisches Testen

Nachdem die Punktewerte aus der externen Datei eingelesen sind, soll nun der beste Spieler oder die beste Spielerin ermittelt werden. Es wird also das Maximum der Punktewerte gesucht. Die Funktion könnte entweder den größten Wert oder die Position des größten Wertes im Array zurückgeben. Es ist sinnvoller die Position zurückzugeben, da damit der Wert leicht ermittelt werden kann. Umgekehrt ist das nicht möglich.

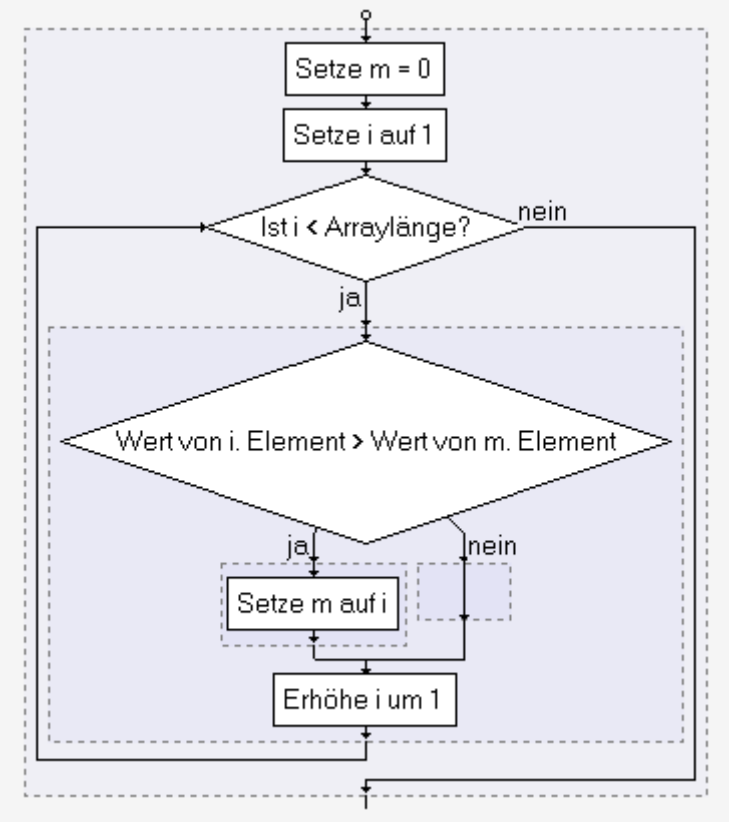
Der Algorithmus ist der erste einer Reihe von Algorithmen, die alle darauf basieren, dass das Array mit einer for-Schleife durchlaufen werden muss. Der Algorithmus müsste den Schülern aus der Klasse 8 bekannt sein, da er dort behandelt wurde. Dort wurde allerdings der Wert des Maximums ermittelt und nicht die Position im Array[9]:

8 Auch im Informatikabitur werden Zufallszahlen auf diese Weise erzeugt, um unabhängig von der Programmiersprache zu sein.

9 Scratch wird von der Lifelong-Kindergarten-Group am MIT-Media-Lab entwickelt. Siehe *http://scratch.mit.edu.* Scratch ist lizenziert unter *CC BY-SA 2.0* (*https://creativecommons.org/licenses/by-sa/2.0/deed.en*).

Abbildung 1: Maximumsuche in Scratch

Um den Algorithmus zu wiederholen, kann der Film *maximumsuche.mov* (Ordner 05_praesentationen) gezeigt werden. Die SuS sollten selbstständig erkennen, welches Array-Element ausgewählt wurde. Mit Hilfe des Puzzles (*13_alg_maximumsuche_puzzle.odt*) kann der Algorithmus als Flussdiagramm gelegt werden. Dabei müssen die Verzweigung und die Schleife zerschnitten und die Pfeile von Hand eingezeichnet werden.



Die Umsetzung in der Programmiersprache erfolgt in einer eigenen Methode `int maximumsuche()`. Eindrucksvoll ist es, wenn man den Verlauf der Ausführung verfolgen kann. Das jeweils aktuell untersuchte Element und das aktuell maximale Element sollen farblich

hervorgehoben werden. Dazu muss zunächst die graphische Darstellung angepasst werden. Die Methode `maximumsuche()` muss dann parallel zu der graphischen Darstellung arbeiten und durch eine entsprechende Verzögerung in angemessener Geschwindigkeit ausgeführt werden:

- In **Processing** gibt es den Befehl thread(String methodenname), der sehr einfach einen parallelen Thread startet. Durch den Befehl delay(int millisekunden) kann die Ausführung verzögert werden. Damit das Bild neu gezeichnet wird, muss außerdem der Befehl redraw() aufgerufen werden. Dadurch wird die Methode draw() von Processing automatisch neu ausgeführt. In dieser muss die Methode zeichneBalken aufgerufen werden.
- In **Greenfoot** wird bei einem Greenfoot.delay(int millisekunden) die Ausführung automatisch an Greenfoot zurückgegeben. In dieser Zeit arbeitet Greenfoot ausstehende Anzeigebefehle ab. Es ist also kein paralleler Thread erforderlich.
- In normalem Java (z.B. Entwicklungsumgebung: **JavaEditor**) muss auch ein Thread gestartet werden. Die Syntax ist nicht ganz einfach, so dass die SuS dies nicht selbst programmieren sollten.

Da es sich hierbei um Spezialitäten der jeweiligen Umgebung handelt, wird den Schülerinnen und Schülern der Rahmen vorgegeben. Sie müssen lediglich in der Methode `maximumsuche()` die notwendigen Befehle aufrufen. Die Hinweise dazu stehen im Quelltext der jeweiligen Programmiersprache.

In der Softwareentwicklung gibt es die Methode der "testgetriebenen Entwicklung". Dabei werden die Tests der Methoden geschrieben, bevor die Implementation der Methoden selbst erfolgt. In jedem Entwicklungsschritt wird überprüft, ob die Tests immer noch korrekt ausgeführt werden. Dies ist auch für die Schule eine gute Möglichkeit, die korrekt Implementation von Schülern selbst testen zu lassen. Der Lehrer gibt dabei die Tests vor. In BlueJ gibt es dafür die Möglichkeit von JUnit-Tests. In Processing, Greenfoot und Java-Editor gibt es diese Möglichkeit nicht. Man kann aber leicht eigene Tests konstruieren, deren Ergebnisse in der Konsole ausgegeben werden. Dies ist hier realisiert. Es werden verschiedene Datensätze zur Kontrolle der Maximumsuche verwendet. Die Schüler starten die Tests durch

- Processing: Drücken der Taste "t"
- Greenfoot: Aufruf der Methode testen() im Kontextmenü
- Java-Editor: Klick auf den Button "Testen"

Zur Übung können die Schüler auch die Minimumsuche implementieren. Sie sollten dabei vor der Implementation der Minimumsuche die Tests entsprechend anpassen.

Highscoretabelle Teil 5: Zeitungsmeldung

Inhalt: Methoden der Fehlersuche, weitere Array-Algorithmen: Durchschnitt, Anzahl unterschiedlicher Array-Elemente

In Zeitungen werden heutzutage immer mehr Meldungen automatisch von Algorithmen geschrieben[10]. Diese Situation wird hier in einem einfachen Programm nachgestellt. Die Daten der Spielergebnisse werden automatisch ausgewertet und in einem Text angezeigt.

Der Algorithmus zur Berechnung der Summe ist dabei noch fehlerhaft. Die Schüler sollen gezielt mit den zur Verfügung stehenden Möglichkeiten (Ausgabe von Variablenwerten in der Konsole und der Einsatz des Debuggers) die Fehler finden. Führen Sie den Schülern die

10z.B. Roboter-Journalismus - Wenn Maschinen berichten, Joao da Mata, TAZ vom 21.04.2018, URL:http://www.taz.de/!166317/ (Stand: 07.02.2019)

Verwendung des Debuggers und der Konsolenausgabe an dieser Stelle vor.

Einer der Fehler ist die typische Verwechslung von Index und Wert eines Arrayelements. Achten Sie hierauf im Unterricht besonders.

Die Bestimmung der Anzahl der Spieler mit unterschiedlichem Namen ist nicht einfach, da der aktuelle Name jeweils mit allen vorangegangenen verglichen werden muss. Diese Aufgabe ist als Ergänzungsaufgabe gedacht.

Highscoretabelle Teil 6: Bubblesort

Inhalt: Dreieckstausch, Array-Algorithmen: Bubblesort, Untersuchung von Algorithmen

Die Schülerinnen und Schüler untersuchen zunächst den Dreieckstausch, der für Sortieralgorithmen auf Arrays notwendig ist. Sie sollen dazu ohne Computer den Quelltext analysieren. Gegebenenfalls können sie mit dem angegebenen Film eine schöne Veranschaulichung betrachten, bei der man sieht, dass die notwendigen Überlegungen keine Selbstverständlichkeit sind und dem Vogel Schwierigkeiten bereiten.

Danach wird die Untersuchung auf den ganze Algorithmus erweitert. Die Schülerinnen und Schüler sollen analysieren, welche Änderung am Array er bewirkt. Es handelt sich um einen Durchgang des Bubblesorts, der das kleinste Element an das Ende befördert. Die Schülerinnen und Schüler sollen erkennen, dass n-1 Durchgänge notwendig sind, um auf jeden Fall ein vollständig sortiertes Array zu erhalten. Ihre Vermutungen überprüfen sie durch die Ausführung des vorgegebenen Algorithmus.

Sie verändern daraufhin den Algorithmus so, dass er den vollständigen Bubblesort ausführt. Dies kann entweder durch verschachtelte for-Schleifen geschehen oder man implementiert eine neue Methode, die "wastutes" genügend oft aufruft.

XY-Diagramm einer Schallpegelmessung

Inhalt: Vertiefung des Umgangs mit Arrays, Anwendung der Array-Algorithmen, Typecast Double->Int

Um das bisher Gelernte zu vertiefen, sollen die Schülerinnen und Schüler eine weitere Aufgabenstellung bearbeiten. Ausgearbeitet liegt hier die Erstellung eines XY-Diagramms der Ergebnisse einer Schalldruckpegelmessung vor. Alternativ könnte auch die Implementierung der Schrittweitensteuerung bei der Funktionsanpassung (vergleichen Materialien von M: Funktionen im Kontext).

Bei der Erstellung des XY-Diagramms wird die Suche nach dem Maximum wiederverwendet, um eine Skalierung der Werte durchzuführen. Außerdem werden wie beim Bubblesort auffeinanderfolgende Werte benutzt, hier um ein Liniendiagramm zu erstellen. Die Schleifenbedingung der for-Schleife muss daran angepasst werden. Abschließend soll das Diagramm geglättet werden, indem für das Diagramm immer der Durchschnitt von n Messwerten benutzt wird. Die Schülerinnen und Schüler können untersuchen, wie sich die Wahl von n auswirkt. Programmiertechnisch ist dafür eine Schleife notwendig, in der der Zähler um n und nicht um 1 erhöht wird.

Die notwendigen Messwerte für das Diagramm können in einer der vorangehenden Stunden selbstständig aufgezeichnet werden. Die in den Materialien verwendeten Daten wurden mit der App "Phyphox"[11] erfasst. Diese erlaubt viele physikalische Experimente unter Verwendung der Sensoren des Smartphones. Hier wurde das Experiment "Audio Amplitude"

11Kostenlos für Android und iOS verfügbar, Homepage: https://phyphox.org/ (Stand 12.02.2019)

verwendet. Möchte man exakte Werte messen, ist eine Kalibrierung des Geräts mit einer Schallquelle mit bekannter Amplitude notwendig (Hinweis: für die Kalibrierung muss die Messung mit ► gestartet werden). Alternativ kann die App "Spaichinger Schallpegelmesser"[12] eingesetzt werden. Dort wird die Kalibrierung durch Zerreißen von Papier in einem definierten Abstand durchgeführt. Für die Darstellung als XY-Diagramm ist eine exakte Kalibrierung aber nicht unbedingt erforderlich, da der Verlauf der Lautstärke im Klassenzimmer qualitativ auch ohne exakte Werte interessant ist. Um die Größenordnung zu testen kann man in etwa von folgenden Schalldruckpegeln ausgehen:

• Atmen, Blätterrauschen, Schneefall	10 Dezibel
• sehr ruhiges Zimmer, Ticken einer Armbanduhr, leichter Wind	30 Dezibel
• Flüstern, leise Musik, ruhige Wohnstraße nachts	40 Dezibel
• Regen, Kühlschrank, leises Gespräch, Geräusche in der Wohnung	55 Dezibel
• normales Gespräch, Nähmaschine, Fernseher in Zimmerlautstärke	65 Dezibel
• Staubsauger, Wasserkocher, laufender Wasserhahn	70 Dezibel
• Kantinenlärm, Waschmaschine beim Schleudern, Großraumbüro	75 Dezibel
• laute Sprache, Streitgespräch, Klavierspiel	80 Dezibel
• Saxofonspiel, Hauptverkehrsstraße	85 Dezibel
• Kammerkonzert, Orchestergraben, Türknallen	90 Dezibel
• Musik (Kopfhörer), Holzfräsmaschine	95 Dezibel
• Schlagzeug/Rockkonzert, Motorsäge	110 Dezibel[13]

Im Klassenzimmer werden etwa zwischen 40dB (sehr leise) und 80dB (sehr laut) erreicht. Ab 85dB können die Ohren irreparablen Schaden nehmen.

Bei beiden Apps können die Daten als CSV-Datei gespeichert werden. Phyphox speichert nur den Zeitpunkt (in s) und den Schalldruckpegel. Der Spaichinger Schallpegelmesser speichert eine laufende Nummer, Schalldruckpegel und einige andere Werte. Achten Sie darauf, welches Trennzeichen für die Trennung der Spalten verwendet wird (Komma oder Semikolon). Processing kann nur CSV-Dateien mit Kommas verarbeiten. Bei der Table-Klasse in Greenfoot und im Java-Editor kann beim Aufruf des Konstruktors das Trennzeichen angegeben werden. Benennen Sie die Spalten um, so dass der Spaltenname aus einem einzigen Wort besteht. Bei Verwendung des Spaichinger Schallpegelmessers müssen Sie außerdem einige Kopfzeilen entfernen, so dass die CSV-Datei nur die Spaltenüberschriften und die Messwerte enthält.

Die Schüler beginnen zunächst, das Laden der Werte an die neue CSV-Datei anzupassen. Die Werte werden als Balkendiagramm angezeigt. Danach wird dies in ein XY-Diagramm umgewandelt. Da die Datei sehr viele Daten (ca. 12000 Datensätze) enthält, kommt es bei Greenfoot und beim Java-Editor zu längeren Ladezeiten. Daher sollen die Schüler zunächst nur die ersten 40 Werte laden und anzeigen. Zusätzlich ist bei Greenfoot notwendig, die Aktualisierung der Darstellung auf dem Bildschirm nicht nach jedem Zeichenbefehl durchzuführen, sondern erst am Ende der Zeichenbefehle. Das ist im vorgegebenen Programm schon realisiert (`setAutoRefresh(false)` und `refresh()`). In der Konsole wird außerdem ausgegeben, welche Zeile gerade geladen wird, damit deutlich wird, wie lange man noch warten muss.

12Kostenlos für Android und iOS verfügbar, Homepage: http://www.spaichinger-schallpegelmesser.de/ (Stand 12.02.2019)
13Quelle: "Wenn aus Geräuschen Lärm wird: Dezibel-Übersicht", Hörex, 2019, https://www.hoerex.de/service/presseservice/trends-fakten/wie-laut-ist-das-denn.html (Stand 17.02.2019)

Das XY-Punktdiagramm wird dann skaliert. Dazu müssen die Schülerinnen und Schüler die Suche nach dem Maximum an double-Zahlen anpassen. Mit dem ermittelten Maximum und der Bildschirmgröße wird ein Skalierungsfaktor bestimmt und angewendet. Nun füllt das Diagramm den Bildschirm möglichst gut aus.

Die Punkte werden im nächsten Schritt zu einem Liniendiagramm verbunden. Dabei muss die Schleifenbedingung angepasst werden, damit es nicht zu einer arrayindex-out-of-bounds-Exception kommt, wenn die Linie zum jeweils nächsten Punkt gezeichnet wird. Die Schleife darf nur noch bis zum vorletzten Punkt laufen.

Man sieht, dass die Werte im Diagramm sehr stark schwanken. Daher ist es sinnvoll, die Kurve zu glätten, um den Verlauf der Lautstärke im Klassenzimmer realistischer darzustellen. Dazu wird jeweils der Durchschnitt einiger aufeinander folgender Messwerte bestimmt. Dies sollen die Schüler zunächst von Hand für verschiedene Schrittweiten durchführen und dann in Quellcode umsetzen. Dabei kann das Programm zur Berechnung des Durchschnitts (siehe Zeitungsmeldung) verwendet und angepasst werden.

Literaturliste und Internetseiten:

Processing

Reas, C. & Fry, B.; Make: Getting Started with Processing; San Francisco; Maker Media; 2015[2]

Deussen O., Ningelgen, T.; Programmieren lernen mit Computergrafik – Eine Einführung mit Java und Processing; Wiesbaden; Springer-Verlag; 2018

Colubri, A.; Processing for Android; Cambridge, Massachussets; APress; 2017

Artikel bei Heise über Processing:

https://www.heise.de/make/artikel/Processing-Programmieren-fuer-Anfaenger-4106608.html (abgerufen am 24.04.2019)

Processing Homepage: *https://processing.org/* (abgerufen am 24.04.2019)

Vorlesung zu Processing: *http://michaelkipp.de/processing/* (abgerufen am 24.04.2019)

Processing für Android: *https://android.processing.org/index.html* (abgerufen am 24.04.2019)

Optische Täuschungen

Picon, D.; Optische Täuschungen (Sonderausgabe); Potsdam; Tandem Verlag GmbH

Akiyoshi‘s illusion pages: *http://www.ritsumei.ac.jp/~akitaoka/index-e.html* (abgerufen am 08.05.2019)