

Studienseminar für das Lehramt an Gymnasien Darmstadt

LiV: Muhammet Sözgen
Fächer: Informatik / Mathematik
Ausbildungssemester: Prüfungssemester

Entwurf für den Unterrichtsbesuch im Prüfungssemester in Informatik

Thema der Unterrichtsreihe: Kontrollstrukturen

Thema der Unterrichtsstunde: While-Schleifen und intelligentes Üben von if-Anweisungen sowie Struktogrammen

Schwerpunkt: Lernprozesse im Fach Informatik professionell nach schulformbezogenen Prinzipien, nach Prinzipien der inneren Differenzierung und individuellen Förderung und fachübergreifenden oder Fächer verbindenden Elementen gestalten

Schule: Lichtenbergschule
Ludwigshöhstr.105
64285 Darmstadt
06151 / 132563

Klasse: Einführungsphase

Uhrzeit:

Raum:

Ausbilder:

Beratende Ausbilderin:

Seminarleiterin:

Schulleiter:

Ort: Darmstadt

Datum: 14.03.2012

1. Lerngruppenanalyse

Ich unterrichte seit Anfang des Schuljahres eigenverantwortlich eine Lerngruppe der Einführungsphase. Im Unterrichtsraum sind Laptops am Rand angeordnet und im Zentrum des Raumes befindet sich die Möglichkeit für eine computerfreie Plenumsphase. Der Raum ist weiterhin mit einem Smartboard und einem Whiteboard ausgestattet. Die Lerngruppe besteht aus 18 Schülerinnen und Schülern¹, davon drei Schülerinnen und 15 Schüler. In der Lerngruppe befinden sich drei Schüler mit Migrationshintergrund. Diese Schüler zeigen keine sprachlichen Defizite und sind sehr gut in die Lerngruppe integriert, sodass deshalb keine besonderen Maßnahmen ergriffen werden müssen. Zum heutigen Unterricht werden zwei Schüler wegen schulischer Angelegenheiten fehlen.

Bezüglich der **Fach- und Sachkompetenzen** schätze ich die Lerngruppe als sehr heterogen ein. Ein Schüler zeigt sehr gut ausgeprägte Kompetenzen beim Implementieren von Problemlösungen in Java. Diese Kompetenzen hat sich der Schüler selbstständig angeeignet und zeigt sich auch im Unterricht fachlich sehr interessiert. Die theoretischen Komponenten, wie z.B. Syntaxdiagramme und Struktogramme, lernte dieser Schüler zwar erst im Unterricht kennen, jedoch zeigt er sich auch bei dem Umgang mit diesen Elementen sehr leistungsstark. Eine Gruppe von fünf SuS zeigt sich leistungsstark. Diese SuS lernten Java erst in meinem Unterricht kennen, doch sie sind sehr interessiert und arbeiten die Unterrichtsinhalte sorgfältig auf. Diese SuS sind in der Lage Struktogramme sowie Syntaxdiagramme zu lesen und selbstständig zu erstellen. Sie können ebenfalls Problemlösungen erarbeiten und diese in Java implementieren, wobei sie auch adäquat mit if-Anweisungen umgehen können. Vier weitere SuS zeigen sich fachlich interessiert, jedoch benötigen sie häufiger die Unterstützung durch mich oder die anderen SuS. Sie gelangen dabei ebenfalls zu zufriedenstellenden Ergebnissen bezüglich der Implementierung und der Erstellung der Struktogramme. Bei der Implementierung von if-Anweisungen zeigen diese SuS Defizite bei der Formulierung der Bedingungen. Zwei SuS haben krankheitsbedingt mehrere Doppelstunden gefehlt und haben die Fachinhalte bisher nicht aufgearbeitet. Diese SuS können mit dem Java-Editor die Grundstruktur von Programmen erzeugen und einfache Struktogramme interpretieren, jedoch scheitern sie bereits bei einfachen Variablenzuweisungen in Java. Die Erstellung von Struktogrammen ist für diese SuS ebenfalls sehr problematisch. Ein Schüler zeigt sich sehr leistungsschwach. Bei ihm scheitert es bereits an der Bedienung des JAVA-Editors, denn er ist nicht in der Lage ein einfaches „Hello World“ Konsolenprogramm zu implementieren. Unter Anleitung eines der anderen SuS oder durch mich gelingt ihm dies zwar, jedoch sind die dabei angelegten Kompetenzen in der nächsten Doppelstunde nicht mehr zu diagnostizieren. Der Schüler zeigt jedoch ausreichende

¹ Zur besseren Lesbarkeit verwende ich die Abkürzung SuS für „Schülerinnen und Schüler“.

Kompetenzen im Umgang mit Struktogrammen, indem er diese interpretiert und einfache Struktogramme selbst erstellt. Die übrigen SuS zeichnen sich durch solide Mitarbeit und einem durchschnittlichem Interesse am Fach aus.

Der Umgang mit Syntaxdiagrammen wurde bereits im ersten Halbjahr an HTML eingeübt und wird in diesem Halbjahr, im Sinne eines Spiralcurriculums, mit Java in einem neuen Kontext wieder aufgegriffen. Alle SuS sind in der Lage ein Syntaxdiagramm zu lesen und zwei Drittel der SuS sind in der Lage eines selbst zu erstellen. Seit drei Doppelstunden habe ich alle SuS dazu verpflichtet, vor der Implementierung immer ein Struktogramm anzufertigen. Für dieses Vorgehen plädiert Humbert ebenfalls.² Dies stärkte die Kompetenzen beim Lesen und Erstellen von Struktogrammen, jedoch konnte ich auch diagnostizieren, dass die logische Struktur in den Implementierungen häufig vom zuvor erstellten Struktogramm abweicht. Das habe ich bereits thematisiert und werde den Zusammenhang zwischen Struktogrammen und der Implementierung in Java im weiteren Unterricht mehr herausstellen, als ich dies bisher getan habe.

Insgesamt sind die **Sozialkompetenzen** der SuS sehr gut ausgeprägt. Die SuS kooperieren in den Arbeitsphasen miteinander und helfen sich gegenseitig bei Problemen. Es herrscht ein wertschätzendes Miteinander, wodurch sich die SuS in den Präsentationsphasen trauen, auch fehlerhafte Ergebnisse vorzustellen. Probleme zwischen Mädchen und Jungen sind nicht auszumachen.

Die **Methodenkompetenzen** der SuS sind ebenfalls gut ausgeprägt. Die SuS sind es gewohnt, in Einzelarbeit, Partnerarbeit und Gruppenarbeit zu arbeiten. In den regelmäßigen Plenumsphasen präsentieren sie ihre Lösungen und Lösungsansätze selbstbewusst vor der Lerngruppe, auch wenn die Arbeitsergebnisse nicht fehlerfrei sind (vgl. Sozialkompetenzen). Die Ergebnisse werden durch die Lerngruppe kritisch hinterfragt und durch konstruktives Feedback verbessert. Die selbstständige Aneignung von Fachinhalten mit Skripten wurde eingeübt und funktioniert zufriedenstellend. Des weiteren nutzen die SuS das virtuelle Klassenzimmer und die Zusatzangebote, wie z. B. meine Lösungsvorschläge und animierte Syntaxdiagramme mit dem Zusatzprogramm ANIMAL³. Die SuS arbeiten seit dem Beginn des Halbjahres mit dem JAVA-Editor, wodurch die Kompetenzen im Umgang mit dieser Software gestärkt wurden. Der HUS-Struktogrammer wurde ebenfalls häufig eingesetzt und alle SuS sind mit diesem Werkzeug vertraut, wenngleich es vor allem bei den leistungsschwächeren SuS immer wieder zu Bedienfehlern kommt.

Die **Selbstkompetenzen** der SuS sind unterschiedlich ausgeprägt. Die leistungsstärkeren SuS und die oben erwähnten sehr fleißigen SuS sind im hohen Maße bereit Verantwortung für ihr eigenes Lernen zu übernehmen. Dieses

² [10], Humbert, S. 158f

³ ANIMAL ist eine frei verfügbare Animationssoftware zur Visualisierung von Algorithmen.

verantwortungsbewusste Handeln der SuS wird im Umgang mit ihren Arbeitsaufträgen und Arbeitsergebnissen sowie ihren Fragen und Kommentaren zu den Lösungen der anderen SuS deutlich. Der Umgang mit den Hausaufgaben ist ebenfalls ein Indikator für die gut ausgebildeten Selbstkompetenzen dieser SuS. Die leistungsschwächeren SuS reichen nur selten Hausaufgaben ein.

2. Einordnung der Stunde in die Unterrichtseinheit

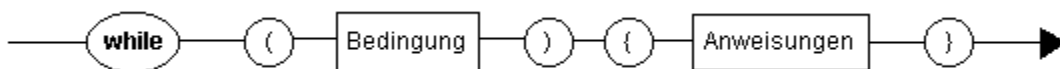
Stunden	Thema	Didaktisches Zentrum
		Im Zentrum der Stunde steht ...
1 / 2	If - Anweisungen	... die Erarbeitung von if-Anweisungen mit Hilfe von Syntaxdiagrammen.
3 / 4	Erarbeitung von komplexen if-Anweisungen	... die Modellierung von Problemlösungen mit Struktogrammen und deren Umsetzung in Java
5	Präsentation und Diskussion von freiwilligen Zusatzübungen durch SuS (SuS haben in Heimarbeit Problemlösungen zu gegebenen Problemen erarbeitet, Struktogramme erstellt und diese in Java umgesetzt)	
6	While-Schleifen	... die Erarbeitung von while-Schleifen sowie das intelligente Üben von if-Anweisungen unter dem Einsatz von Struktogrammen im Kontext eines Ticketautomaten.
7 / 8	Intelligentes Üben der while-Schleife (Kleingruppen von SuS erhalten individuelle Probleme)	... die arbeitsteilige Modellierung und Implementierung von Problemlösungen zu gegebenen Problemen.

3. Sachanalyse⁴

Schleifen bieten die Möglichkeit, Abschnitte vom Quelltext wiederholt auszuführen. Die while-Schleife ist eine kopfgesteuerte Schleife, in welcher die Anweisungen, welche mit geschweiften Klammern zu einem Schleifenblock zusammengefasst werden, so lange ausgeführt werden, bis die im Kopf angegebene Bedingung nicht mehr zutrifft.

Syntaxdiagramm:

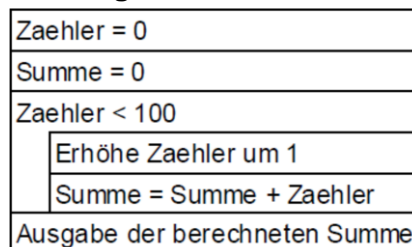
Dies ist eine einfache Version eines Syntaxdiagrammes für while-Schleifen. Alternativ hätte der Anweisungsblock auch mit einzelnen Anweisungen aufgebaut werden können.



Beispiel:

```
int Zaehler = 0;
int Summe = 0;
while (Zaehler < 100) {
    Zaehler = Zaehler + 1;
    Summe = Summe + Zaehler;
}
```

Struktogramm:



⁴ In Anlehnung an <http://de.wikipedia.org/wiki/While-Schleife> [Stand: 11.03.2012]; [15], Erlenkötter, S.127ff; [13], Sierra und Bates, S.11

4. Didaktische Analyse

Nach dem Hessischen Lehrplan lautet das Thema für die E2 „Grundlagen der Programmierung“.⁵ Dabei gehören Schleifen im Rahmen der Kontrollstrukturen zu den verbindlichen Unterrichtsinhalten.⁶ Das Verständnis von Kontrollstrukturen und insbesondere der Schleifen ermöglicht es den SuS Hintergrundwissen über die Wirkprinzipien und Möglichkeiten automatischer Informationsverarbeitung zu erwerben.⁷ Den SuS wird dadurch vor Augen geführt, dass der Computer bei der Durchführung von wiederholten Routinearbeiten dem Menschen deutlich überlegen ist. Hier besteht die Möglichkeit die Wechselwirkungen zwischen Informatiksystemen und ihrer gesellschaftlichen Einbettung zu thematisieren, was einer der Intentionen des Inhaltsbereiches Informatik, Mensch und Gesellschaft der Bildungsstandards Informatik genügt.⁸ Dieser Aspekt ist zwar wichtig, jedoch steht er in der heutigen Unterrichtsstunde nicht im Zentrum. Er wird in der Anekdote zum kleinen Gauß angerissen, worin der kleine Gauß eine langwierige Strafarbeit durch seinen menschlichen Intellekt schnell bewältigt. Dieser schnellen Lösung wird die Verarbeitung mit einer while-Schleife gegenübergestellt. Das Wissen um die Automatisierung geistiger Tätigkeiten kann als ein Aspekt zur Vorbereitung der SuS auf das zukünftige Leben angesehen werden, wodurch der allgemeinbildende Wert der Informatik herausgestellt wird.⁹ In dieser Stunde sollen die SuS das Programmieren als Mittel zum Problemlösen weiter festigen und durch die while-Schleife ein neues Werkzeug zum Modellieren und Strukturieren kennenlernen. Zuvor haben die SuS Variablen und if-Anweisungen kennengelernt, zur Modellierung von Anwendungssituationen genutzt sowie diese in GUI- und Konsolenprogrammen implementiert. Die Erweiterung der Modellierungsmittel um das Konzept der Schleifen ist für mich nun der nächste logische Schritt. Im Lehrplan wird kein bestimmter Schleifentyp vorgeschrieben.¹⁰ Ich habe mich für die Behandlung der while-Schleife entschieden, da diese m.E. nach am einfachsten zu verstehen ist. Die Syntax dieser kopfgesteuerten Schleife ist besonders einfach, was auch an dem übersichtlichen Syntaxdiagramm (vgl. Sachanalyse) erkennbar wird.¹¹ Die for-Schleife hingegen hat einen vergleichsweise komplexen Aufbau. Hier müssen bereits in der ersten Zeile viele Dinge, wie z.B. die Variablendeklaration und die Veränderung dieser Variable für einen Schleifenabbruch berücksichtigt werden, die einen Anfänger schnell überfordern können. Zur Visualisierung der entwickelten Algorithmen dienen in dieser Stunde wieder Struktogramme (vgl. Lerngruppenanalyse), welche nach dem Lehrplan zu den

⁵ [5], Hessischer Lehrplan, S. 14

⁶ [5], Hessischer Lehrplan, S. 15

⁷ Vgl. [5], Hessischer Lehrplan, S. 14

⁸ Vgl. [6], AKBSI, S. 42f

⁹ Vgl. [2], Hubwieser, S. 62f

¹⁰ [5], Hessischer Lehrplan, S. 15

¹¹ Vgl. [13], Sierra und Bates, S.11

verbindlichen Unterrichtsinhalten gehören.¹² In den EPA-Informatik gehören Struktogramme ebenfalls zu den verbindlichen Fachinhalten.¹³ Durch den verstärkten Einsatz von Struktogrammen sollen die Denkweisen beim Entwickeln von Programmen in den Vordergrund gerückt werden, wodurch wir uns weiter von der Hackermentalität beim Programmieren entfernen.¹⁴

Bei den ausgewählten Aufgaben¹⁵ steht das Thema „Ticketautomat“ im Mittelpunkt, wodurch ich der Aufforderung nach einem themenorientierten Zugang nachkomme.¹⁶ Die SuS erhalten hierbei einen Zugang zum Inhaltsbereich Sprachen und Automaten¹⁷, indem sie die gegebenen Automaten analysieren und anschließend eine Modellierung mit einem Struktogramm visualisieren. Diese Veranschaulichung informatischer Sachverhalte genügt eine der Intentionen des Prozessbereiches Darstellen und Interpretieren.¹⁸ Die Modellierung setzen sie dann in Java um, was wiederum dem Prozessbereich Modellieren und Implementieren genügt.¹⁹ Ein Schwerpunkt liegt auch auf dem Inhaltsbereich Algorithmen.²⁰ Die SuS sollen Algorithmen entwerfen, implementieren und anschließend im Plenum diskutieren. In der Aufgabe eins und bei der Nutzung der Hilfestellung zur Aufgabe zwei modifizieren und ergänzen die SuS die angegebenen Quelltexte. Zur Binnendifferenzierung nutze ich des Weiteren die Animation zu while-Schleifen, welche auch ein Syntaxdiagramm beinhaltet, und vorgefertigte Beispielprogramme, welche vor allem die leistungsschwächeren SuS unterstützen sollen. Für die leistungsstärkeren SuS stehen Zusatzaufgaben zur Verfügung. Bereits während meiner Einstiegspräsentation werden die SuS dazu angehalten, ein einfaches Programm zu lesen und zu verstehen. Damit wird eine Kompetenz aus dem Inhaltsbereich Algorithmen gefördert.²¹ Diese Gelegenheit nutze ich, um ein dazugehöriges Struktogramm mit dem gegebenen Quelltext in Verbindung zu setzen, was den SuS schwerfällt (vgl. Lerngruppenanalyse). Hartmann spricht sich ebenfalls dafür aus, dass die SuS zuerst gegebenen Quelltext lesen sollen, bevor sie selbst ähnliche Problemlösungen umsetzen und führt als ein gewichtiges Argument für dieses Vorgehen an, dass das Lesen und Analysieren von Quelltext einen wichtigen Teil der realen Arbeit von Informatikern darstellt.²² Nach der Erstellung des Erwartungshorizontes zu den Aufgaben sind mir Rundungsfehler zur Laufzeit aufgefallen. Diese lassen sich zwar relativ leicht beheben, doch die dazu nötigen Quellcodezeilen und das Verständnis der dahinter liegenden Problematik lässt sich m.E. nach nicht adäquat in dieser Stunde aufarbeiten. Als didaktische Reduktion werde

¹² [5], Hessischer Lehrplan, S. 15

¹³ [9], EPA-Informatik, S. 6

¹⁴ [5], Hessischer Lehrplan, S. 14

¹⁵ Die Aufgaben sind an das Skript von Herrn Röhner angelehnt, [11]

¹⁶ [5], Hessischer Lehrplan, S. 15

¹⁷ Vgl. [6], AKBSI, S. 13 und S. 57

¹⁸ Vgl. [6], AKBSI, S. 14

¹⁹ Vgl. [6], AKBSI, S. 13

²⁰ Vgl. [6], AKBSI, S. 15f

²¹ Vgl. [6], AKBSI, S. 32f

²² Vgl. [8], Hartmann, S. 131-138

ich auftretende Rundungsfehler als solche benennen und für die Behebung dieser Probleme auf die kommenden Unterrichtsstunden verweisen. Das Thema „Ticketautomat“ leistet ebenfalls eine propädeutische Funktion für die zustandsorientierte Modellierung und dem Verständnis von endlichen Automaten.²³ Diese Modellierung gehört auch zu den verbindlichen Fachinhalten der EPA-Informatik.²⁴ Zur Vermeidung einer hohen Elementinteraktivität habe ich an dieser Stelle auf die Darstellung mittels eines Zustandsdiagramms verzichtet, wobei ich mich an die Cognitive Load Theorie stütze.²⁵ Ein Zustandsdiagramm für die erste Aufgabe werde ich in einer der Folgestunden nachreichen und die SuS selbstständig eines für die zweite Aufgabe entwickeln lassen. Ich habe mit dem Thema „Ticketautomat“ bewusst eines gewählt, welches stark in der Informatik verortet ist. Ich hätte zum Beispiel auch die Simulation von Populationen als Thema wählen können, um ein fächerübergreifendes und fächerverbindendes Lernen mit dem Fach Biologie oder mit dem Fach PoWi zu gewährleisten. Hierdurch würden Themen der anderen Fächer in den Informatikunterricht einbezogen und nach einer Modellierung in Java implementiert werden. Für die Einstiegsstunde in Schleifen sehe ich jedoch die Gefahr, dass die SuS die Fachinhalte der anderen Fächer möglicherweise nicht parat haben. Dann müsste ein erheblicher Teil der Unterrichtszeit zur Auffrischung dieser Fachinhalte aufgebracht werden, wodurch das eigentliche Unterrichtsziel dann vielleicht gar nicht erreicht werden würde.²⁶ Den fächerverbindenden Ansatz werde ich in den Folgestunden bei den individuellen Problemen für die einzelnen Gruppen stärker berücksichtigen (vgl. Einordnung der Stunde in die Unterrichtsreihe).

Als Hausaufgabe erhalten die SuS eine semi-enaktive²⁷ Webseite mit einem lückenhaften Quelltext, welcher eine while-Schleife beinhaltet.²⁸ Hier sollen die SuS Quelltextfragmente in die Lücke ziehen und die Ausgabe dieses Programmes mit Papier und Stift, also ohne sie zu implementieren, herausfinden. Dies regt die SuS dazu an über Variablenbelegungen, Abbruchbedingungen und die Anzahl der Wiederholungen nachzudenken. Die Erkenntnisse und Probleme aus dieser Aufgabe nutze ich in der anschließenden Unterrichtsstunde als Einstieg, bei dem ich die SuS über diese Themen diskutieren lasse.

5. Didaktisches Zentrum der Stunde

Im Zentrum der Stunde steht die Erarbeitung von while-Schleifen sowie das intelligente Üben von if-Anweisungen unter dem Einsatz von Struktogrammen im Kontext eines Ticketautomaten.

²³ [5], Hessischer Lehrplan, S. 16

²⁴ [9], EPA-Informatik, S. 6

²⁵ Vgl. [17], Wikipedia

²⁶ Vgl. [3], Schubert und Schwill, S. 50f

²⁷ Vgl. [8], Hartmann, S. 117

²⁸ Die Aufgabe ist angelehnt an [13], Sierra und Bates, S. 23

6. Kompetenzen und Indikatoren

In dieser Stunde werden vor allem die Kompetenzbereiche **Modellieren und Implementieren, Darstellen und Interpretieren** sowie durch die Wahl der Sozialformen das **Kommunizieren und Kooperieren** gestärkt.

Die SuS können nach dieser Stunde ...

- Fachkompetenzen:
 - ... eine einfache while-Schleife implementieren.
 - ... while-Schleifen in Modellierungen nutzen.
 - ... eine Problemlösung mit Hilfe von Struktogrammen visualisieren.
 - ... if-Anweisungen adäquat in Modellierungen nutzen.
- Sozial-, Selbst- und Methodenkompetenzen:
 - ... ihr Ergebnisse präsentieren und gemeinsam diskutieren.

7. Methodische Analyse der Stunde

Bei der Gestaltung des heutigen Unterrichts habe ich mich am Sandwichprinzip nach Wahl orientiert, in dem sich Phasen der Vermittlung von Inhalten und Phasen zur individuellen Auseinandersetzung abwechseln.²⁹ Als **Einstieg** in das Sandwich nehme ich einen informierenden Unterrichtseinstieg. Hier komme ich auch den SuS entgegen, da sie mir als Feedback zum Unterricht zurückgemeldet haben, dass sie sich besonders in den ersten Phasen, bei der Erarbeitung neuer Themen, eine längere lehrerzentrierte Phase wünschen. Nach einer kurzen Motivationsphase, die für Transparenz sorgen soll, zeige ich als Eyecatcher³⁰ eine Strafarbeit aus der bei den SuS beliebten Zeichentrickserie die Simpsons (siehe Folien). Im Kontext dieses Beispiels zeige ich eine while-Schleife, ohne sie zu erläutern und gehe sofort auf die Auswirkungen dieser Schleife ein. Der Eyecatcher ermöglicht den SuS einen Zugang zum Thema auf der symbolischen und auf der ikonischen Repräsentationsebene.³¹ Die symbolische Ebene wird durch den Quelltext erreicht. Die ikonische Ebene wird durch die Visualisierung der Auswirkung der dargestellten while-Schleife erreicht. Am Beispiel des kleinen Gauß werde ich die Syntax von while-Schleifen thematisieren und diese mit einem Struktogramm in Beziehung setzen (vgl. Did. Analyse). Alternativ hätte ich auch nach einem sehr kurzen informativen Einstieg direkt auf das Skript verweisen können, was jedoch den Bedürfnissen der SuS nicht gerecht werden würde.

Für die Überleitung in die **Erarbeitungsphase** stelle ich den Kontext „Ticketautomat“ vor und gebe zur Schaffung von Transparenz einen Überblick über die Aufgaben. Die SuS arbeiten anschließend selbstständig mit dem Skript, wobei ihnen die Präsentationsfolien zur Orientierung an Beispielen (vgl. Did. Analyse) ebenfalls zur Verfügung stehen. Leistungsschwächere SuS können sich an dieser Stelle auch die Animation sowie das Konsolenprogramm zum kleinen Gauß aus dem virtuellen

²⁹ Vgl. [1], Wahl, Kapitel 5.1

³⁰ Vgl. [1], Wahl, S. 115

³¹ Vgl. [8], Hartmann, S. 116

Klassenzimmer herunterladen und analysieren. Leistungsstärkere SuS können bereits mit den weiterführenden Aufgaben beginnen, wenn sie die Aufgabe eins bearbeitet haben. Den SuS ist es freigestellt, ob sie in Einzel- oder Partnerarbeit arbeiten. Die Partner dürfen sich die SuS ebenfalls selbstständig wählen. Diese offene Gestaltung der Sozialform zeigte sich bereits in den vergangenen Stunden sehr produktiv und die SuS stärkten weiter ihre Kompetenzen im selbstorganisierten Lernen. Durch die guten Sozialkompetenzen helfen sich die SuS in diesen Arbeitsphasen gegenseitig (vgl. Lerngruppenanalyse), was von mir explizit erwünscht und an gegebenen Stellen auch eingefordert wird. Meine Rolle hierbei ist weitgehend zurückhaltend und ich stehe den SuS beratend zur Seite, falls dies nötig sein sollte. Dadurch erhalten die SuS genügend Zeit zur individuellen Auseinandersetzung mit den Unterrichtsinhalten. Bei der anschließenden **Ergebnissicherung** kommen die SuS im Zentrum des Raumes zusammen und ich wähle SuS zur Präsentation der Arbeitsergebnisse zur Aufgabe eins aus. Diese Gelenkstelle schafft eine bewusste Trennung der Übungs- und Sicherungsphase, indem sich die SuS von den Computern wegsetzen.³² Die SuS sollen in dieser Phase ihr Struktogramm sowie ihre Implementierung in Java vorstellen und in Beziehung setzen. Ich werde mich dabei zurückhalten und lediglich Impulse geben, falls mir die Diskussion nicht fruchtbar erscheint, oder ins Stocken gerät. Falls dies nicht bereits erfolgte, werde ich am Ende dieser Phase Feedback zur Kommentierung des Quellcodes einfordern. Hier setze ich das Minimalziel für diese Stunde. Falls noch Zeit sein sollte, werde ich zur Überleitung in die **Vertiefungsphase** die SuS an den Inhalt der zweiten Aufgabe erinnern und sie zur Weiterarbeit auffordern. Die Sozialform und meine Rolle sind hierbei wie in der Erarbeitungsphase. Für die leistungsschwächeren SuS steht zur Binnendifferenzierung eine Hilfe-Datei zur Bearbeitung der Aufgabe bereit, die ihnen eine Orientierung geben soll. Für die leistungsstärkeren SuS stehen Zusatzaufgaben zur Verfügung. Am Ende der Vertiefungsphase gibt es nun zwei mögliche **Unterrichtsausstiege**. Falls genügend Zeit sein sollte, kann wieder eine **Ergebnissicherung** nach dem beschriebenen Vorgehen erfolgen. Hier setze ich mein Maximalziel für diese Stunde. Falls die Zeit dafür nicht ausreicht, werden die SuS im Plenum einen **Überblick über ihren bisherigen Arbeitsstand** geben und sich über die Aufgabe kurz austauschen. Dies schafft Transparenz bezüglich der Lernfortschritte und der eventuell aufgetretenen Probleme. Beide Varianten ermöglichen einen gemeinsamen Stundenabschluss, welcher auch eine Reflexion der Stunde ermöglicht. Zum Abschluss der Stunde werde ich die Hausaufgabe (vgl. Did. Analyse) zur weiteren Festigung von while-Schleifen stellen.

³² Vgl. [8], Hartmann, S.142

8. Verlaufsplan

Hauptphasen / Funktion	Lehreraktivität / Impulse	Schüleraktivität / mögliche Ergebnisse / Kompetenzerwerb	Lehr- /Lernform, Medien	Anmerkungen / Varianten
Einstieg	L. gibt einen informierenden Einstieg für while-Schleifen mit zwei Beispielen.	SuS hören zu und stellen ggf. Nachfragen.	Plenum PowerPoint Beamer	
Erarbeitung	L. präsentiert den heutigen Kontext und gibt im virtuellen Klassenzimmer die Unterrichtsmaterialien frei.	SuS bearbeiten die erste Aufgabe. Leistungsschwächere SuS beschäftigen sich zunächst mit der ANIMAL-Animation.	Einzelarbeit oder Partnerarbeit Skript	Leistungsstarke SuS können bereits mit der zweiten Aufgabe beginnen, wenn sie fertig sind.
Ergebnissicherung	L. fordert zur Präsentation auf und hält sich zurück.	SuS präsentieren ihr Programm und setzen es in Beziehung zum zuvor erstellten Struktogramm. Die Ergebnisse werden diskutiert.	Plenum SuS – Präsentation Beamer	Möglicher Impuls: „Setzt das Struktogramm in Beziehung zu eurem Programm.“
MINIMALZIEL				
Vertiefung	L. fordert zur Weiterarbeit an den Aufgaben auf.	SuS bearbeiten die zweite Aufgabe. Leistungsstärkere SuS bearbeiten die nächsten Aufgaben, wenn sie fertig sind.	Einzelarbeit oder Partnerarbeit Skript	
Ergebnissicherung	L. fordert zur Präsentation auf und hält sich zurück.	SuS präsentieren ihr Programm und setzen es in Beziehung zum zuvor erstellten Struktogramm. Die Ergebnisse werden diskutiert.	Plenum SuS – Präsentation Beamer	Möglicher Impuls: „Setzt das Struktogramm in Beziehung zu eurem Programm.“

9. Literaturverzeichnis

- [1] Diethelm Wahl
Lernumgebungen erfolgreich gestalten
Vom trägen Wissen zum kompetenten Handeln
2. Auflage mit Methodensammlung
Klinkhard, 2006
- [2] Didaktik der Informatik
Peter Hubwieser
Springer-Verlag Berlin Heidelberg 2000, 2001, 2004 und 2007
- [3] Didaktik der Informatik
Sigrid Schubert, Andreas Schwill
Spektrum Akademischer Verlag, Heidelberg
2. Auflage 2011
- [4] Schulcurriculum für den Wahlunterricht Informatik in der Sekundarstufe 1
Fachkonferenz Informatik (Lichtenbergschule)
- [5] Lehrpläne Sekundarstufe 1 (G8) und gymnasiale Oberstufe 2010
Hessisches Kultusministerium
Wiesbaden 2010
- [6] AKBSI
Grundsätze und Standards für die Informatik in der Schule
Bildungsstandards Informatik für die Sekundarstufe I
Gesellschaft für Informatik e. V. (GI), Bonn, 2008
- [7] Planung von Unterrichtseinheiten und –stunden im Informatikunterricht
Rüdiger Baumann
LOG IN 13, Heft 6, 1993
- [8] Informatikunterricht planen und durchführen
Werner Hartmann (et al.)
Springer Verlag Berlin Heidelberg 2006
1. Korrigierter Nachdruck, 2007
- [9] EPA – Informatik
http://www.kmk.org/fileadmin/veroeffentlichungen_beschluesse/1989/1989_12_01-EPA-Informatik.pdf [Stand: 11.03.2012]
- [10] Didaktik der Informatik
mit praxiserprobtem Unterrichtsmaterial
Ludger Humbert
B.G. Teubner Verlag / GWV Fachverlage GmbH, Wiesbaden 2006
2. Auflage
- [11] Unterrichtsskript von Gerhard Röhner, 2007/08
- [12] Informatik 1
Objekte Strukturen Algorithmen
Hubwieser (et al.)
Ernst Klett Verlag GmbH, Stuttgart, 2004
- [13] Java von Kopf bis Fuß
Sierra und Bates
O'Reilly Verlag GmbH & Co. KG, Köln, 2006
3. Korrigierter Nachdruck, 2008

- [14] Einführung in die Informatik
Objektorientiert mit Java
Küchlin und Weber
Springer – Verlag Berlin Heidelberg 2003
2. Überarbeitete und erweiterte Auflage

- [15] Java-Applikationen
Anwendungsprogrammierung mit JFC
Erlenkötter
Rowohlt Taschenbuch Verlag GmbH, Hamburg, 1998
2. Auflage, 2001

- [16] Fundamente der Informatik
Hubwieser und Aiglstorfer
Ablaufmodellierung, Algorithmen und Datenstrukturen
Oldenbourg Wissenschaftsverlag GmbH, München, 2004

- [17] Wikipedia-Artikel zur Cognitive Load Theorie
http://de.wikipedia.org/wiki/Cognitive_Load_Theory [Stand: 12.03.2012]

10. Anlagen

- Skript zu while-Schleifen
- ANIMAL-Animation zu einem Syntaxdiagramm für eine while-Schleife
- Konsolenprogramm zum kleinen Gauß
- Hilfe zur Aufgabe 2 (Abgestufte Hilfe, in Form einer JAVA-Datei)
- Drag and Drop Webseite zu while-Schleifen

WHILE - SCHLEIFE

Mit unseren bisherigen Mitteln haben wir eine einmalige Ausführung der Anweisungen erreicht. Durch die if-Anweisung bekamen wir die Möglichkeit Anweisungen nur unter bestimmten Bedingungen auszuführen und ggf. zu überspringen. Mit der while-Schleife haben wir nun auch die Möglichkeit Anweisungen unter bestimmten Bedingungen zu wiederholen, bis die Bedingung nicht mehr erfüllt wird. Hierbei zeigt sich die wahre Stärke des Computers. Er ist in der Lage Anweisungen in kurzer Zeit sehr häufig zu wiederholen und ist in diesem Punkt dem Menschen überlegen.

Ein Beispiel:

Der kleine Gauß (Johann Carl Friedrich Gauß 1777 – 1855) saß im Mathematikunterricht und störte durch sein Geschwätz den Unterricht. Als Strafarbeit sollte er die Zahlen von 1 bis 100 aufaddieren. Diese Arbeit wurde als Strafe angesehen, da die Rechnungen stupide waren und viel Zeit in Anspruch nahmen.

$$1 + 2 + 3 + 4 + 5 + \dots + 98 + 99 + 100 = 5050$$

Solche stupiden Arbeiten können wir an den Computer abgeben. Eine Umsetzung in Java sieht folgendermaßen aus:

Java - Quellcode	Struktogramm
<pre>// Deklaration der Variablen int Zaehler = 0; int Summe = 0; // Verarbeitung // Berechnung der Summe aller Zahlen von 1 bis 100. // Solange die Bedingung „Zaehler < 100“ stimmt, // werden die Anweisungen wiederholt. while (Zaehler < 100) { Zaehler = Zaehler + 1; Summe = Summe + Zaehler; } //Ausgabe System.out.println(„Gesuchte Summe: “ + Summe);</pre>	<pre> graph TD Start([Start]) --> Init1[Zaehler = 0] Init1 --> Init2[Summe = 0] Init2 --> LoopCond[Zaehler < 100] LoopCond --> LoopBody subgraph LoopBody [] direction TB B1[Erhöhe Zaehler um 1] B2[Summe = Summe + Zaehler] end LoopBody --> LoopCond LoopCond --> End([Ausgabe der berechneten Summe]) End --> Stop([Stop]) </pre>

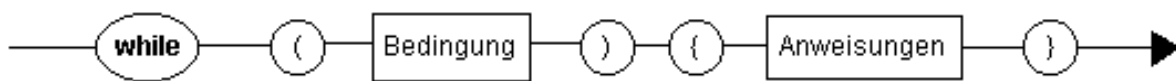
Der kleine Gauß zeigte an dieser Stelle die Überlegenheit des Menschen durch den menschlichen Verstand, indem er eine intelligente Methode zur Bestimmung dieser Summe fand. Er rechnete

$$50 * (101) = 5050$$

und gelangte so sehr schnell zum Ergebnis.

Hinweis: Das Konsolenprogramm zur aufwendigeren Lösung dieser Strafarbeit könnt ihr aus dem Virtuellen Klassenzimmer herunterladen.

Syntaxdiagramm zur while-schleife

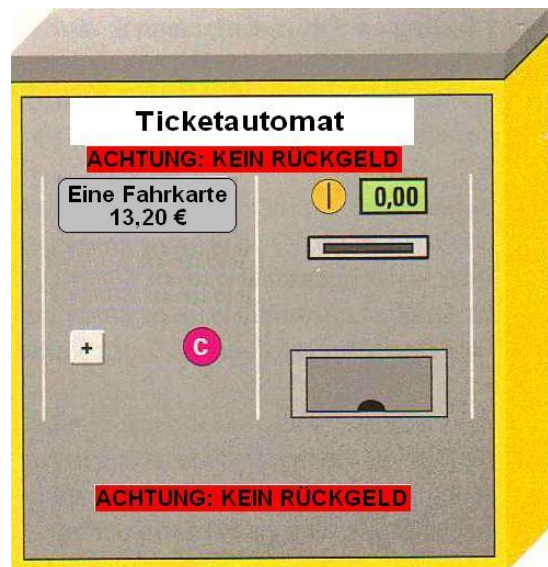


Hinweis: Im Virtuellen Klassenzimmer findet ihr eine ANIMAL-Animation zur while-Schleife.

Aufgabe 1

Vor einem Vergnügungspark steht der abgebildete Ticketautomat. Ein Ticket kostet 13,20 €. Nach der Betätigung des Knopfes muss solange Geld eingeworfen werden, bis der Automat das Ticket ausgibt. Der Automat akzeptiert 2 €, 1 €, 50 Cent, 20 Cent und 10 Cent Münzen.

Die Geldrückgabe ist defekt, sodass der Automat das überschüssige Geld einbehält.



- Lade dir das Konsolenprogramm „Ticketautomat“ aus dem Virtuellen Klassenzimmer herunter und probiere verschiedene Eingaben aus.
- Erstelle ein Struktogramm zu dem gegebenen Programm.
- In der aktuellen Version akzeptiert das Programm auch unmögliche Eingaben, wie z.B. 1,75 €. Verändere das Struktogramm aus b) so, dass nur die in der Aufgabenstellung angegebenen Münzen akzeptiert werden.
- Ändere das Programm gemäß deinem Struktogramm aus c)
- Bonus: Ändere das Programm so ab, dass am Anfang die Anzahl der zu kaufenden Tickets abgefragt wird.

Aufgabe 2

Heureka, der Ticketautomat gibt endlich wieder Geld zurück. Erstellt in Partnerarbeit ein Konsolenprogramm, welches das Wechselgeld in Münzen zurückgibt. Dabei sollen möglichst wenige Münzen benutzt werden.

Beispiel:

Bei einem Fahrpreis von 13,20 € und einer Eingabe von 20 € gibt er als Wechselgeld drei 2 € Münzen, eine 50 Cent, eine 20 Cent und eine 10 Cent-Münze zurück.



- Erstellt ein Struktogramm zu dieser Aufgabe.
- Implementiert ein Konsolenprogramm gemäß eures Struktogramms aus a).

TIPP: Falls ihr nicht vorankommt, nutzt das unvollständige Konsolenprogramm „Hilfe“ aus dem Virtuellen Klassenzimmer

Aufgabe 3 - BONUS

Implementiere ein Konsolenprogramm, das den kompletten, funktionsfähigen Automaten simuliert. Dabei kannst du alle Erkenntnisse aus den Aufgaben 1 und 2 nutzen.

Aufgabe 4

Bearbeite die im Virtuellen Klassenzimmer bereitgestellte Übung zu while-Schleifen. Die Aufgabe sollst du **nicht** implementieren, sondern mit Papier und Bleistift lösen. In dieser Übung kannst du die Quellcodefragmente in die Lücke ziehen, um einen besseren Überblick zu gewinnen.