



**IMST – Innovationen machen Schulen Top**

Informatik kreativ unterrichten

**KONZEPTE DER THEORETISCHEN  
INFORMATIK AN DER AHS:  
ENDLICHE AUTOMATEN MIT HILFE VON  
„BRAINFUCK“ VERSTEHEN LERNEN**

Projekt-ID: 1454

Mag. Stefan Otti

BG/BRG Villach-Peraustraße

Villach, April 2015

# INHALTSVERZEICHNIS

<b>ABSTRACT</b> .....	<b>3</b>
<b>1 EINLEITUNG</b> .....	<b>4</b>
1.1 Motivation und Rahmenbedingungen des Projekts.....	4
1.2 Ziele.....	4
1.2.1 Ziele auf SchülerInnenebene.....	4
1.2.2 Ziele auf LehrerInnenebene.....	4
1.3 Zeitplan.....	5
<b>2 HINTERGRUNDWISSEN ZU ENDLICHEN AUTOMATEN UND TURINGMASCHINEN</b> .....	<b>5</b>
<b>3 PROJEKTABLAUF</b> .....	<b>7</b>
<b>4 EVALUATION</b> .....	<b>8</b>
4.1 Evaluation projektspezifischer Ziele.....	10
4.1.1 Inhaltliche Aspekte.....	10
4.1.2 Prozessaspekte.....	12
4.2 Evaluation aus Sicht übergeordneter IMST Ziele.....	13
4.2.1 Gender- und Diversitätsaspekte.....	13
4.2.2 Schulentwicklungs- und Disseminationsaspekte.....	13
<b>5 ZUSAMMENFASSUNG</b> .....	<b>14</b>
<b>6 QUELLEN</b> .....	<b>14</b>
<b>ANHANG 1: STARTERHEBUNG</b> .....	<b>15</b>
<b>ANHANG 2: ABSCHLUSSTEST</b> .....	<b>16</b>

## ABSTRACT

Mit diesem Projekt soll gezeigt werden, dass man Elemente der theoretischen Informatik so in den Informatikunterricht einfließen lassen kann, dass die Jugendlichen erkennen, dass diese Theorie auch jenseits des Einsatzes von Computern relevant ist. Es sollen den Jugendlichen abstrakte Konzepte der Informatik nachvollziehbar veranschaulicht werden, insbesondere endliche Automaten und Turingmaschinen.

Als Einstiegswerkzeug für das Verständnis von Turingmaschinen wird die Lernumgebung „TuringKara“, eine an Turingmaschinen angepasste Version des Klassikers „Kara“ verwendet. Mit TuringKara lassen sich auf anschauliche Weise die typischen Aufgaben rund um Turing-Maschinen wie Grundrechenarten oder Mustererkennung illustrieren.

Anschließend werden komplexere Programme mit der esoterischen Programmiersprache Brainfuck (BF) implementiert. BF ist zwar für die Erstellung umfangreicher Software zu ineffizient, durch die Turing-Vollständigkeit aber gut geeignet, um wichtige Grundlagen der Programmierung zu erlernen. Besonders praktisch für den Einsatz im Unterricht ist die Tatsache, dass BF-Programme stark der formalen Definition einer Turingmaschine ähneln.

Schulstufe: 11
Fächer: Wahlpflichtfach Informatik
Kontaktperson: Mag. Stefan Otti
Kontaktadresse: BG/BRG Villach-Peraustraße Peraustraße 10 9500 Villach



Abbildung 1: Schülergruppe des WPF Informatik 2014/15 am BG/BRG Villach-Peraustraße

*Kategorien:* 11. Schulstufe, Alternative\_Unterrichtsmethode, Gruppenarbeit, Informatik, Informatikunterricht, Theoretische\_Informatik, Programmierung, Endliche\_Automaten, Modelle, Abstraktion

# 1 EINLEITUNG

## 1.1 Motivation und Rahmenbedingungen des Projekts

Der Autor möchte mit diesem Projekt eine Möglichkeit zeigen, wie Inhalte der theoretischen Informatik in den Unterricht der Sekundarstufe II so eingebaut werden können, dass den Jugendlichen abstrakte Konzepte der Informatik verständlich und nachvollziehbar veranschaulicht werden.

Entgegen der Annahmen im Projektantrag, der zu einem Zeitpunkt, an dem die Lehrfächerverteilung noch nicht endgültig war, formuliert wurde, wurde das Projekt mit einer Gruppe des Wahlpflichtfachs Informatik mit 6 Schülern durchgeführt, sodass sich die Rahmenbedingungen wie folgt darstellen:

Klassen: 7ABC

Schulstufe: 11.

Anzahl der Schüler: 6 männlich, 0 weiblich

## 1.2 Ziele

### 1.2.1 Ziele auf SchülerInnenebene

- Konstruktion einfacher endlicher Automaten und Turingmaschinen
- Grundlegendes Verständnis für Konzepte der theoretischen Informatik
- Kennenlernen von verschiedenen Notationen endlicher Automaten und Turingmaschinen
- Beschäftigung mit komplexen Aufgabenstellungen insbesondere bei der Implementierung der Turingmaschinen in Brainfuck
- Förderung der Teamarbeit beim Lösen von Gruppenaufgaben
- Wecken von Interesse / Erkennen des eigenen Talents für informatische Sachverhalte und Aufgabenstellungen

### 1.2.2 Ziele auf LehrerInnenebene

- Gestaltung eines Themenschwerpunktes „Endliche Automaten und Turingmaschinen im Informatikunterricht“
- Richtiger Einsatz didaktischer Software
- Rollenwechsel zum „Coach“, der den SchülerInnen beim Lösen der Aufgaben zur Seite steht
- Auffrischen/Erweitern des eigenen Methodenpools
- Rückmeldung über und ggf. Anpassung der Unterrichtsmethoden und des –stoffes

## 1.3 Zeitplan

Zeitraum	Maßnahme
Sommer 2014	Konzeption der Unterrichtseinheiten Erstellung der Unterrichtsmaterialien
Oktober 2014	Starterhebung über die Vorkenntnisse und Interessen der Schüler
Bis Feber 2015	Abhalten der Unterrichtseinheiten
Bis März 2015	Verfassen des Zwischenberichts
Bis Juli 2015	Verfassen des Endberichts

## 2 HINTERGRUNDWISSEN ZU ENDLICHEN AUTOMATEN UND TURINGMASCHINEN

Nicht nur für den Autor, sondern auch für LehrerInnen, die erwägen, das vorgestellte Unterrichtsprojekt oder Teile daraus zu übernehmen, ist die Beschäftigung mit einigen Grundlagen der Informatik unbedingte Voraussetzung. Dies geschieht nicht zuletzt, um tiefer gehende Fragen von SchülerInnen fundiert beantworten zu können.

Daher ist das Kapitel 2 als Lektüre für interessierte LehrerInnen gedacht, um vielleicht schon etwas in Vergessenheit geratenes Wissen zu den behandelten Themen aufzufrischen und um weiterführende Fragen von SchülerInnen angemessen beantworten zu können. In der vorliegenden Aufbereitung sind die Inhalte keinesfalls für die direkte Vermittlung an SchülerInnen geeignet.

In diesem Kapitel möchte der Autor in der gebotenen Kürze die wichtigsten Informationen zu Turingmaschinen darlegen, als Referenz sowie für weiterführende Lektüre wird auf Herold, Lurz & Wohlrab (2007, S. 620ff) verwiesen.

Eine deterministische Turingmaschine ist formal als 7-Tupel  $M = (Z, \Sigma, \Gamma, \delta, z_0, \#, E)$  definiert, wobei die Symbole:  $Z, \Sigma, \Gamma, \delta, z_0, \#, E$  die folgenden Bedeutungen haben:

$Z$ : endliche Menge von Zuständen

$\Sigma \subset \Gamma$ : endliches Eingabealphabet

$\Gamma$ : endliches Bandalphabet

$\delta: Z \times \Gamma \rightarrow Z \times \Gamma \times \{L, 0, R\}$ : Überföhrungsfunktion

$z_0 \in Z$ : Startzustand

$\#$ : leeres Feld

$E \subseteq Z$ : Menge von Endzuständen

Konzeptionell besteht das Grundmodell einer Turingmaschine aus einem unbegrenzten Eingabeband, das in Felder unterteilt ist und auf dem ein Lese-/Schreibkopf hin- und herbewegt wird. Es gibt auch Varianten mit links- oder rechtsseitig begrenztem Eingabeband, die sich bei der Mächtigkeit aber nicht vom Grundmodell unterscheiden. Die Maschine befindet sich zu jedem Zeitpunkt in einem von endlich vielen Zuständen und der Lese-/Schreibkopf steht jeweils genau über einem Feld des Bandes. Sie kann erkennen, welches Zeichen auf dem aktuellen Feld steht und kann es durch ein (gleiches oder verschiedenes) Zeichen überschreiben.

Ein Arbeitsschritt einer Turingmaschine besteht also daraus, in Abhängigkeit von dem Zeichen, das der Lese-/Schreibkopf gerade liest sowie dem aktuellen Zustand das Feld auf dem Band mit einem Zeichen zu beschreiben, ggf. eine Bewegung auf dem Eingabeband nach links oder rechts zu machen und in einen neuen Zustand überzugehen.

Turingmaschinen gibt es in vielen Varianten, die aber in ihrer Mächtigkeit gleichwertig sind und sich gegenseitig simulieren lassen, so z. B. welche mit einseitig beschränktem Eingabeband oder mehreren Bändern und Lese-/Schreibköpfen. Größere Eingabealphabete lassen sich durch Turingmaschinen mit kleinem Alphabet (z. B.  $\{0,1\}$ ) simulieren, indem eine andere Zeichencodierung (z. B. Binärdarstellung) verwendet wird.

Allgemein lassen sich (partielle) Funktionen  $f : N^k \rightarrow N$  mit einer Turingmaschine berechnen, indem man die Argumente in der Form  $x_1 \# x_2 \# \dots \# x_k$  auf das Band schreibt, den Lese-/Schreibkopf auf das erste Zeichen von  $x_1$  setzt und die Turingmaschine startet.

Wenn die Maschine nach einer endlichen Anzahl von Arbeitsschritten anhält, so ist das Ergebnis vom Band ablesbar, je nach Definition vor oder nach der letzten Position des Lese-/Schreibkopfs. Ist die Eingabe  $(x_1, x_2, \dots, x_k)$  eine für die Funktion  $f$  gültige, so ist das Ergebnis  $f(x_1, x_2, \dots, x_k)$ .

Ein Beispiel für ein mit Turingmaschinen nicht entscheidbares Problem ist das sog. *Halteproblem*. Es beschäftigt sich mit der Frage, ob eine Turingmaschine konstruiert werden kann, die für ein Paar aus einer anderen, kodierten Turingmaschine und einer Eingabe entscheiden kann, ob die kodierte Turingmaschine mit der gegebenen Eingabe terminiert.

Die Frage lautet also: Kann man eine Turingmaschine konstruieren, die als Eingabe die Kodierung einer anderen Turingmaschine sowie deren Eingabewerte erhält und die dann entscheidet, ob die andere Turingmaschine terminiert?

Turing gelang 1936 der Beweis (auf diesen wird im Rahmen der vorliegenden Arbeit verzichtet), dass es keine Turingmaschine gibt, die das Halteproblem für alle Eingaben lösen kann. Es gibt zwar keine Turingmaschine, die, wenn sie als Eingabe die Kodierung einer anderen Turingmaschine  $M$  und die dazugehörige Eingabe  $w$  erhält, „terminiert“ ausgibt, wenn  $M$  auf  $w$  hält und „terminiert nicht“ ausgibt, wenn  $M$  auf  $w$  nicht hält, sehr wohl gibt es aber eine Turingmaschine, die immer „terminiert“ ausgibt, wenn  $M$  auf  $w$  hält, aber endlos arbeitet, wenn  $M$  auf  $w$  nicht hält. Man spricht in diesem Fall von einem *semi-entscheidbaren Problem*.

Für die Softwareentwicklung bedeutet das, dass im Rahmen einer automatisierten Überprüfung von Programmen im Allgemeinen nicht feststellbar ist, ob diese jemals terminieren.

### 3 PROJEKTBLAUF

Das Projekt wurde im Schuljahr 2014/15 mit einer Gruppe des Wahlpflichtfachs Informatik (11. Schulstufe) durchgeführt. Insgesamt wurden sechs Doppelstunden dafür aufgewendet, die sich in jeweils zwei Doppelstunden für endliche Automaten, Turingmaschinen und Brainfuck aufteilen.

Noch vor der ersten Einheit und damit vor Beginn des eigentlichen Projekts wurden die Schüler aufgefordert, eine kurze Erhebung zur Interessenlage und den Vorkenntnissen auszufüllen (siehe Abschnitt „Evaluation“), um die Komplexität der geplanten Aufgabenstellungen ggf. anpassen zu können. Diese Maßnahme war allerdings nicht notwendig, weil erwartungsgemäß durchwegs keine Vorkenntnisse vorhanden waren.

Die Projektinhalte verteilen sich wie nachfolgend dargestellt auf die sechs Doppelstunden, die dazugehörigen Unterrichtsmaterialien finden sich im Anhang.

1. Thema „Endliche Automaten“
-------------------------------

- |  |
|--|
| <ul style="list-style-type: none"><li>a. Einführung in das Konzept von EA mit Beispielen</li><li>b. Eine mögliche Notation von EA: EBNF</li><li>c. Eine andere Notation von EA: graphische Darstellung</li><li>d. Gruppenarbeit: Konstruktion von EA aus EBNF-Angaben</li><li>e. EA in Kara: einfache Beispiele</li><li>f. EA/EBNF als Werkzeug zum Verständnis mathematischer/informatischer Sachverhalte</li><li>g. Kara-Beispiele mit höherer Komplexität</li></ul> |
|--|

2. Thema „Turingmaschinen“
----------------------------

- |  |
|--|
| <ul style="list-style-type: none"><li>a. Einführung in das Konzept von TM mit Beispielen, besonderes Augenmerk auf die Erweiterung von EA zu TM, um an vorhandenes Wissen anzuknüpfen</li><li>b. Eine mögliche Notation von TM: EA + Band</li><li>c. Exkurs: Binäres Zahlensystem (notwendig durch mangelndes Vorwissen der Schüler, kann ggf. entfallen) zum Verständnis der nachfolgenden TM-Aufgaben</li><li>d. Gruppenarbeit: Konstruktion von (einfachen) TM aus Textangaben</li><li>e. TM in Turingkara: Implementierung und Simulation der zuvor konstruierten TM</li></ul> |
|--|

3. Thema „Brainfuck“
----------------------

- |  |
|--|
| <ul style="list-style-type: none"><li>a. BF als mögliche Notation von TM</li><li>b. Äquivalenz BF-TM</li><li>c. Implementierung der TM-Beispiele in BF</li><li>d. BF-Beispiele mit höherer Komplexität</li></ul> |
|--|

## 4 EVALUATION

Bei diesem Projekt wurden zwei begleitende evaluierende Maßnahmen gesetzt:

Noch vor Beginn der ersten Einheit wurden die Schüler aufgefordert, eine kurze Erhebung zur Interessenlage und den Vorkenntnissen (Erhebungsbogen siehe Anhang) auszufüllen um die Komplexität der geplanten Aufgabenstellungen ggf. anpassen zu können. Beim Erhebungsbogen zielten die Fragen 1 und 2 auf mögliche Vorkenntnisse ab, die Fragen 3 und 4 sollten die Interessenlage beleuchten.

Die Auswertung ergab, dass keine Vorkenntnisse vorhanden waren und eine Differenzierung der Schülergruppe nach Leistungsniveau daher nicht notwendig war.

Die Ergebnisse im Detail:

Frage 1: Hast du bereits im letzten Jahr im Wahlpflichtfach Informatik mit endlichen Automaten oder Turingmaschinen gearbeitet?

Antworten: 0 „ja“, 6 „nein“

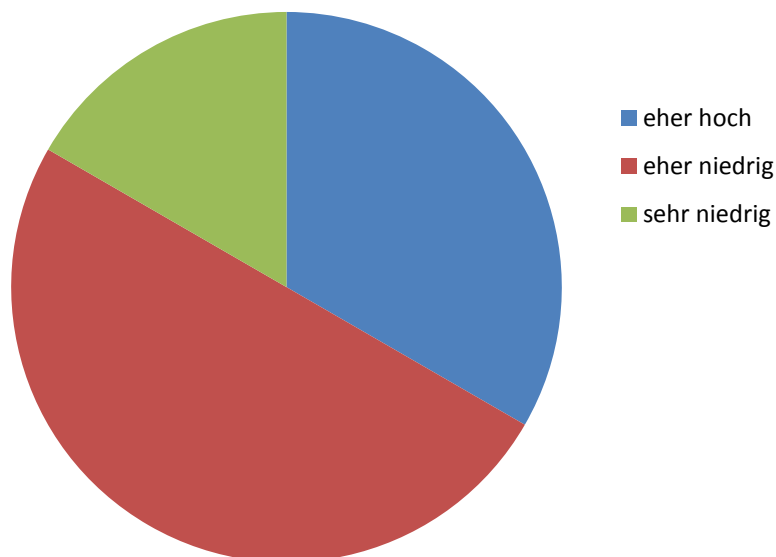
Frage 2: Hast du dich bereits aus eigenem Interesse mit endlichen Automaten oder Turingmaschinen beschäftigt?

Antworten: 0 „ja“, 6 „nein“

Frage 3: Wie würdest du dein Interesse, dich mit den Grundlagen der Informatik (also den Dingen, die hinter den Anwendungen stecken) zu beschäftigen, einschätzen?

Antworten: 0 „sehr hoch“, 2 „eher hoch“, 3 „eher gering“, 1 „sehr gering“

**Interesse an den Grundlagen der Informatik**

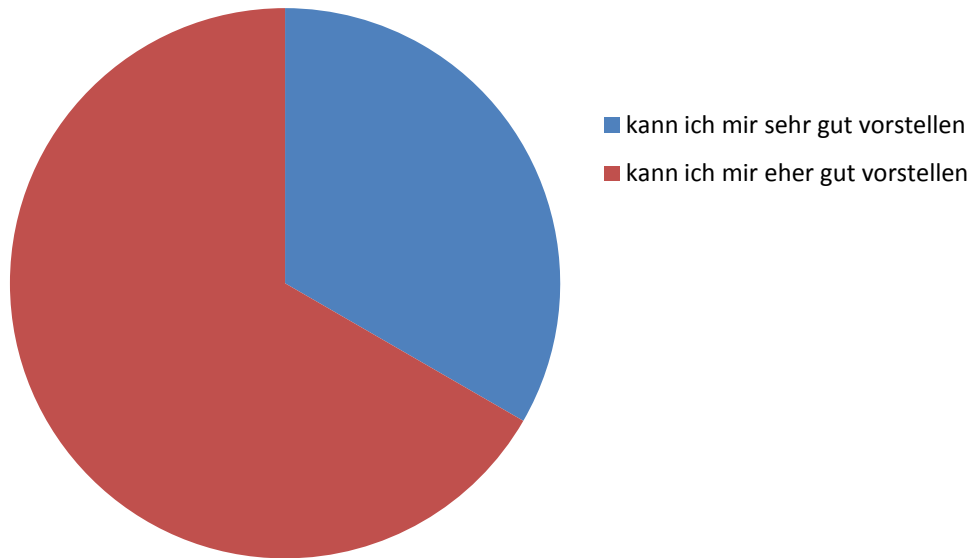




Frage 4: Kannst du dir vorstellen, dass die Inhalte des Informatikunterrichts auch ohne Computer-  
verwendung auf interessante Art vermittelt werden können?

Antworten: 2 „kann ich mir sehr gut vorstellen“, 4 „kann ich mir eher gut vorstellen“

**Informatikunterricht ohne Computer**



Die zweite durchgeführte Evaluation stellt der Abschlusstest am Ende der letzten Einheit dar (siehe Anhang). Die Ergebnisse werden nachfolgend in Diagrammform dargestellt:

	Aufg. 1 (20 Pkte.)	Aufg. 2 (30 Pkte.)	Aufg. 3 (25 Pkte.)	Aufg. 4 (25 Pkte.)	Aufg. 5 (20 Pkte.)	Aufg. 6 (30 Pkte.)	$\Sigma$ (150 Pkte.)
Schüler 1	20	30	25	25	20	30	150,00
Schüler 2	20	30	20	25	20	30	145,00
Schüler 3	15	10	15	25	20	5	90,00
Schüler 4	15	5	10	25	20	5	80,00
Schüler 5	10	5	5	20	10	5	55,00
Schüler 6	5	5	5	20	10	5	50,00
$\bar{x}$	14,17	14,17	13,33	23,33	16,67	13,33	
$\bar{x}$ (in %)	70,83	47,22	53,33	93,33	83,33	44,44	

Betrachtet man die Ergebnisse spaltenweise, stechen die Aufgaben 4 und 5 positiv hervor, die zu rund 93 bzw. 83 Prozent von den Schülern gelöst werden konnten. Aufgabe 1 liegt hinsichtlich der erreichten Punkte im Mittelfeld, während bei den Aufgaben 2, 3 und 6 nur wenige Punkte erreicht wurden.

Bei zeilenweiser Betrachtung (die Zeilen sind nach erreichter Punktzahl absteigend geordnet) stechen zwei Schüler besonders hervor, die (beinahe) die Höchstpunktzahl erreichen konnten. Es folgen zwei weitere Schüler, die noch mehr als die Hälfte der Punkte erreicht haben und weit abgeschlagen die beiden Schüler, die nur ein Drittel der Punkte bekamen.

## 4.1 Evaluation projektspezifischer Ziele

### 4.1.1 Inhaltliche Aspekte

Im Folgenden soll die Erreichung der Ziele auf SchülerInnenebene überprüft werden:

- **Konstruktion einfacher endlicher Automaten und Turingmaschinen**

Dieses Ziel wird im Abschlusstest in den Aufgaben 1, 3 und 4 überprüft. Nachdem diese Aufgaben mit rund 71%, 53% und 93% alle zu überwiegend erfüllt bzw. im Fall der Aufgabe 4 sogar in sehr großem Maße erfüllt wurden ist davon auszugehen, dass dieses Ziel erreicht wurde.

- **Grundlegendes Verständnis für Konzepte der theoretischen Informatik**

Auch wenn nach Dafürhalten des Autors durch das Projekt zumindest zu einem solchen Verständnis beigetragen wurde, lässt sich nicht ohne weiteres überprüfen, in welchem Ausmaß dieses auch erreicht wurde, zumal auch beim Abschlusstest keine Aufgabe direkte Schlüsse zulässt. Hier liegt das Problem wohl bei der Zielformulierung am Anfang des Projekts, das eventuell zu hoch gesteckt, vielleicht aber auch einfach nicht detailliert genug formuliert wurde.

- **Kennenlernen von verschiedenen Notationen endlicher Automaten und Turingmaschinen**

Die Repräsentation von Turingmaschinen (als Erweiterung von endlichen Automaten) sowohl als „pen and paper“-Konzept als auch formalisiert als BF-Programm hat auf jeden Fall zur gewünschten Erkenntnis, dass sich informatische Sachverhalte auf unterschiedliche Weise darstellen lassen, beigetragen.

- **Beschäftigung mit komplexen Aufgabenstellungen insbesondere bei der Implementierung der Turingmaschinen in Brainfuck**

Es war dem Autor ein Anliegen, jeweils nach der Einführung in das verwendete Werkzeug (Kara, Turingkara, BF) mit komplexitätsreduzierten Beispielen, die Schüler auch zum Lösen komplexerer Aufgaben anzuleiten, was insbesondere beim Abschlussbeispiel aus BF gelungen scheint.

- **Förderung der Teamarbeit beim Lösen von Gruppenaufgaben**

Schon nach der ersten Unterrichtseinheit mit den verschiedenen Sozialformen war offenkundig, dass die SchülerInnen bereits reichlich Erfahrung mit Gruppenarbeiten hatten und keiner speziellen Förderung in diese Richtung bedurften. Auch wenn der Autor kein Experte bezüglich Sozialformen ist, gilt für ihn das Ziel als erreicht. Nach seiner Überzeugung werden durch ständiges Anwenden verschiedener Sozialformen diese fest im täglichen Arbeitsablauf verankert.

- **Wecken von Interesse und Erkennen des eigenen Talents für informatische Sachverhalte und Aufgabenstellungen**

Auch wenn es sich bei den Schülern natürlich um solche handelt, die sich bewusst für das Wahlpflichtfach Informatik entschieden haben und denen damit ein gewisses „Grundinteresse“ für informatische Sachverhalte unterstellt werden kann, ist der Autor davon überzeugt, dass die Themen der theoretischen Informatik mit ihrem direkten Zugang zu informatischen Fragestellungen für das gestellte Ziel besonders wertvoll und daher auch im AHS-Unterricht Platz haben sollen.

Auf LehrerInnenebene wurden die folgenden Ziele definiert:

- **Gestaltung eines Themenschwerpunktes „Endliche Automaten und Turingmaschinen im Informatikunterricht“**

Die Rückmeldungen des Kollegen, der während des gesamten Unterrichtsprojekts in der Klasse anwesend war, sprechen dafür, dass es sich dabei um einen Schwerpunkt handelt, der wohl wahrscheinlich auch in verschiedenen Schultypen der AHS und BHS umgesetzt werden kann. Die Möglichkeiten insbesondere der Programmiersprache Brainfuck sind beim dargestellten Projekt noch lange nicht erschöpft, was auch für eine Verwendung in HTL spricht.

- **Richtiger Einsatz didaktischer Software**

Auch wenn sich die Erfahrung des Autors auf diesem Gebiet auf seine Unterrichtstätigkeit beschränkt, ist nach den Aussagen des hospitierenden Kollegen davon auszugehen, dass die Menge und Intensität des Computereinsatzes den Anforderungen gerecht wurde.

- **Rollenwechsel zum „Coach“, der den SchülerInnen beim Lösen der Aufgaben zur Seite steht**

Bei diesem Ziel mag dem Autor vielleicht in die Hände gespielt haben, dass er seine Rolle nie als Lehrer, der „von oben herab“ auf die Schüler predigt, gesehen hat und daher auch kein „Rollenwechsel“ stattfinden musste.

- **Auffrischen/Erweitern des eigenen Methodenpools**

Gerade dieser Themenschwerpunkt bietet in puncto Methodenvielfalt einen großen Spielraum. Bei den Einheiten ohne Computereinsatz können zuerst im Plenum die Grundlagen erarbeitet und anschließend in Zweier- oder Dreiergruppen die Übungsphase gestaltet werden. Die Einheiten am Computer sind eher auf Einzelarbeit ausgelegt, was aber nicht heißt, dass diese ohne größere konzeptionelle Änderungen nicht auch für Teams gestaltet werden könnten.

- **Rückmeldung über und ggf. Anpassung der Unterrichtsmethoden und des –stoffes**

Wie so oft im Schulunterricht hätte vielleicht auch im vorliegenden Projekt mehr verfügbare Unterrichtszeit den Unterrichtsertrag noch erhöht. Nach seinen Eindrücken befragt, würde der Autor wohl angeben, dass aus seiner Sicht zwei Schüler den behandelten Stoff zur Gänze und die anderen vier überwiegend verstanden haben. Für diese vier Schüler wäre wahrscheinlich eine längere Übungsphase nötig gewesen, um sich intensiv mit dem Gelernten auseinanderzusetzen.

### **4.1.2 Prozessaspekte**

Im Großen und Ganzen konnte der Zeitplan (siehe Kap. 1.2) eingehalten werden, auch wenn sich im Detail die Planung mit jeweils zwei Doppelstunden für endliche Automaten, Turingmaschinen bzw. BF als zu starr erwies.

Der Autor würde für den Fall, dass er das Projekt in den kommenden Jahren erneut durchführen sollte, eine dynamischere Zeiteinteilung vornehmen. Diese würde Möglichkeiten für gegebenenfalls notwendige verlängerte Übungsphase einschließen.

## **4.2 Evaluation aus Sicht übergeordneter IMST Ziele**

### **4.2.1 Gender- und Diversitätsaspekte**

Die Tatsache, dass die Umsetzung des IMST-Projekts mit einer männlichen Schülergruppe erfolgte, wirft natürlich die Frage auf, ob und welche Anpassungen für die Verwendung in gemischten Gruppen nötig wären.

Aus Sicht des Projektnehmers können die Unterrichtsmaterialien ohne Adaptionen auch für koedukative Gruppen verwendet werden, da die Formulierungen geschlechtsneutral und im Falle von Handlungsaufforderungen in der zweiten Person gehalten wurden.

Als Diversitätsaspekt sei angeführt, dass bei Verwendung der Materialien im Erwachsenenbildungsbereich bei den Handlungsaufforderungen die Höflichkeitsform verwendet werden sollte.

### **4.2.2 Schulentwicklungs- und Disseminationsaspekte**

Als Auswirkung auf die Unterrichtsentwicklung an unserer Schule sei an dieser Stelle angeführt, dass ein weiterer Informatiklehrer an den meisten Einheiten teilgenommen und in adaptierter Form bereits in seinen Unterricht des Wahlpflichtfachs der achten Klasse eingebaut hat. Dadurch hat das Thema auch Eingang in den Themenpool der Informatikmatura Eingang gefunden.

Die bei der Abfassung dieses Berichts gesammelten Unterrichtsmaterialien stehen auch allen Fachkollegen der Schule über das Netzwerk zur Verfügung, darüber hinaus auch die persönliche Expertise des Autors, die auch im Rahmen der regelmäßigen Fachkonferenzen zur Unterrichtsentwicklung beiträgt und den Stellenwert der theoretischen Informatik im Lehrplan stetig in Erinnerung ruft.

Neben der Veröffentlichung des Projektberichts im IMST-Wiki wurde dem Autor die Möglichkeit eingeräumt, sein Projekt auf dem IMST-Tag am 20. März 2015 im Haus der Industrie in Wien vor großem Publikum zu präsentieren.

## 5 ZUSAMMENFASSUNG

Während in den Medien von der „wissensbasierten Gesellschaft“ die Rede ist und Unternehmen mit der Entwicklung von Hard- und Software viel Geld verdienen, sehen wir an den Universitäten einen Rückgang der Studierendenzahlen im informatischen Bereich. Die Politik versucht ebenfalls, insbesondere Mädchen für Informations- und Kommunikationstechnologie zu begeistern, was allerdings angesichts des breiten Spektrums dieses Begriffs nicht einfach ist.

Erfolgreiche Unternehmen wie Google oder Facebook benötigen innovative und gut ausgebildete Entwickler, die über Kompetenzen in vielen Bereichen der Informatik verfügen, darunter auch ein gutes Verständnis von Zahlensystemen und Operationen mit Zahlen, logischen Verknüpfungen und grundlegender Programmierung.

Wie auch Tim Bell in seinem Paper über seine Beweggründe zur Konzeptionierung von CS Unplugged schreibt, steht im Informatikunterricht leider allzu oft die ausschließliche Vermittlung von Werkzeugwissen und oberflächliches Programmieren am Stundenplan. Das vorliegende Projekt stellt eine Möglichkeit dar, aus diesem Schema auszubrechen und sich fundamentalen Konzepten zu widmen, ohne die Aufmerksamkeit der SchülerInnen zu verlieren.

Neben einer auf das Wesentliche reduzierten Darstellung der erforderlichen fachlichen Grundlagen liefert dieser Projektbericht einen umfassenden Ablaufplan sowie die benötigten Materialien für eine mögliche Implementierung dieses Themenschwerpunkts.

Der Autor würde sich freuen, wenn dieser von LehrerInnen als Grundlage zur Adaption an den eigenen Unterricht verwendet würde.

## 6 QUELLEN

Brainfuck (esoterische Programmiersprache). Entwickelt von Urban Müller im Jahr 1993. Online unter <http://esolangs.org/wiki/Brainfuck> [20.02.2015]

Herold Helmut, Lurz Bruno & Wohlrab Jürgen (2007): Grundlagen der Informatik. München: Pearson.

# ANHANG 1: STARTERHEBUNG

## Starterhebung „Endliche Automaten/Turingmaschinen/Brainfuck“

Name:

Klasse:

1. Hast du bereits im letzten Jahr im Wahlpflichtfach Informatik mit endlichen Automaten oder Turingmaschinen gearbeitet?  
 ja  nein
2. Hast du dich bereits aus eigenem Interesse mit endlichen Automaten oder Turingmaschinen beschäftigt?  
 ja  nein
3. Wie würdest du dein Interesse, dich mit den Grundlagen der Informatik (also den Dingen, die hinter den Anwendungen stecken) zu beschäftigen, einschätzen?  
 sehr hoch  eher hoch  eher gering  sehr gering
4. Kannst du dir vorstellen, dass die Inhalte des Informatikunterrichts auch ohne Computerverwendung auf interessante Art vermittelt werden können?  
 kann ich mir sehr gut vorstellen  
 kann ich mir eher gut vorstellen  
 kann ich mir eher nicht vorstellen  
 kann ich mir gar nicht vorstellen

## ANHANG 2: ABSCHLUSSTEST

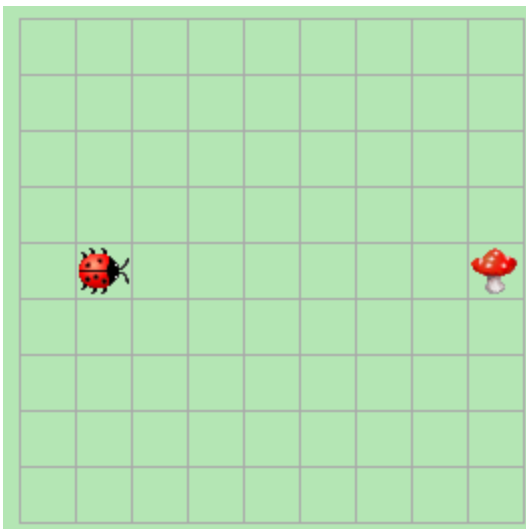
### Abschlusstest „Endliche Automaten/Turingmaschinen/Brainfuck“

Name:

Klasse:

1. Entwirf nachfolgend einen endlichen Automaten, der einen einfachen Zuckerlautomat simuliert. (20 Punkte)
  - a. Es werden Münzen zu 10 und 20 Cent akzeptiert
  - b. Es gibt zwei verschiedene Zuckerl: Eiszuckerl um 20 Cent und Nimm2 um 30 Cent

2. Schreib ein Programm, das Kara geradeaus bis zum nächsten Pilz führt. Lass Dein Programm in einer Welt ablaufen, die so aussieht wie am nachfolgenden Bild dargestellt. (30 Punkte)



Wenn Dein Programm fertig ist, speichere es unter K:\Kara\deinname.kara, wobei Du „deinname“ durch Deinen Familiennamen ersetzt.





5. Schreib nachfolgend ein kommentiertes Brainfuck-Programm, das die Werte aus den Zellen 2 und 4 addiert und die Summe in Zelle 6 speichert. (20 Punkte)

6. Schreib nachfolgend ein kommentiertes Brainfuck-Programm, das den Text „BRAINFUCK“ auf dem Bildschirm ausgibt. Verwende dabei zur Vorbelegung der Zellen mit den Zahlenwerten (siehe Tabelle) geeignete Schleifen! (30 Punkte)

B	66
R	82
A	65
I	73
N	78
F	70
U	85
C	67
K	75