FETCH- EXECUTE

...das etwas andere Computerspiel!

Welche Motivation steckt hinter dem Spiel?

„Die Anwendungen des Computers drängen seine Theorie und Technik immer mehr in den Hintergrund. Das führt nicht nur zu einer Neubestimmung der Informatik, sondern auch zu einer Besinnung darüber, was die Grundgedanken der Informatik, ihre Kernideen sind.“

Peter Rechenberg, „Was ist Informatik?“,
In: Informatik Spektrum (2010), Volume 33, Nr. 1, S. 54-60

Ein zeitgemäßer Informatikunterricht muss mehr als den „sicherer Umgang mit Standardsoftware“ vermitteln. So steht es auch im Lehrplan der 9. Schulstufe für die allgemein bildenden höheren Schulen in Österreich. Um den Schülerinnen und Schülern ein umfassenderes und spannenderes Bild von Informatik zugänglich zu machen, habe ich im Rahmen meiner Diplomarbeit offen lizenzierte Lehr- und Lernmaterialien für den Informatik-Unterricht entwickelt.

Das Spiel „Fetch-Execute“ wurde speziell für den Einsatz im Informatikunterricht ab der 9. Schulstufe konzipiert. Es basiert auf der Lerneinheit H3 („Hardware, EVA-Prinzip, Von-Neumann-Architektur“) von Ernestine Bischof und Roland Mittermeir, welche im Rahmen des Projektes „Informatik erLeben“ an der Alpen-Adria-Universität Klagenfurt erstellt wurde.[1] Es soll auf spielerische Weise zeigen, wie ein Computer im „Inneren“ aussieht und funktioniert. Dabei wird nicht nur die Aufgabe der einzelnen Bauteile besprochen, sondern auch maschinennah programmiert. Die Zusammenarbeit von CPU, Hauptspeicher, Festplatte, Bus und Eingabe-Ausgabegeräten wird im Rahmen des Spieles „erlebbar“.

---

[1] Ernestine Bischof & Roland Mittermeir, Universität Klagenfurt (2009),
Informatik erLeben, Lerneinheit H3: *Hardware, EVA-Prinzip, Von-Neumann-Architektur*,
Link: http://informatik-erleben.uni-klu.ac.at/einheiten/h/h3/ (letzter Aufruf: 05.01.2017)

Lizenzbedingungen

Dieses Material steht unter der **Creative-Commons-Lizenz Namensnennung 4.0 International**:

Um eine Kopie dieser Lizenz zu sehen, besuchen Sie http://creativecommons.org/licenses/by/4.0/.

Hinweis zur sprachlichen Gleichbehandlung

Aus Gründen der Lesbarkeit wird bei der Spielanleitung darauf verzichtet, geschlechtsspezifische Formulierungen zu verwenden.
Sind personenbezogene Bezeichnungen nur in männlicher Form angeführt, so beziehen sie sich auf Männer und Frauen in gleicher Weise.

Spielanleitung

Eine Spielgruppe besteht aus **3 Personen**:

Spieler 1
Spieler 2
Spielleiter

Der Spielleiter steuert den Ablauf des Spieles und kontrolliert die Spielzüge der beiden Spieler.

1. Unterlagen und Materialien

Der Spielleiter erhält zu Beginn des Spieles die folgenden Unterlagen und Materialien:

- Übersichtsplan
- Spielblatt für Spieler 1
- Spielblatt für Spieler 2
- Spielfeld CPU
- Spielfeld Hauptspeicher
- Spielfeld Bus
- Spielfeld Festplatte
- 1 Tastatur = Eingabeeinheit (in Papier-Form)
- 1 Monitor = Ausgabeeinheit (in Papier-Form)
- 3x Befehlssatz
- 4 Schutzkarten
- Aktionskarten
- Testprogramm
- Programm 1 von Spieler 1
- Programm 1 von Spieler 2
- Programm 2 von Spieler 1
- Programm 2 von Spieler 2
- leere Befehlskärtchen
- 1 Pfeil
- 1 Würfel
- 1 Bleistift
- 1 Radierer

Offen lizenzierte Lehr- und Lernunterlagen für den Informatikunterricht
Maria Grandl

2. Theoretische Einführung

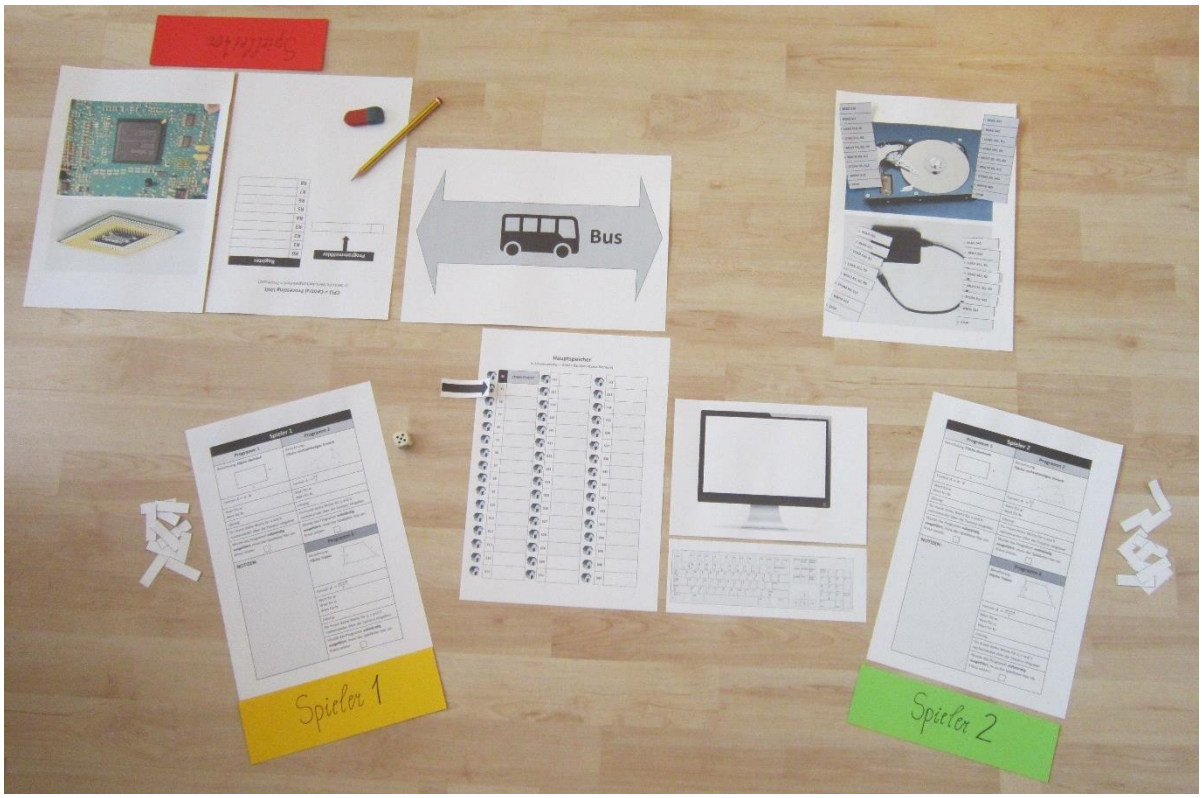
Der Spielleiter übernimmt die theoretische Einleitung und den Aufbau des Spielfeldes.

	Herzlich Willkommen zum Spiel *Fetch- Execute*. Heute spielen wir ein etwas anderes Computerspiel. Die folgenden Fragen und Erklärungen sollen zum besseren Verständnis des Spieles beitragen und werden in Form von Quiz-Fragen während des eigentlichen Spieles erneut gestellt. Das bedeutet: Gut Aufpassen! Gleich zu Beginn zwei Fragen:
	Was versteht ihr unter einem Automaten?
	Könnt ihr Beispiele für Automaten nennen?
	Spielleiter wartet auf Antworten der Spieler
	Beispiele für Automaten sind: alle Arten von Getränke – und Snackautomaten, Fernbedienung, Radio, Computer, etc.
	Was haben alle diese Automaten (=informationsverarbeitende Systeme) gemeinsam?
	Spielleiter wartet auf Antworten der Spieler
	Es gibt Ein- und Ausgabeelemente. Die Verarbeitungselemente und Vorgänge, die zwischen Eingabe und Ausgabe erfolgen, sind meistens nicht sichtbar.
	Welche Eingaben und Ausgaben erfolgen bei einem Getränkeautomaten?
	Spielleiter wartet auf Antworten der Spieler
	Eingabe: Geld und Knopfdruck, Ausgabe: gewähltes Produkt
	Was entspricht der Ein- und Ausgabe bei Computern?
	Eingabe: Befehle, Daten (über Maus, Tastatur, Mikrofon, Scanner, etc.) Ausgabe: grafische Ausgabe über Monitor, Ton über Lautsprecher, etc.
	Jede künstliche Verarbeitung von Daten erfolgt also nach dem EVA-Prinzip. EVA steht für **E**ingabe-**V**erarbeitung-**A**usgabe.
	Wir werden uns in diesem Spiel näher damit befassen, was im „Inneren“ eines Computers vorgeht, also wie die Verarbeitung von Daten erfolgt und was genau bei der Ausführung eines Computerprogrammes passiert. Um das zu verstehen, müssen wir uns zuerst damit beschäftigen, aus welchen Bauteilen (Hardware) sich ein Computer zusammensetzt.
	Unser vereinfachter Computer besteht im Wesentlichen aus 6 Teilen.
	Übersichtsplan aushändigen
	Beginnen wir mit der CPU. Die CPU (= **C**entral **P**rocessing **U**nit = Zentrale Verarbeitungseinheit = Prozessor) ist das „Gehirn“ des Computers. Sie übernimmt alle Rechenoperationen und setzt sich aus mehreren Teilen zusammen.
	Das Steuerwerk (auch Controller genannt) erzeugt die Steuersignale für die weiteren Komponenten der CPU und des Computers.
	Das Rechenwerk (= ALU = **A**rithmetic **L**ogical **U**nit) besteht aus Hilfs- und Statusregistern und dem Schaltnetz, das die Rechenoperationen durchführt. Register sind Speicher mit super-schneller Zugriffszeit.
	Spielfeld CPU aushändigen
	Wir sehen hier eine schematische Darstellung der CPU. Das Rechenwerk besteht hier aus 8 Registern R1-R8. Der Programmzähler ist ein Register, in dem die Speicheradresse des Befehles steht, der soeben vom Steuerwerk aus dem Hauptspeicher geholt wurde. Der Programmzähler „zeigt“ also immer auf eine Speicheradresse im Hauptspeicher.
	Die Arbeit der CPU übernehme ich als Spielleiter.

Offen lizenzierte Lehr- und Lernunterlagen für den Informatikunterricht
Maria Grandl

	Spielfeld Hauptspeicher aushändigen
	Die CPU versteht Maschinensprache und kann ein Maschinenprogramm ausführen. Das Programm und die zu verarbeitenden Daten müssen sich dabei im Hauptspeicher (=Arbeitsspeicher = RAM = **R**andom Access **M**emory) befinden. Die CPU kann direkt auf die Speicheradressen und somit auf den Speicherinhalt des Hauptspeichers zugreifen. Ein Maschinenbefehl wird zuerst vom Hauptspeicher in die CPU geholt (fetch) und danach ausgeführt (execute). Die berechneten Ergebnisse kommen wieder in den Hauptspeicher zurück und können dort von der Ausgabeeinheit abgeholt werden.
	Der Hauptspeicher enthält im Moment noch keine Programme oder Daten. Ziel des Spieles ist es, ein oder mehrere Programme und die benötigten Daten in den Hauptspeicher zu laden und mit viel Glück die Aufmerksamkeit der CPU zu bekommen, welche dann Befehl für Befehl des Programmes ausführt.
	Legt das Spielfeld Hauptspeicher so vor euch hin, dass es beide gut erreichen können.
	Spielfeld BUS aushändigen
	Das Bussystem eines Rechners ist dafür zuständig, Daten zwischen den einzelnen Einheiten des Computers zu transportieren. Der Eingabe-/Ausgabe-Bus ist für den Datentransport zwischen externen Geräten (Eingabe, Ausgabe, externe Speicher) und dem Hauptspeicher zuständig. Für die Kommunikation zwischen Arbeitsspeicher und den schnellen Registern der CPU sorgt der Daten- bzw. der Befehlsbus. Wir wollen zur Vereinfachung allerdings nur von einem gemeinsamen Bus ausgehen.
	(Papier-)Tastatur aushändigen
	Legt die Tastatur so vor euch hin, dass sie beide gut erreichen können.
	Die Eingabeeinheit hat die Aufgabe, Eingaben an den Computer weiterzuleiten, indem sie diese an den Bus weitergibt. Von dort geht die Eingabe zuerst in den Hauptspeicher. Während der Ausführung eines Programmes (z.B. Taschenrechner) sind die Eingaben (Zahlen), die vom Programm zu verarbeitenden Daten.
	(Papier-)Monitor aushändigen
	Legt den Monitor so vor euch hin, dass er für beide gut erreichbar ist.
	Die Ausgabeeinheit dient dazu, die berechneten Ergebnisse wieder in eine für den Menschen lesbare Form zu bringen (Monitor, Drucker, Lautsprecher, etc.). Sie erhält die Daten über den Eingabe-/Ausgabe-Bus aus dem Arbeitsspeicher.
	Festplatte aushändigen
	Die Festplatte kann als Standardbeispiel für externe Permanentspeicher angesehen werden. Informationen, die auf der Festplatte stehen, bleiben dort erhalten, auch wenn man den Computer ausschaltet. Damit ein auf der Festplatte gespeichertes Programm von der CPU ausgeführt werden kann, muss es zuerst in den Hauptspeicher geladen werden.
	Jetzt bin ich mit den Erklärungen schon fast am Ende. Wir wollen nun ein erstes eigenes Programm schreiben bzw. programmieren, das später von der CPU ausgeführt werden soll. Das Programm soll die Fläche eines Quadrats mit der Seitenlänge a = 5 berechnen. Damit wir diese Rechenaufgabe mithilfe des Computers lösen können, müssen wir dessen Befehlssatz oder Instruktionssatz kennen.
	Das Spielfeld sollte nun etwa folgendermaßen aussehen:

Offen lizenzierte Lehr- und Lernunterlagen für den Informatikunterricht
Maria Grandl



An jeden Spieler einen Befehlssatz aushändigen. Selbst einen Befehlssatz behalten.

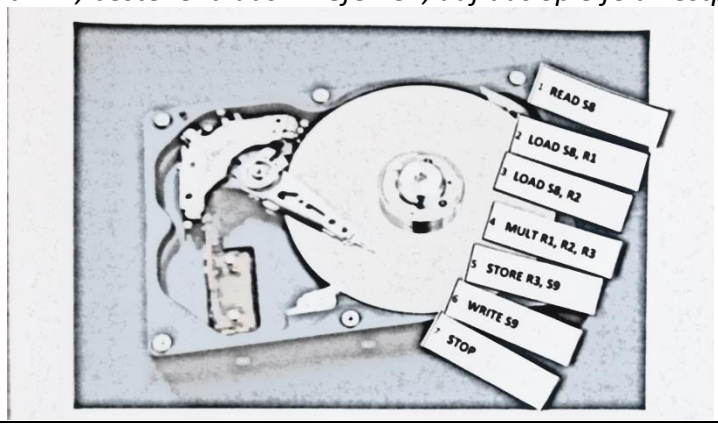
Unterschiedliche CPUs haben eine unterschiedliche Architektur und daher auch unterschiedliche Befehlssätze. Der Befehlssatz ist die Menge aller Befehle, welche die CPU verarbeiten kann. Auf dem Blatt findet ihr eine Übersicht über die zulässigen 8 Befehle READ, WRITE, LOAD, STORE, ADD, MULT, MULTF und STOP.

Ihr habt nun 5 Minuten Zeit, den Befehlssatz durchzugehen.

Auch der Spielleiter muss mit dem Befehlssatz umgehen können. Achte darauf, dass die Zeit von 5 Minuten eingehalten wird. Sollten Unklarheiten auftreten, dann besprecht diese bitte in der Spielgruppe oder bittet eure(n) Lehrer/in um Hilfe.

Nun werden wir gemeinsam ein erstes Testprogramm zur Berechnung der Fläche eines Quadrats durchgehen.

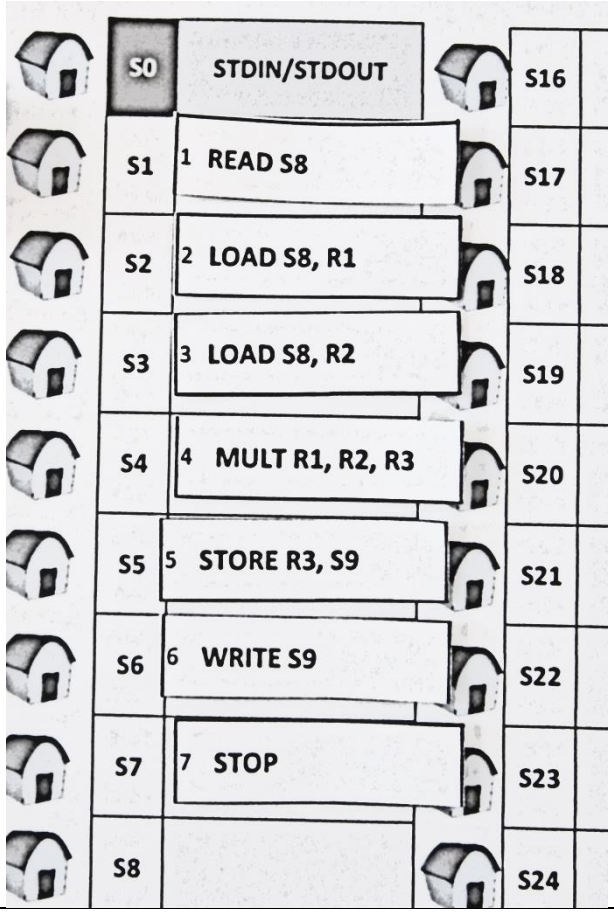
Lege das Testprogramm, bestehend aus 7 Befehlen, auf das Spielfeld Festplatte.



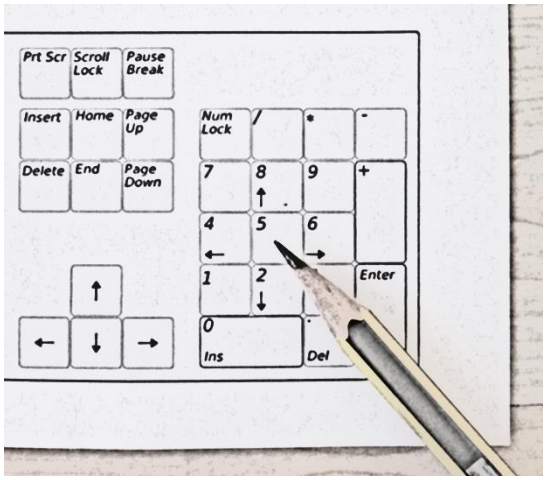
Wie kommt nun das Testprogramm in den Hauptspeicher?
Stellt euch vor, das Programm kann durch einen Doppelklick auf ein entsprechendes Icon am Desktop gestartet werden. Klickt nun doppelt auf das imaginäre Icon. Dadurch wird das

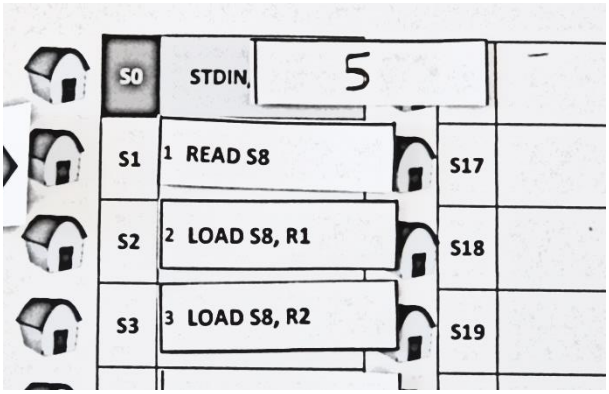
Offen lizenzierte Lehr- und Lernunterlagen für den Informatikunterricht
Maria Grandl

Programm von der Festplatte in den Hauptspeicher geladen. Dem Programm wird ein bestimmter Adressbereich im Hauptspeicher (User-Space) zur Verfügung gestellt. Der erste Befehl des Programmes kommt in das Haus mit der Adresse S1, der zweite Befehl kommt in das Haus mit der Adresse S2, usw.



Das Programm erwartet eine Eingabe, nämlich den Wert für die Seitenlänge **a** des Quadrates. Wir wählen für **a = 5**. Wir schreiben den Wert 5 auf ein leeres Befehlskärtchen und speichern diesen Wert im Haus mit der Speicheradresse **S0**, dem **Standard-Input**.
Die Speicheradresse S0 wird für Input und Output verwendet.



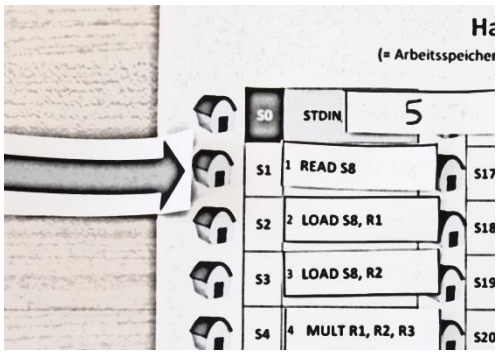


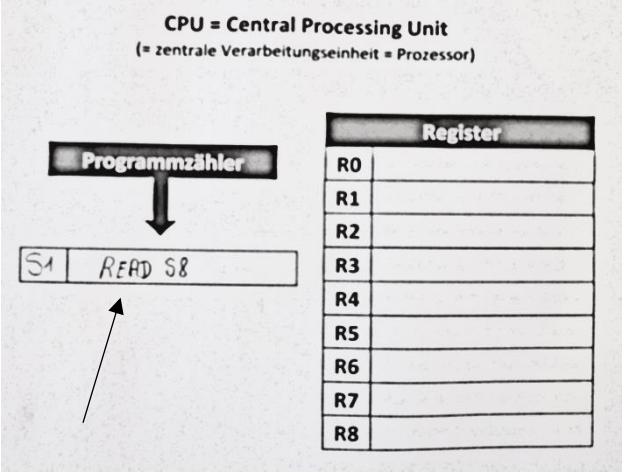
Nun wird der erste Befehl ausgeführt. Der Programmzähler muss dabei auf den ersten Befehl des Programmes zeigen, was durch den Pfeil symbolisiert wird. Der Wert des

CC BY

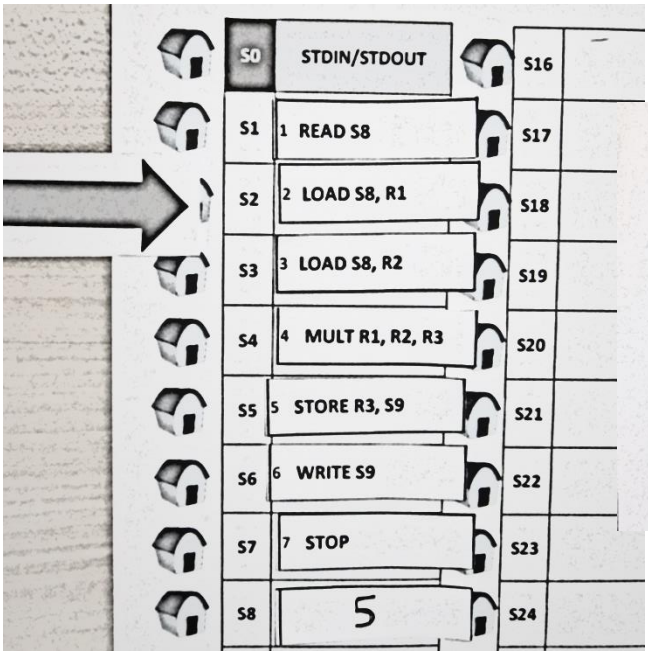
Offen lizenzierte Lehr- und Lernunterlagen für den Informatikunterricht
Maria Grandl

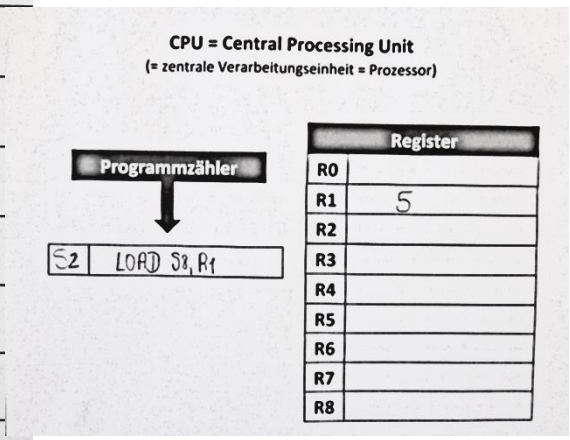
Programmzählers muss auf **S1** gesetzt werden. Der Spielleiter muss dies mit Bleistift eintragen. Der Befehl wird nun in die CPU geholt (fetch) und ausgeführt (execute).





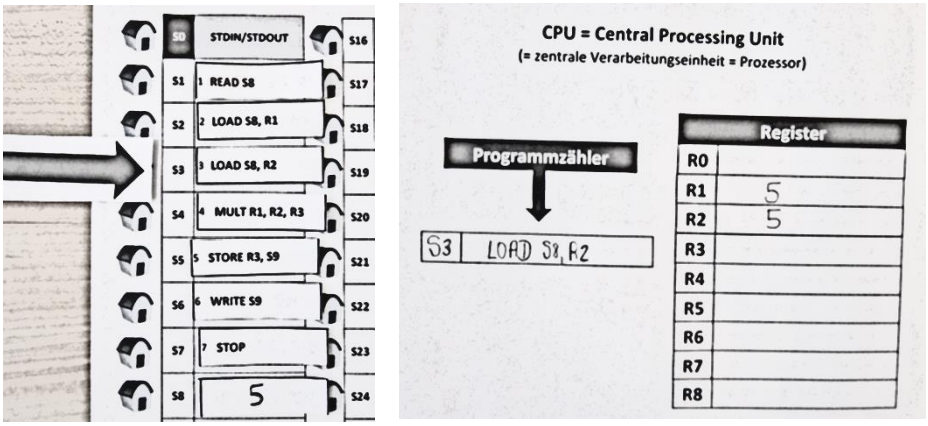
Die Ausführung des Befehles bewirkt, dass die Eingabe, also die Zahl 5, in das Haus mit der Speicheradresse **S8** gespeichert wird. Der Programmzähler wird um 1 erhöht und zeigt jetzt auf die Speicheradresse des Befehles **S2**. Der Spielleiter muss den Wert des Programmzählers aktualisieren. Der Befehl wird nun in die CPU geholt und ausgeführt. Er bewirkt, dass der Wert, der im Haus mit der Speicheradresse **S8** gespeichert ist, ins Register **R1** geschrieben wird. Zur Wiederholung: Register sind ebenfalls Speicher.

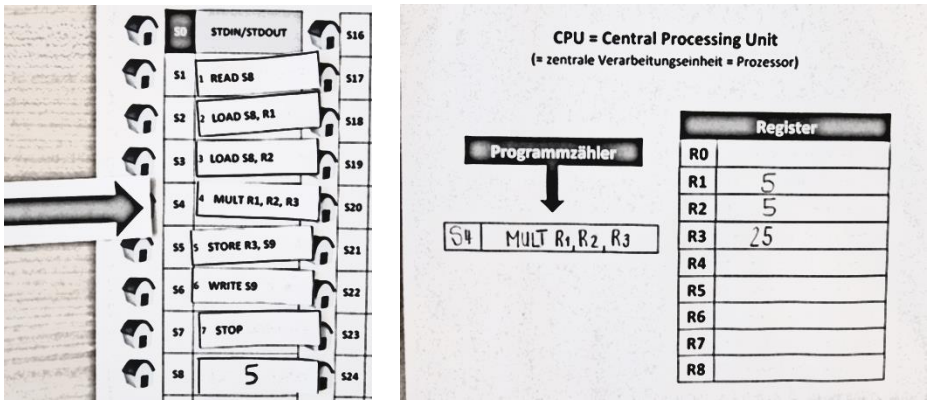


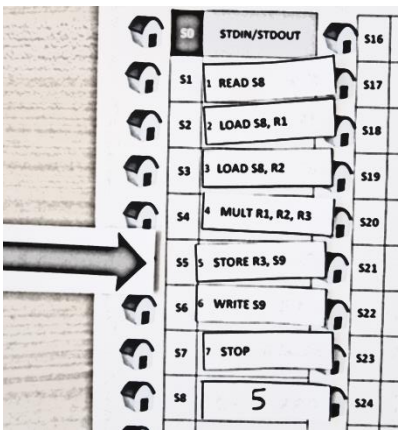


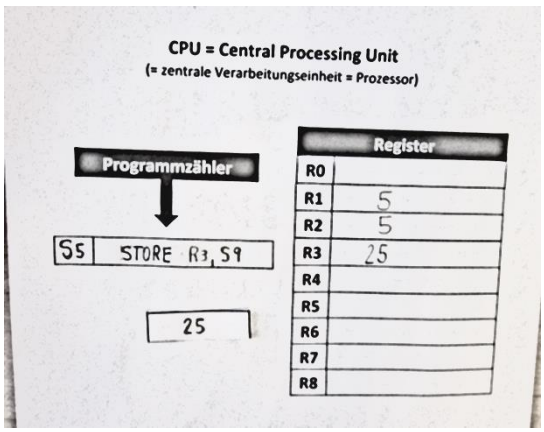
Die weiteren Schritte werden hier bildlich dokumentiert und sind zur Kontrolle gedacht.

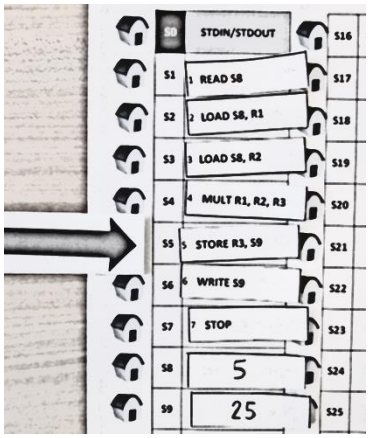
Offen lizenzierte Lehr- und Lernunterlagen für den Informatikunterricht
Maria Grandl



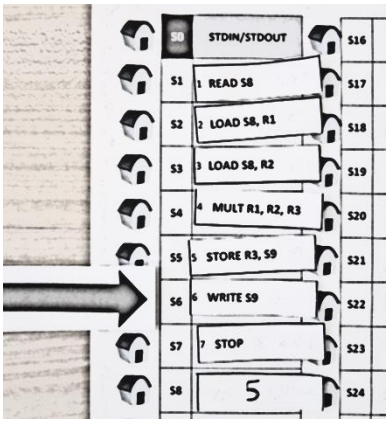


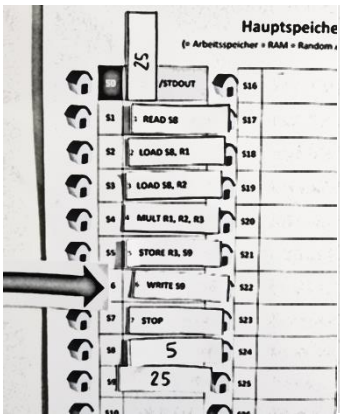


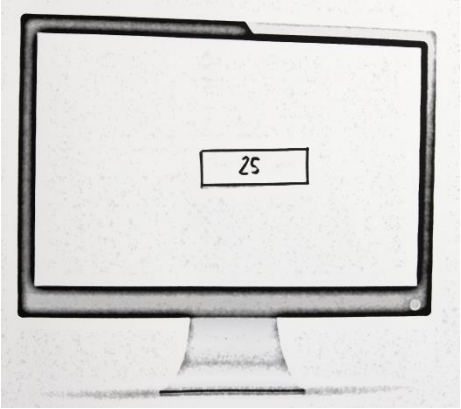




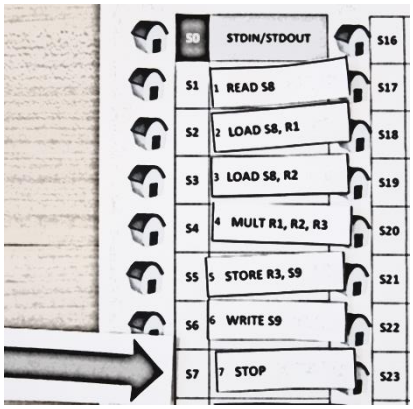
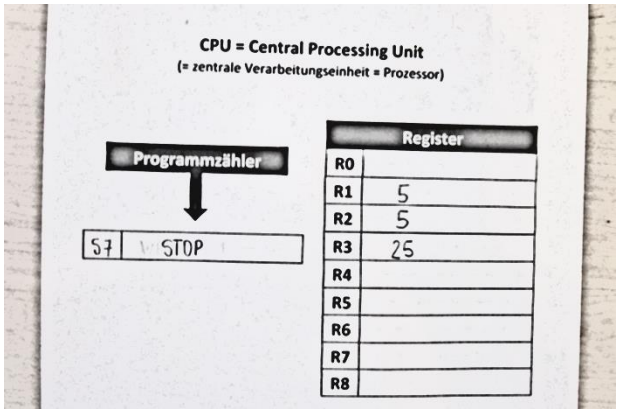
Die Ausführung des Befehles **WRITE S9** bewirkt, dass der Wert der im Haus mit der Speicheradresse **S9** gespeichert ist, also die Zahl 25, auf den **Standard-Output** geschrieben wird. Das bedeutet, dass die Zahl 25 im Haus mit der Speicheradresse **S0** gespeichert wird. Die Zahl 25 wird dann am **Monitor** ausgegeben.







CC BY

	Nachdem die CPU den Befehl STOP ausgeführt hat, wird das Programm beendet.  
	Man kann also sagen, dass die CPU den **FETCH-EXECUTE-Algorithmus** ausführt, der namengebend für dieses Spiel ist: **Wiederhole** **{** **Hole den nächsten Befehl aus dem Hauptspeicher.** **Führe diesen Befehl aus.** **}** Ein Algorithmus ist nichts anderes als eine eindeutig festgelegte Abfolge von endlich vielen Anweisungen zur Lösung eines bestimmten Problems. Jede Anweisung kann in endlicher Zeit ausgeführt werden.
	Nun geht es zum eigentlichen Spiel. Ihr erhält nun beide jeweils 1 Spielblatt.
	Spielblätter an die Spieler aushändigen.
	Die beiden Programme, die ausgeführt werden sollen, benötigen jeweils 2 Eingaben. Die Werte für die Variablen können beliebig gewählt werden. Sobald das Programm in den Hauptspeicher geladen wurde, müsst ihr die Werte über die Tastatur eingeben.
	Achtet darauf, dass eure Programme nur in den angegebenen Bereich im Speicher geladen werden dürfen.
	Programm 1 und Programm 2 jeweils an Spieler 1 und Spieler 2 aushändigen.
	Das dritte Programm muss selbständig programmiert werden. Dazu erhält ihr von mir Befehlskärtchen.
	Jeweils ungefähr 15 leere Befehlskärtchen an die Spieler aushändigen.
	Eure Programme sind auf der Festplatte gespeichert.
	Legt den Programmcode von Programm 1 und Programm 2 auf das Spielfeld Festplatte.

3. Jetzt wird gespielt!

Wer darf beginnen?

Beide Spieler würfeln nacheinander mit einem Würfel.
Der Spieler mit der höheren Augenzahl darf beginnen.
Bei gleicher Augenzahl muss erneut gewürfelt werden.

CC BY

Offen lizenzierte Lehr- und Lernunterlagen für den Informatikunterricht
Maria Grandl

Erste Würfelphase

(Die erste Würfelphase ist vergleichbar mit dem Beginn des Spieles „Mensch ärgere dich nicht“)

Jeder Spieler würfelt 3 Mal hintereinander.

Würfelt ein Spieler die Augenzahl **1,2,3,4** oder **5**, so muss er erneut würfeln.

Würfelt ein Spieler die Augenzahl **6**, so darf das Programm von der Festplatte in den auf dem Spielblatt angegebene User-Space im Hauptspeicher geladen werden. Er geht nun in die zweite Würfelphase über.

Danach ist der andere Spieler an der Reihe.

Zweite Würfelphase

Befindet sich der Programmcode im Hauptspeicher, so kann mit der Ausführung des Programmes begonnen werden. Dies hängt aber von der gewürfelten Augenzahl ab.

Augenzahl	
1	Es muss **1** Befehl ausgeführt werden.
	Es wird nur derjenige Befehl ausgeführt, auf den der Programmzähler „zeigt“. Der Spielleiter muss die richtige Speicheradresse mit Bleistift in das Feld „Programmzähler“ schreiben.
2	Es müssen **2** Befehle ausgeführt werden.
3	Es muss eine **Aktionskarte vom Spielleiter** abgehoben werden.
ℹ	Auf jeder Aktionskarte befindet sich eine Frage und die Antwort der Frage. Es gibt offene Fragen und Fragen mit vorgegebenen Antwortmöglichkeiten. Die Aktionskarte darf **nur vom Spielleiter** abgehoben und vorgelesen werden. Die Aktionskarte muss dann wieder unter den Kartenstapel gelegt werden. Wurde die Frage **richtig** beantwortet, so darf 1 Befehl ausgeführt werden. Wurde die Frage **falsch** beantwortet, so ist der andere Spieler an der Reihe.
4	Es muss eine **Aktionskarte** abgehoben werden.
5	Es muss eine **Schutzkarte** abgehoben werden.
ℹ	Die Schutzkarte schützt vor einem Computerabsturz, der durch das Würfeln der Augenzahl 6 hervorgerufen wird. Ein Spieler darf maximal 2 Schutzkarten besitzen. Sollte ein Spieler schon 2 Schutzkarten besitzen und erneut die Augenzahl 5 würfeln, so ist der andere Spieler an der Reihe.
6	Diese Augenzahl löst einen **Absturz des Computers** aus.
ℹ	Das bedeutet, dass alle Inhalte des Hauptspeichers und der CPU-Register verloren gehen. Der Spielleiter muss die Inhalte der CPU-Register leeren (ausradieren) und die Programme beider Spieler aus dem Hauptspeicher entfernen und wieder auf der Festplatte platzieren. Besitzt der Spieler eine Schutzkarte, so kann er einen Computerabsturz abwehren. Die Schutzkarte muss dann an den Spielleiter abgegeben werden.

Offen lizenzierte Lehr- und Lernunterlagen für den Informatikunterricht
Maria Grandl

Was passiert, wenn Programm 1 vollständig ausgeführt wurde?

Programm 1 wurde dann vollständig und richtig ausgeführt, wenn das Ergebnis auf dem Monitor ausgegeben wurde und der Programmzähler auf die Speicheradresse des Befehles STOP „zeigt“. Der Spielleiter muss das Programm nun aus dem Hauptspeicher nehmen und die verwendeten Register leeren, d.h. die Inhalte ausradieren. Der Spielleiter muss die vollständige Ausführung des Programmes am Spielerblatt des jeweiligen Spielers vermerken.

Der Spieler startet wieder mit der ersten Würfelphase.

Was passiert, wenn Programm 1 und Programm 2 vollständig ausgeführt wurden?

Programm 3 (siehe Spielerblatt) muss selbständig programmiert werden. Hier geht es um Schnelligkeit und Korrektheit. Die einzelnen Befehle müssen händisch auf die leeren Kärtchen geschrieben werden. Sobald ein Spieler mit dem Programm fertig ist, startet er wieder mit der ersten Würfelphase.

Anforderungen an Programm 3

Programm 3 soll die Fläche eines Trapez berechnen. Das Programm nimmt 3 Werte als Input, nämlich die Seitenlängen a und c und die Höhe h. Diese müssen über die Tastatur eingegeben werden. Das Ergebnis muss am Monitor ausgegeben werden. Um Konflikte zu vermeiden, darf das Programm nur in den vorgegebenen Adressbereich im Speicher geladen werden (siehe Spielerblatt). Außerdem dürfen nur die vorgegebenen Register verwendet werden (siehe Spielerblatt).

Mögliche Lösung für Programm 3:

```
S5:  READ S1
S6:  READ S2
S7:  READ S3
S8:  LOAD S1, R1
S9:  LOAD S2, R2
S10: LOAD S3, R3
S11: ADD R1, R2, R4
S12: MULT R3, R4, R4
S13: MULTF R4, 0.5
S14: STORE R4, S4
S15: WRITE S4
S16: STOP
```

Was passiert, wenn Programm 3 fehlerhaft ist?

Stellt ein Spieler, während der Ausführung von Programm 3, fest, dass das Programm fehlerhaft ist, so muss der Spieler auf einen Computerabsturz hoffen. Nur dann kann er das fehlerhafte Programm richtigstellen. Hat er den oder die Fehler behoben, startet er wieder mit der ersten Würfelphase.

Offen lizenzierte Lehr- und Lernunterlagen für den Informatikunterricht
Maria Grandl

Wer gewinnt das Spiel?

Der Spieler, der als **erstes** 3 Programme (siehe Spielerblatt) vollständig und korrekt ausgeführt hat, ist der Gewinner des Spieles.

Offen lizenzierte Lehr- und Lernunterlagen für den Informatikunterricht
Maria Grandl