ETH zürich

ABZ AUSBILDUNGS- UND BERATUNGSZENTRUM FÜR INFORMATIKUNTERRICHT

Programmierunterricht: Von der Primarschule bis zur Maturität mit Logo und Python

Workshop am Schweizer Tag für den Informatikunterricht 2016

Tobias Kohn & Giovanni Serafini

Agenda

1. Begrüssung und Einführung (5‘)
2. Logo, Python und die Turtle-Grafik (25‘)
3. Informatik und allgemeine Bildung (15‘)
4. Programmieren in Tracks: Python & Logo (30‘)
5. Mathematik und Programmierung (10‘)
6. Zusammenfassung und Abschluss (5‘)

1. Begrüssung und Einführung

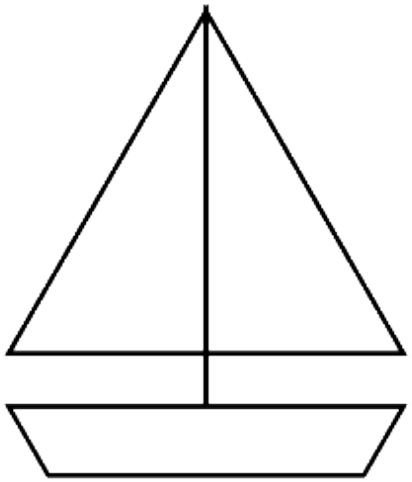
Who are we?

- Tobias Kohn
  - Doktorand in Didaktik der Informatik, ETH Zürich
  - Gymnasiallehrer für Mathematik und Informatik
  - Entwickler von TigerJython
- Giovanni Serafini
  - Dozent für Didaktik der Informatik, ETH Zürich
  - SVIA-Vorstand
  - Ausbildung der PrimaLogo-Lehrpersonen
  - Organisationskomitee Prämierung der besten Maturitätsarbeiten von SI, SVIA & ABZ.

2. Logo, Python und die Turtle-Grafik

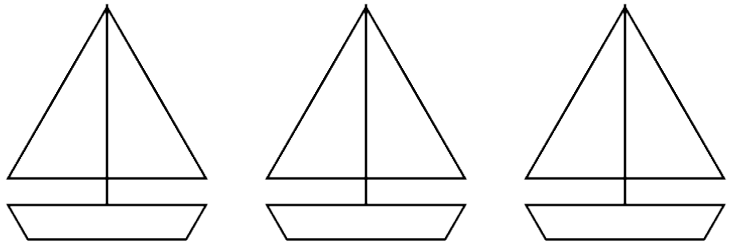
Beispiel: Segelboot (Logo & Python)

- Schreiben Sie je ein Logo- und ein Python-Programm, die das folgende Segelboot zeichnen.



Aufgabe: Eine Flotte

- Schreiben Sie ein Programm (entweder in Logo oder in Python), das eine Flotte aus drei Segelbooten zeichnet.



- 1. fachdidaktischer Schwerpunkt des Workshops: **Modularer Entwurf** im Programmierunterricht [06]

Volksschule(Lehrplan 21)

- „Die Schülerinnen und Schüler können einfache Problemstellungen analysieren, mögliche Lösungsverfahren beschreiben und in Programme umsetzen.“ [04]
- Die primäre Zielsetzung des Unterrichts besteht darin, den Kindern **eine Sprache beizubringen**, mit welcher sie die Schildkröte auf dem Bildschirm steuern können.
- Die Kindern lernen, wie sie schrittweise den **Wortschatz** der Schildkröte erweitern können, indem sie ihr **neue Wörter** und deren **Bedeutung** beibringen, welche von ihnen in Form neuer **Programme** formuliert wird. [08]

Gymnasium/Kantonsschule

- **Problemlösung** mit Programmieren als Werkzeug rückt in den Vordergrund
- Programmieren im **überfachlichen Kontext**: Insbesondere Mathematik, Physik, Philosophie, …
- Programmier-Kompetenzen **anwenden**: Maturitätsarbeit, EFI, Informatik-Olympiade

Lerneffekte für die SuS

- «Lehren ist das Entzünden von Flammen, nicht das Befüllen von Fässern.» – Also: Freude und Selbstvertrauen stärken!
- Beispiel: Abschlussprojekte von Schülern

Warum Turtle-Grafik?

- Identifikationsfigur
- Nicht-technische Sprache und Metaphern
- Direktes visuelles Feedback
- Beobachten der sequentiellen Natur von Programmen

Ziele des Workshops

1. Sie können mit eigenen Worten begründen, woraus der Beitrag der Informatik zur allgemeinen Bildung besteht.
2. Sie können die Bedeutung des Programmierunterrichts im Kontext des allgemeinbildenden Informatikunterrichts einordnen.
3. Sie sind von den didaktischen Chancen des modularen Entwurfs überzeugt. Sie gehen mit den unterschiedlichen Ausprägungen des Variablenkonzepts in Informatik und Mathematik bewusster um.
4. Sie sind in der Lage, die vorgestellten Unterrichtsunterlagen als Ausgangslage für die selbständige Vorbereitung des eigenen Unterrichts einzusetzen.

3. Informatik und allgemeine Bildung

Was ist Informatik?

- „Die Informatik ist die Wissenschaft der systematischen, automatisierten Verarbeitung von Information, der Informationsspeicherung, -verwaltung und -übertragung.“ [Gander, Hromkovic 2012]
- Die Informatik befasst sich mit der Automatisierung der intellektuellen Arbeit. [01,03]
  - Sie ist ein MINT-Fach!
  - Informatik ≠ Medienbildung
  - Informatik ≠ Anwendungen (Textverarbeitung, usw.)

Informatik und allgemeine Bildung

- In den kommenden 10 Minuten werden wir über die Bedeutung bzw. den Beitrag der Informatik zur allgemeinen Bildung nachdenken. [02]
- Dabei geht es insbesondere um drei Kernfragen:
  1. Welchen Wissenschaften kann man die Informatik zuordnen?
  2. Woraus besteht der stabile, nachhaltige Kern der Informatik?
  3. Was unterscheidet die Informatik von den anderen Wissenschaften?

Quintessenz der Informatik

- Welchen Wissenschaften kann man die Informatik zuordnen?
- Ist sie Metawissenschaft (wie Philosophie und Mathematik), Geisteswissenschaft, Naturwissenschaft oder Ingenieurwissenschaft?
- Informatik ist Meta-, Natur-, Ingenieur- und sogar Geisteswissenschaft!

Die Algorithmik

- Woraus besteht der stabile, nachhaltige Kern der Informatik?
  - „Programmieren bzw. Algorithmen zu entwickeln ist die Kunst der **kreativen Konstruktion** und **genauen Beschreibung eines Lösungswegs** für ein gegebenes Problem.“ [N. Wirth, angepasst]
  - Die Informatik hat selbstverständlich einen eigenen, unveränderlichen, stabilen und bewährten Kern: die Algorithmik.

Denken wie ein Informatiker

- Was unterscheidet die Informatik von den anderen Wissenschaften?
  - InformatikerInnen entwickeln Algorithmen
  - Sie denken „algorithmisch“.
- Wie lernt man „algorithmisch zu denken“?
  - Algorithmisch zu denken lernt man, indem man für ein gegebenes Problem eine automatisch auszuführende Lösungsanleitung konstruiert, kritisch hinterfragt und präzise beschreibt, also **nicht zuletzt**, indem man diese programmiert.
  - Informatik ohne Computer („unplugged“, siehe Tim Bell).

Last but not least

- Unser Ziel besteht darin, den Kindern und Jugendlichen **etwas fürs Leben** beizubringen.
- Was? „Computational Thinking“ (Algoritmisches Denken!)
  - Jan Cuny, Larry Snyder und Jeannette M. Wing haben die Bezeichnung “Computational Thinking” eingeführt.
  - “Computational Thinking is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent.”

4. Programmieren in Tracks: Python und Logo

[05,07,10]

5. Mathematik und Programmierung

Was ist eine Variable?

- Wie kann ich es machen, dass der Computer automatisch die Rechnung «2 + 3 = 5» ausgibt? [09]

```
x = 2 + 3
print x, '=', x
```

Was gibt das Programm aus?

```
x = 5
y = 2 * x
x = 8
print y + 1
```

Variablenkonzept

- Mathematik: Symbol als Platzhalter
  y = 2x
- Programmieren: Referenz auf einen konkreten Wert
  y = 10
- Programmieren: Variablen sind veränderlich
  x = x + 1
- 2. fachdidaktischer Schwerpunkt des Workshops:
  das Variablenkonzept.

Modell des Rechners lehren

- Wie funktioniert die Programmausführung?
- Interaktive Auseinandersetzung,
  Probleme thematisieren
- Richtige Abstraktionsebene:
  Logo und Python, nicht Hardware

6. Zusammenfassung und Abschluss

Zusammenfassung und Abschluss

- Computational Thinking „lernt man fürs Leben“:
  - Programmierunterricht soll „im Kleinen“ mit der Betrachtung kleiner algorithmischer Probleme anfangen.
  - Weniger ist mehr: wenige, klare Befehle; komplexere Programme sollen **modular** entwickelt werden.
  - Programmierunterricht ist kognitiv anspruchsvoll und kann dennoch kindergerecht mit grossem Lernerfolg durchgeführt werden.
- Programmieren ist mehr als Syntax:
  - Syntax und Semantik der Programmiersprache
  - Sprachelemente anwenden und kombinieren
  - Modell der Programmausführung
  - Abstraktionsprozesse

Literaturempfehlungen

[01] Homo Informaticus

[02] Informatik und allgemeine Bildung

[03] Das Zeitalter der Informatik

[04] Lehrplan21: Medien und Informatik

[05] Kluger Spass: Programmieren in der Primarschule

[06] Konzepte für den Informatikunterricht an Primarschulen und auf der Sekundarstufe I

[07] Teaching Programming at Primary Schools

[08] Programmierunterricht für Kinder und deren Lehrpersonen

[09] Examples of Algorithmic Thinking in Programming Education

[10] Computing im Physikunterricht