

Debugging mit Plan!

Endlich Schluss mit der Turnschuhdidaktik

Dr. Tilman Michaeli
25.03.21



 tilman.michaeli@fu-berlin.de

 @TilmanMichaeli

 <https://michae.li>

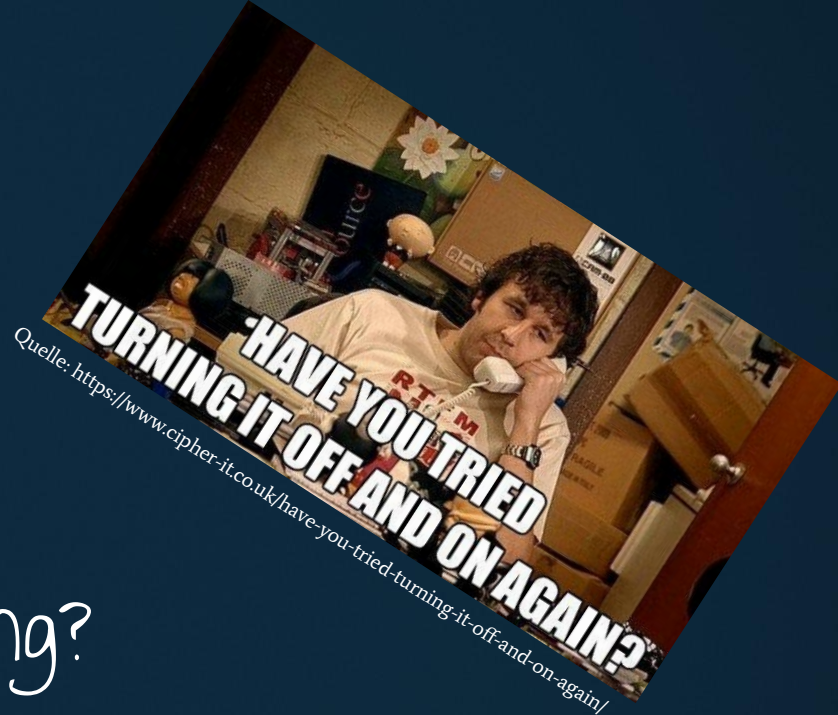
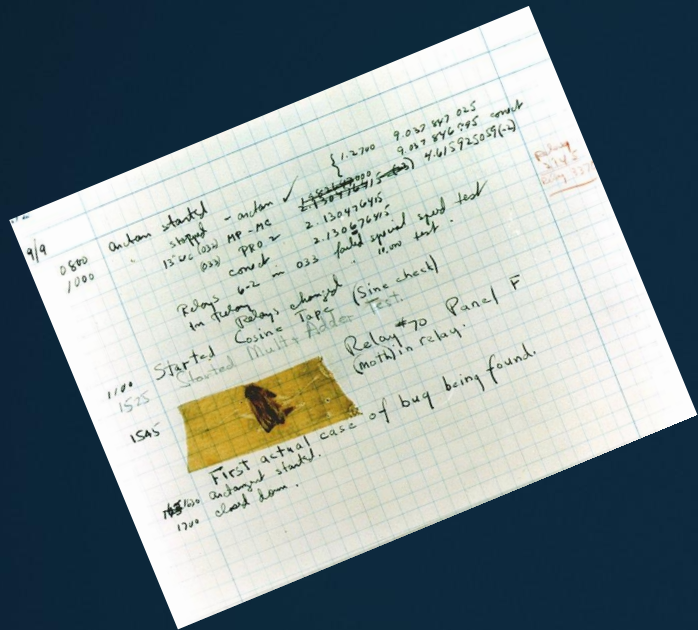
Debugging?



Finden



Beheben



Debugging?

"If debugging is the process of removing software bugs, then programming must be the process of putting them in."

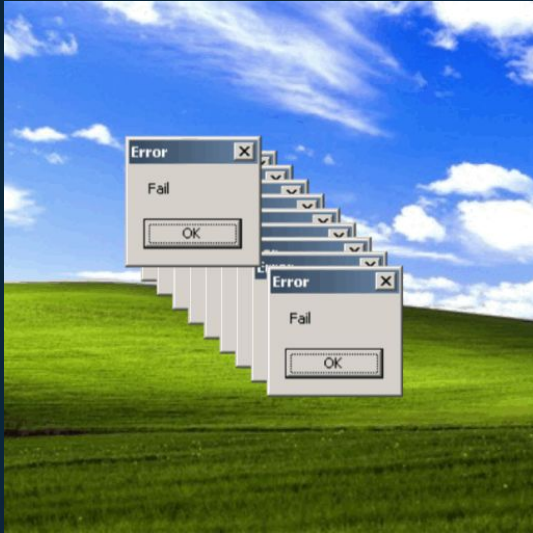
(E. W. Dijkstra)

Die Profis?

As soon as we started programming, we found to our surprise that it wasn't as easy to get programs right as we had thought. Debugging had to be discovered. I can remember the exact instant when I realized that a large part of my life from then on was going to be spent in finding mistakes in my own programs.

Maurice Wilkes, 1949

Die Schülerinnen und Schüler?



Quelle: <http://killer-angel123.tumblr.com/post/125825232636>



Quelle: <https://giphy.com/gifs/movie-agree-the-sandlot-3oKHWfcpURdVIsxZS>

Die Lehrkräfte?

... verbringen auch 40-60% mit dem Debuggen (fremder) Programme?

Stellt Euch kurz vor und beantwortet dabei die folgenden zwei Fragen:

1. Was erwartet ihr Euch vom Workshop?
2. Wie habt ihr selber Debuggen gelernt?





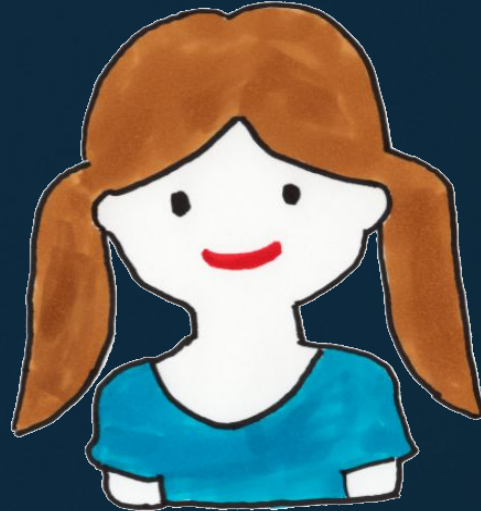
Schlüsselproblem des
Informatikunterrichts.



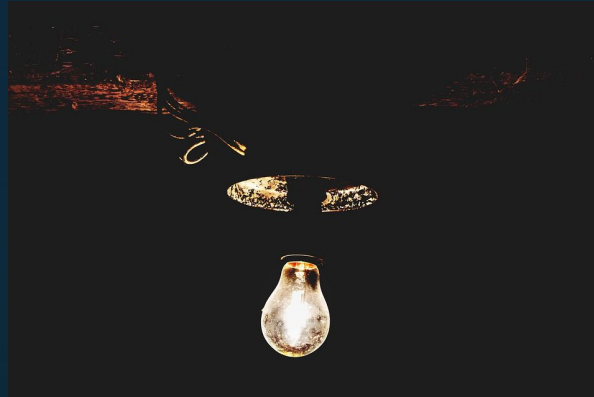
Das Internet ist
kaputt!!!



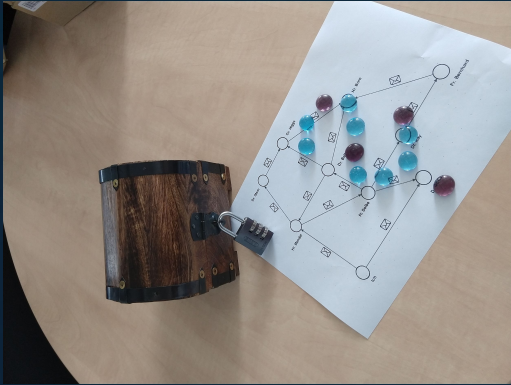
Hm...
Ist mein LAN-Kabel
verbunden?
Sind auch andere
Geräte im Netzwerk
betroffen?



Troubleshooting und Debugging?

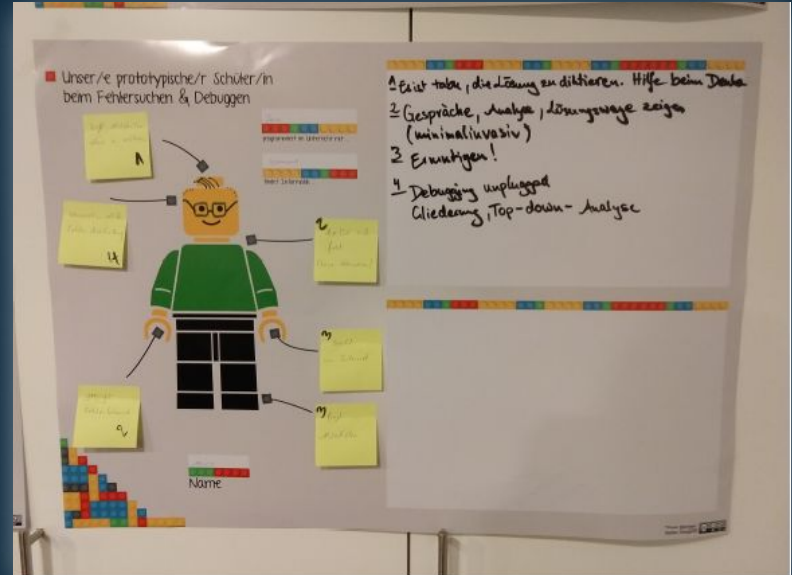


Among Scarabs and Bugs...



Zurück zur Programmierung

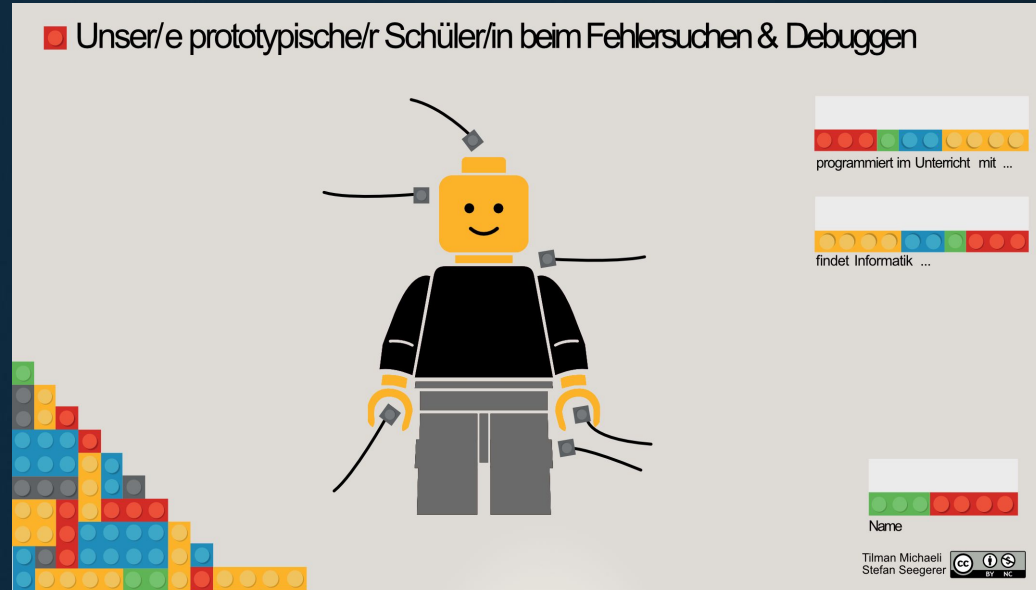
Die "Problemstellung": Unsere Schülerinnen und Schüler?



Personas I

Beschreibt eure/n “prototypische/n” Schüler/in beim Suchen und Beheben von Fehlern im Unterricht in GoogleDocs:

- Wie geht er/sie vor?
- Womit hat er/sie Probleme?



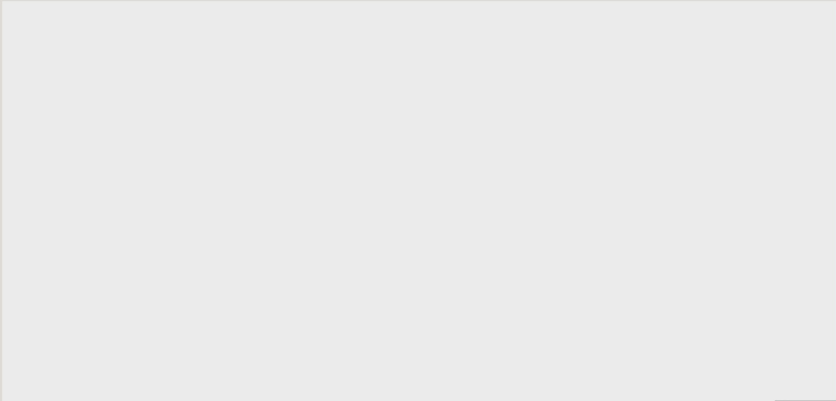
Personas II

Gebt nun “eure” Schülerin bzw. “euren” Schüler weiter an die nächste Gruppe.

Notiert im Feld, was ihr **bisher** macht, um der Schülerin bzw. dem Schüler vor Euch zu helfen!

- Welche Hilfestellungen gebe ich?
- Wie adressiere ich das in meinem Unterricht?

■ So unterstütze wir sie/ihn in unserem Unterricht



Tilman Michaeli
Stefan Seegerer



Der Debugging“novize“ in der Wissenschaft

- Ähnliche Strategien wie “Profis”, aber ineffizient umgesetzt
- Viel trial-and-error
- Angewohnheit, neue Fehler einzubauen
- Probleme im Umgang mit Fehlermeldungen
- Frustration

Programmierfähigkeiten
unterscheiden sich von
Debuggingfähigkeiten.

Debuggingfähigkeiten
entwickeln sich nicht
"von selbst".

Und was kann ich da
vermitteln?

4 Relevante Debuggingfähigkeiten

Systemverständnis (Wie ist mein Code aufgebaut?)



Anwenden eines systematischen
Debuggingprozesses



Anwenden von
Debuggingstrategien



Verwendung von
Werkzeugen

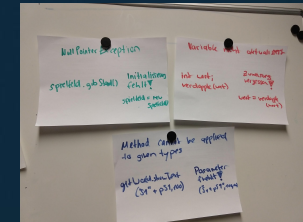
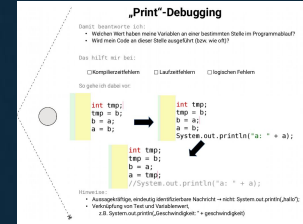
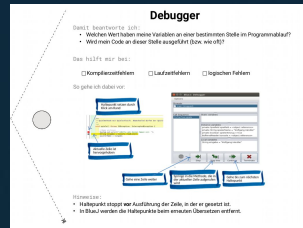


Anwendung von Heuristiken
und Mustern für häufige Bugs

Allgemeine Programmierfähigkeiten

Wie kann ich das jeweils unterrichten und fördern?

1
2
3



1 2 3 Systematisches Vorgehen

1 2 3 Systematisches Vorgehen

Wie gehen Profis beim Debuggen (implizit) vor?

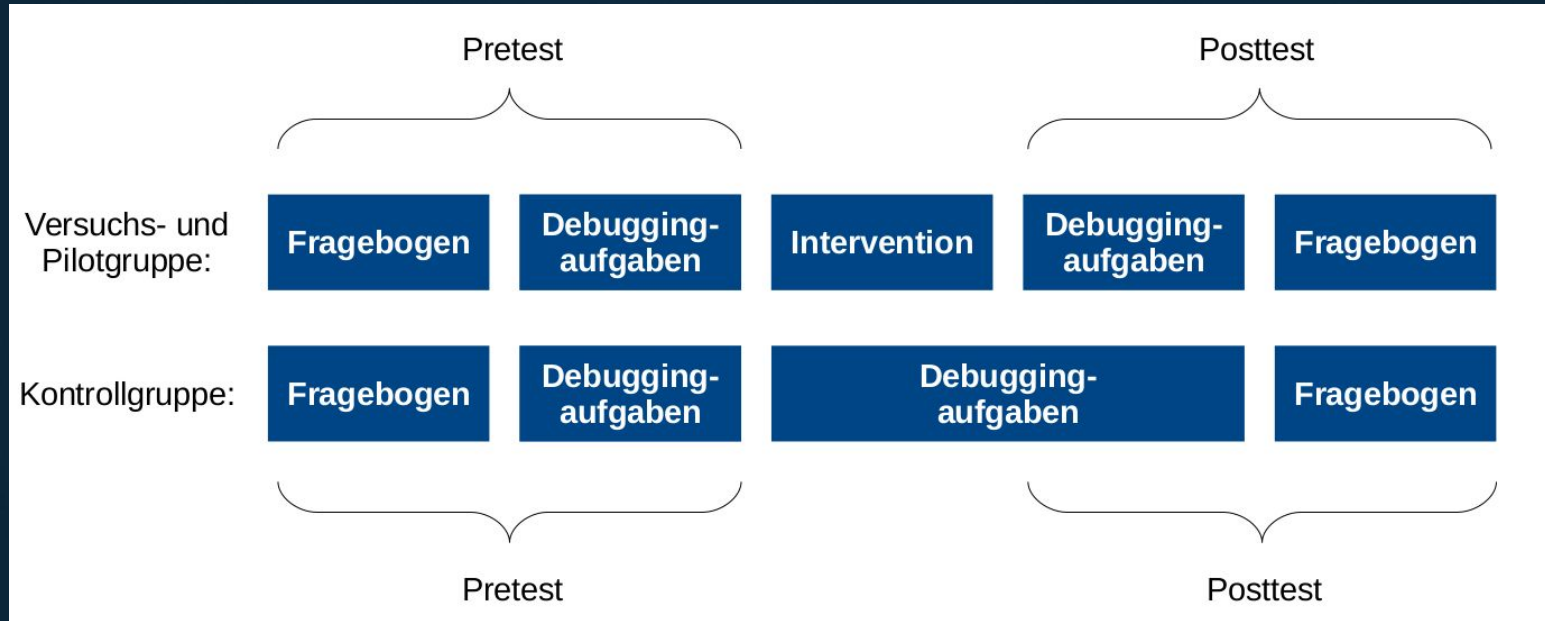
=> "wissenschaftliche Methode"

1. Testen und Fehler reproduzieren.
2. Falls nötig, Überblick über das Programm gewinnen.
3. Eine Hypothese formulieren.
4. Hypothese experimentell überprüfen.
5. Falls nötig, Hypothese verfeinern/anpassen.
6. Den Fehler beheben.
7. Das Programm erneut testen.

1 2 3 Systematisches Vorgehen für Schülerinnen und Schüler?

- didaktisch reduzierte
“Wissenschaftliche Methode”
- Unterscheidung nach Fehlertypen
- Betont Undo
- zielt auf Selbstständigkeit





Unsere Erfahrungen und Ergebnisse.

Empirische Untersuchung mit Versuchs- und Kontrollgruppe:

1. In beiden Gruppen stiegen die Selbstwirksamkeitserwartungen für das Debuggen, aber nur in der Versuchsgruppe signifikant
2. Die Versuchsgruppe hat nach der Intervention signifikant mehr Fehler gefunden, als die Kontrollgruppe.

=> “Es funktioniert” 

Unsere Erfahrungen und Ergebnisse.

Empirische Untersuchung mit Versuchs- und Kontrollgruppe:

1. In beiden Gruppen stiegen die Selbstwirksamkeitserwartungen für das Debuggen, aber nur in der Versuchsgruppe signifikant
2. Die Versuchsgruppe hat nach der Intervention signifikant mehr Fehler gefunden, als die Kontrollgruppe.

=> “Es funktioniert” 

- “Wo ist denn da ein Fehler, es kompiliert doch?”
- “Das mache ich doch eh schon”

Systematisches Vorgehen für Schülerinnen und Schüler!

Intervention:

- SuS bearbeiten Debuggingaufgaben.
- SuS klassifizieren verschiedene Fehler, denen sie begegnet sind.
- Einführung des systematischen Vorgehens.
- Einüben anhand von weiteren Debuggingaufgaben.



Strategien

Debuggingstrategien



Logging & Tracing

- Tracetabelle
- **Printf**
- **Debugger**



Isolieren

- Auskommentieren
- Slicing



Abweichungen erkennen

- Testen
- Ausgabe vergleichen
- Assertions



Hilfe suchen

- Websuche
- Rubberduck-Debugging



Informationen sammeln

- Spezifikation
- API



Debuggingstrategien

- Aktivität, in der Verwendung von konkreter Debuggingstrategie eingeübt wird
- Erstellen von Skill Cards
- Einfördern der Anwendung dieser Strategien

„Print“-Debugging

Damit beantworte ich:

- Welchen Wert haben meine Variablen an einer bestimmten Stelle im Programmablauf?
- Wird mein Code an dieser Stelle ausgeführt (bzw. wie oft)?

Das hilft mir bei:

☐ Kompilierzeitfehlern ☐ Laufzeitfehlern ☐ logischen Fehlern

So gehe ich dabei vor:

```
int tmp;  
tmp = b;  
b = a;  
a = b;
```

```
int tmp;  
tmp = b;  
b = a;  
a = b;  
System.out.println("a: " + a);
```

```
int tmp;  
tmp = b;  
b = a;  
a = tmp;  
//System.out.println("a: " + a);
```

Hinweise:

- Aussagekräftige, eindeutig identifizierbare Nachricht → nicht: `System.out.println("hallo");`
- Verknüpfung von Text und Variablenwert, z.B. `System.out.println("Geschwindigkeit: " + geschwindigkeit)`

Unsere Erfahrungen und Ergebnisse.

- Insbesondere Print/Debugger sind Vorgehensweisen, die erst entwickelt werden müssen.
- Enge Führung in den Aufgaben wichtig.
- Einfordern der Verwendung der Strategien!



Werkzeuge

Debugging-Werkzeuge

Professionell:

- Back-in-time-Debugger
- AFL: Automatic Fault Location
- “neu”: KI
- automatic program slicing

Debugging-Werkzeuge

In der Schule?

- Debugger
- IDE-Feedback

Debugger

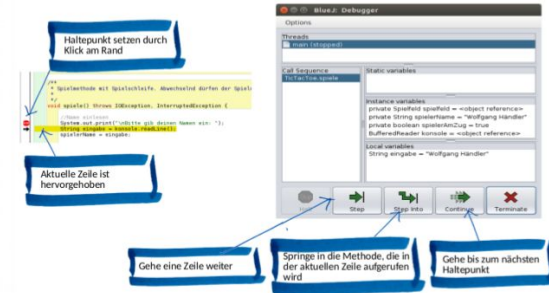
Damit beantworte ich:

- Welchen Wert haben meine Variablen an einer bestimmten Stelle im Programmablauf?
- Wird mein Code an dieser Stelle ausgeführt (bzw. wie oft)?

Das hilft mir bei:

- ☐ Kompilierzeitfehlern ☐ Laufzeitfehlern ☐ logischen Fehlern

So gehe ich dabei vor:



Hinweise:

- Haltepunkt stoppt **vor** Ausführung der Zeile, in der er gesetzt ist.
- In BlueJ werden die Haltepunkte beim erneuten Übersetzen entfernt.



Heuristiken und Muster

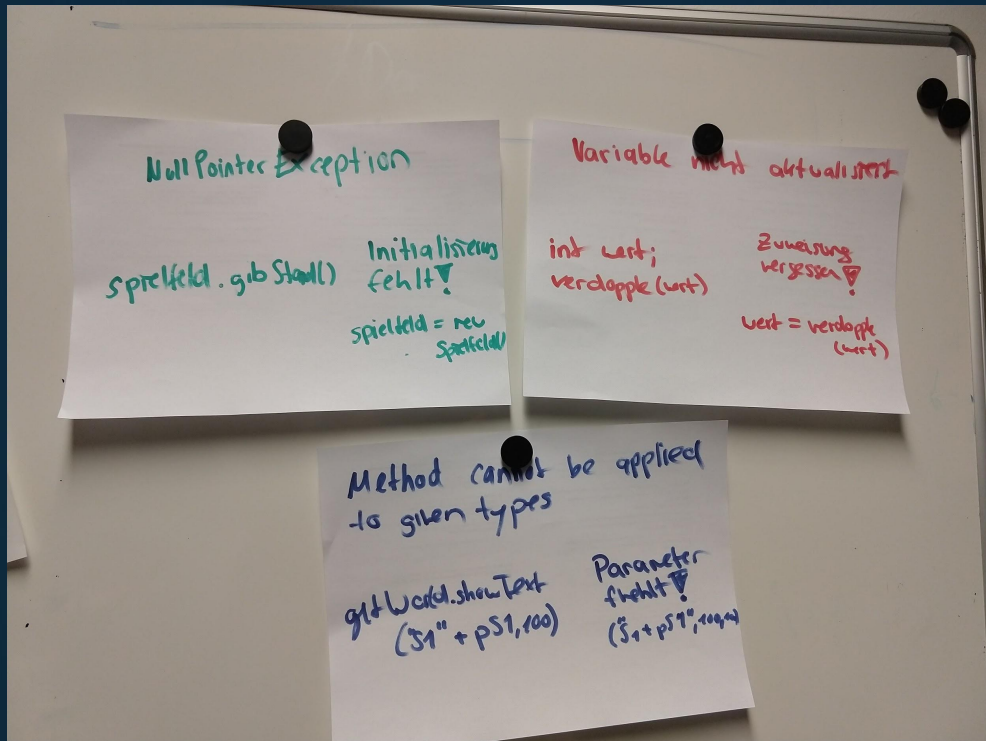
Heuristiken und Muster

Profis:

- Bauen diese Heuristiken und Muster auf
- Debugging-Logs/Diaries

Heuristiken und Muster

Debugginglogs



 Stackoverflow und Co.

Debuggingaufgaben?

Debuggingaufgaben?

Traditionelle Debuggingaufgaben zum Üben von Debugging:

viel fremder Code, in dem Fehler versteckt sind.

Problem:

-  Codeverständnis vs. Debuggen eigener Programme
-  Üben von Testfähigkeiten vs. Debuggingfähigkeiten
-  Debuggingvorgehen für eigenen Code ist anders als für fremden

Lösung: aufeinander aufbauende **Prototypen**

Prototypenbasierte Debuggingaufgaben

Prototyp 1: Ball

Ball
- deltaX: int - deltaY: int
+ act(): void + ueberpruefeWand(): void + ueberpruefeTor(): void

User Story: Der Ball bewegt sich und prallt von allen Wänden ab.

Prototyp 2: Punkte

Ball
- deltaX: int - deltaY: int - mensch: Zaehler - computer: Zaehler
+ act(): void + ueberpruefeWand(): void + ueberpruefeTor(): void + zuruecksetzen(): void

Counter
+ Counter() + add(score: int): void + getValue(): int + setValue(int newValue): void

User Story: Wenn der Ball ein "Tor" trifft, werden die Punkte des jeweils anderen Spielers hochgezählt und der Ball wird in die Mitte des Spielfeldes zurückgesetzt.

Ausprobieren der Materialien

Leitfragen:

- Wann würdet ihr eine solche Aktivität bei euch im Unterricht einsetzen?
- Wie wird eure “Lego”-Schülerin bzw. euer “Lego”-Schüler damit unterstützt?

Ziele

- Selbstständigkeit fordern und fördern.
- das Formulieren von Hypothesen einüben.
- insbesondere die Bedürfnisse „schwächerer“ Schülerinnen und Schüler adressieren.
- Debuggingstrategien explizit vermitteln.
- den aktiven Aufbau von Erfahrung unterstützen.
- unterschiedliches Vorgehensweisen für verschiedene Fehlerarten vermitteln.
- tatsächliche (relevante) Debuggingfähigkeiten fördern.

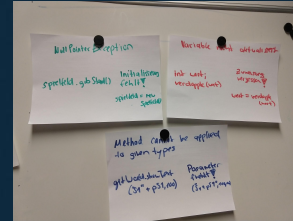
Erfahrungen von Lehrkräften

- Bedeutung des Themas
- über die Programmierung hinaus
- über die Informatik hinaus
- Öffnung von Aufgaben

Kann Hilfslosigkeit erlernt
werden?

Debuggen kann vermittelt
werden.

1	_____
2	_____
3	_____





Ich bin tatsächlich der Meinung, dass es ja nicht ganz fair ist, dem Schüler immer vorzuwerfen, er findet die Fehler nicht, wenn ich ihm doch nicht mal gezeigt habe, was ich normalerweise nutze. Das empfinde ich jetzt als unfair. Deshalb ist eben das Minimum, was ich ihm zeigen muss, das, was ich üblicherweise selber benutzte.

Werbeblock :)

DIE WELT DER KI ENTDECKEN



Stefan Seegerer
Tilman Michaeli

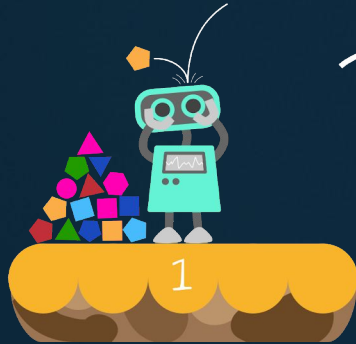


UNSERE REISE

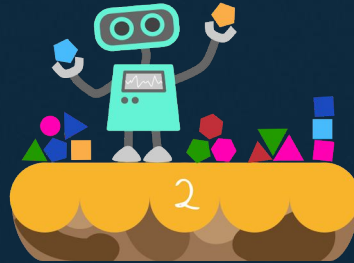


Künstliche Intelligenz in der Schule

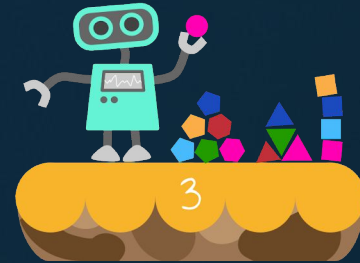
Unbeschriftete Eingaben
erhalten



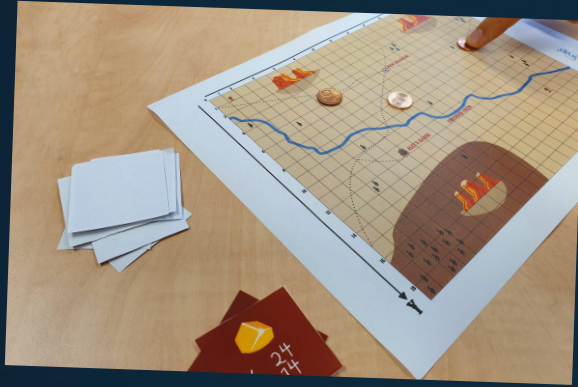
Ähnlichkeiten in den
Eingaben erkennen
und Muster finden



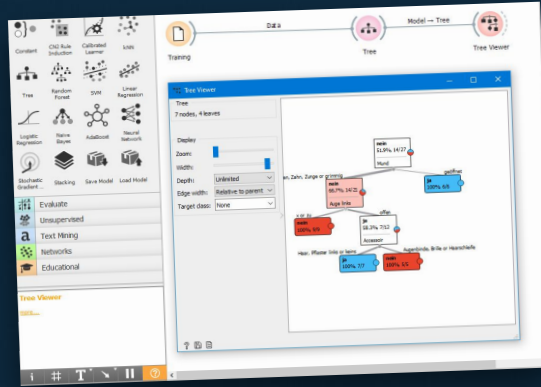
In der Eingabe
Gruppen und Ausreißer
identifizieren



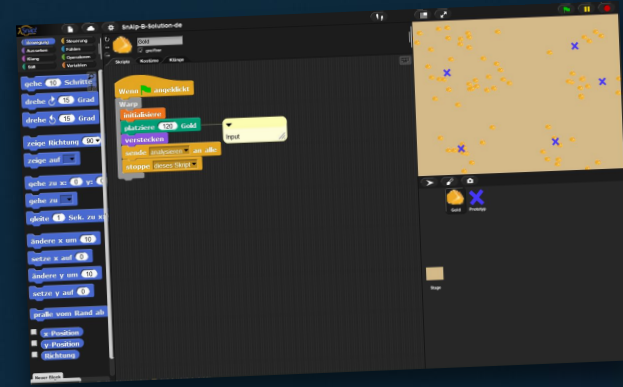
Künstliche Intelligenz in der Schule



AI Unplugged



Mit Hilfe von maschinellem Lernen
Daten auswerten



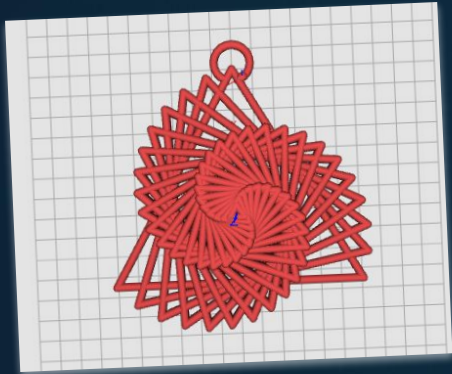
Maschinelles Lernen selber
programmieren

Agile Projektarbeit macht Schule!

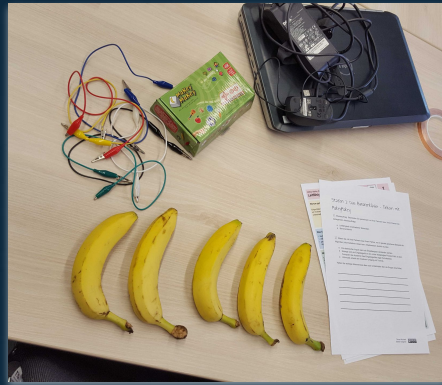


*Methoden für den Projektunterricht in der Informatik
und darüber hinaus*

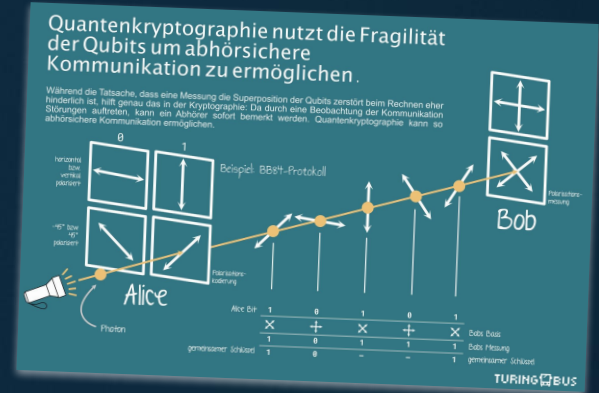
Für Informatik begeistern!



Informatik trifft Design



Die digitale Welt gestalten



Reise in die Quantenzeit!

Lust auf mehr?
computingeducation.de

Vielen Dank für die
Aufmerksamkeit