



# Debugging im Unterricht

Heike Hennig & Tilman Michaeli

[tilman.michaeli@tum.de](mailto:tilman.michaeli@tum.de)

07.07.2023



# Heute

- Praktische Fortführung des Vortrags
- Zwei Bestandteile:
  1. Arbeit mit Videovignetten
  2. Ausprobieren von konkretem Unterrichtsmaterial zur Förderung von Debuggingfähigkeiten im Unterricht



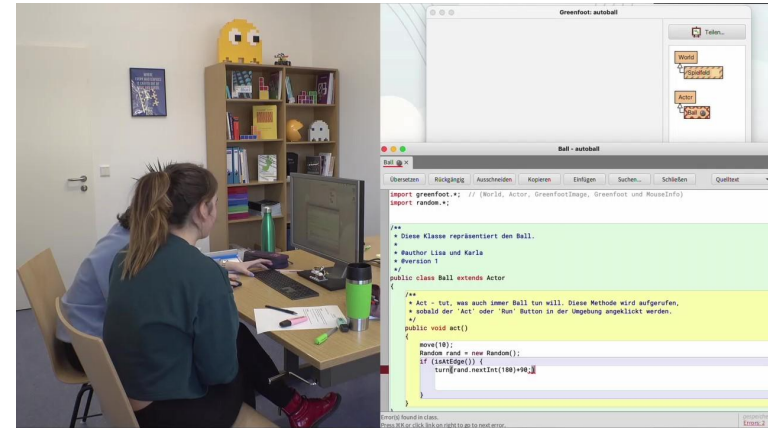
# Videovignetten

Warum verwenden wir Videovignetten im Workshop?

- Für Diagnose- und Interventionsprozess beim Debugging sensibilisieren
- Zeigen wie mit Hilfe von Videovignetten dieser Prozess gefordert werden kann

# Videovignetten

- Videovignetten zeigen typische Debugging-Situationen im Informatikunterricht
- Vorgehen:
  - Vignette gemeinsam ansehen
  - Zeit für Notizen
  - Vignette ein weiteres Mal ansehen
  - Diskussion in Gruppen
  - Sammeln im Plenum





# Videovignetten

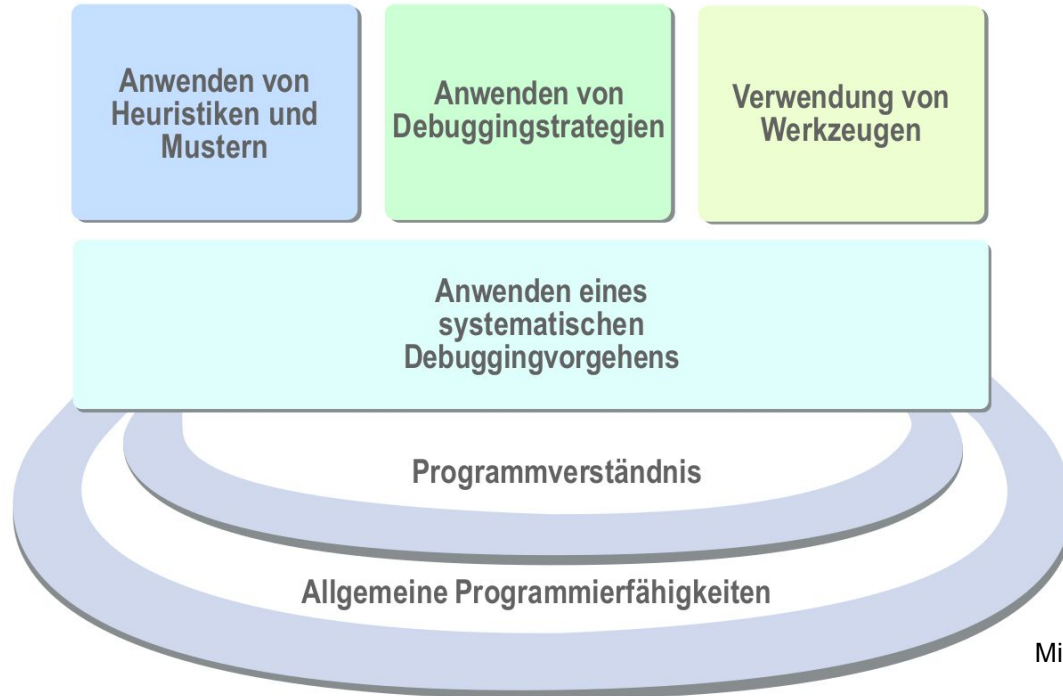
- Machen Sie sich bitte Notizen zu folgenden Punkten
  - Welche Aspekte des Verhaltens fallen Ihnen bei den Schülerinnen auf?
  - Warum sind die Schülerinnen nicht in der Lage das Problem selbstständig zu lösen?
  - Wie würden Sie als Lehrkraft in der Situation reagieren?
  - Analysieren Sie die gezeigte Reaktion der Lehrkraft.



**Kurze Pause**



# Debuggingfähigkeiten für Programmieranfänger



Michaeli, 2021





## Anwenden eines systematischen Debuggingvorgehens

# Systematisches Vorgehen für Schülerinnen und Schüler?

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





# 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.



## Anwenden von Debuggingstrategien



# Debuggingstrategien

- Auskommentieren/Slicing
- Logging und Tracing (z.B. print-debugging, Tracetabelle)
- Assertions
- Testen
- Informationen sammeln (z.B. API)
- Hilfe suchen (z.B. StackOverflow, “Rubberduck”)

## ... im Unterricht

- Aktivität, in der Verwendung von konkreter Debuggingstrategie eingeübt wird
- Erstellen von Skill Cards
- Einfordern 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!



## Verwendung von Werkzeugen

# ... im Unterricht

- 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:

The image shows a screenshot of the BlueJ IDE with a debugger window open. The debugger window displays the 'Options' dialog, 'Threads' list, 'Call Sequence', and 'Static variables' panels. The 'Static variables' panel shows instance variables: `private final Object[] = null; reference=`, `private String spaterAmZug = "aufgang spulder"`, and `private boolean konsole = null; reference=`. Local variables include `String eingabe = "aufgang händler"`. The debugger controls at the bottom include buttons for 'Step', 'Step Info', 'Continue', and 'Terminate'. Annotations in blue boxes point to these controls: 'Gehe eine Zeile weiter' (Step), 'Springe in die Methode, die in der aktuellen Zeile aufgerufen wird' (Step Info), and 'Gehe bis zum nächsten Haltepunkt' (Continue). On the left, a code snippet is shown with a red square breakpoint icon on a line. Annotations in blue boxes point to this icon: 'Haltepunkt setzen durch Klick am Rand' and 'Aktuelle Zeile ist hervorgehoben'.

Hinweise:

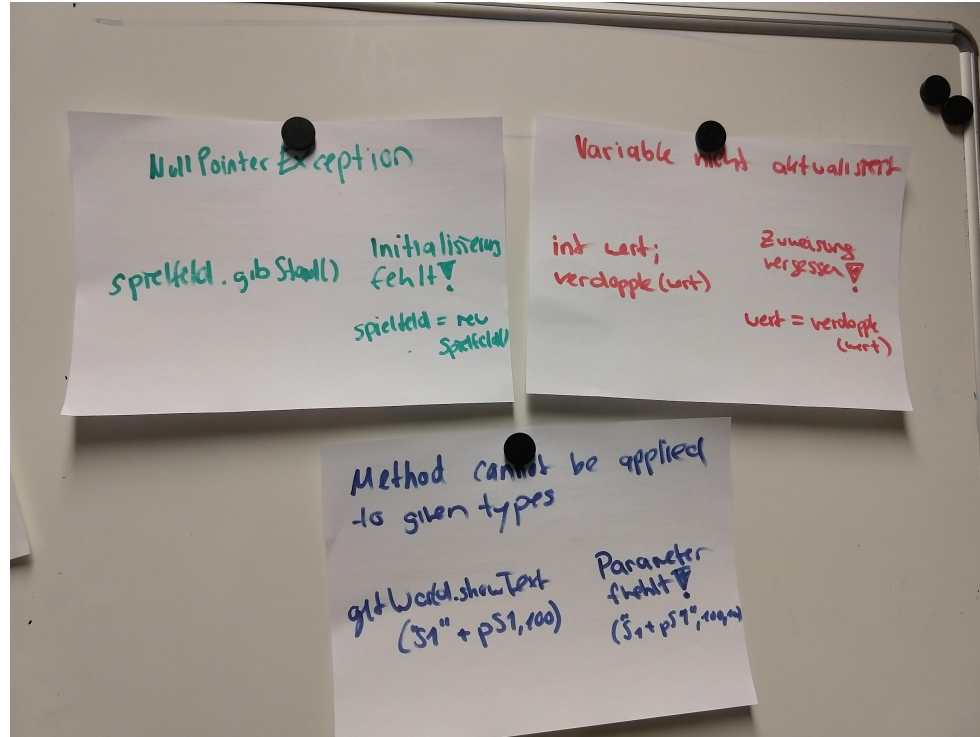
- Haltepunkt stoppt vor Ausführung der Zeile, in der er gesetzt ist.
- In BlueJ werden die Haltepunkte beim erneuten Übersetzen entfernt.



## Anwenden von Heuristiken und Mustern

# Heuristiken und Muster

## Debugginglogs





# Debuggingaufgaben?



# Debuggingaufgaben?

Traditionelle Debuggingaufgaben zum Üben von Debugging: viel fremder Code, in dem Fehler versteckt sind.

## **Problem:**

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

**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

<https://computingeducation.de/proj-debugging/>

AB Systematisches Vorgehen



AB Strategien (BlueJ/Plain Java)

- Debugger
- Print-Debugging





# 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.



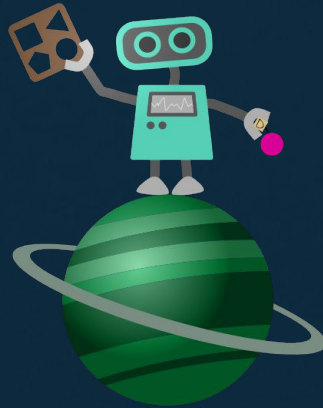
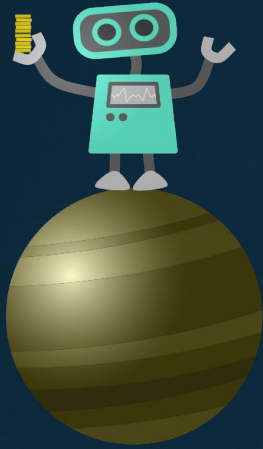


*Das ist eben die Neuerung in meinem Unterricht oder auch in meinem Anspruch oder meinem Herangehen. Das war ja in der Vergangenheit nicht so der Fokus. Da ging es ja darum, das Projekt grundsätzlich so klein zu halten, dass ich eigentlich davon ausgehe, dass das fehlerfrei stattfindet.*



**Werbeblock**

# DIE WELT DER KI ENTDECKEN



über 15.000  
Teilnehmer!



MOOC, kostenlos und OER, alle Informationen  
unter <https://computingeducation.de/proj-it2school/>

Stefan Seegerer  
Tilman Michaeli

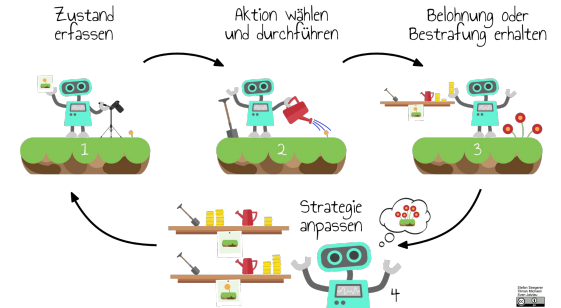
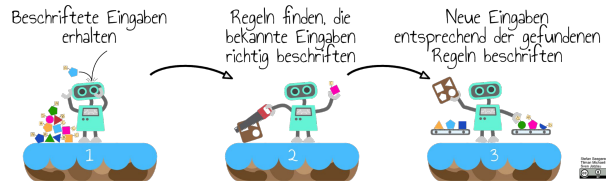
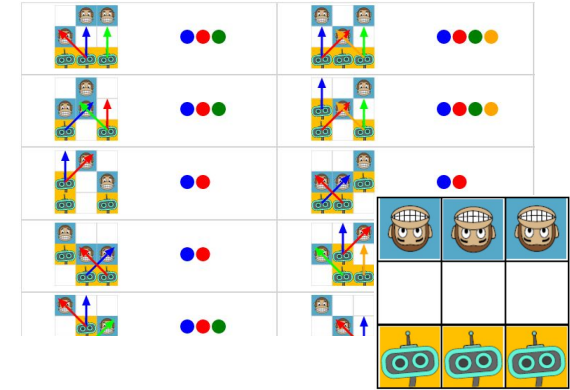


# IT2School Unterrichtsmaterial



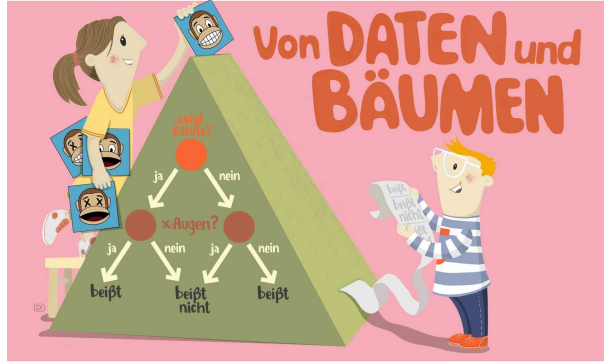


# Schlag den Roboter - Spielerisch KI entdecken



OER-Materialien unter:  
<https://computingeducation.de/proj-it2school/>

# Von Daten und Bäumen - Daten mit KI selbst auswerten



OER-Materialien unter:  
<https://computingeducation.de/proj-it2school/>

**Arbeitsmaterial KI-B4.1.2b**

- Entscheide mithilfe deines Entscheidungsbaumes, ob die folgenden Affen beißen (B) oder nicht beißen (NB) und trage deine Vorhersage in die freien Felder ein!
- Vergleiche deine Lösung mit der Auflösung und markiere, ob deine Vorhersage richtig oder falsch war!

						Deine Vorhersage: Beißt oder nicht richtig?
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	War deine Vorhersage richtig oder falsch?
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

						Deine Vorhersage: Beißt oder nicht richtig?
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	War deine Vorhersage richtig oder falsch?
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

**Konfusionsmatrix**

Jetzt hast du die Testdaten klassifiziert und ihre tatsächliche Zuordnung ist bekannt. Die Ergebnisse trägt du summarisch in die folgende Tabelle (die Konfusionsmatrix) ein.

- Fülle die Felder der Konfusionsmatrix! Beispiel: Das Feld links oben gibt an, für wie viele Affchen du richtig die Kategorie beißt vorhergesagt hast (z. B. 2/3).

	Vorhergesagt: Affchen beißt	Vorhergesagt: Affchen beißt nicht
Tatsächlich: Affchen beißt		
Tatsächlich: Affchen beißt nicht		

- Berechne die folgenden Verhältnisse und gib das Ergebnis als Prozentsatz an!
 
$$\frac{\text{\#korrekt. vorhergesagt}}{\text{\#gesagte. vorhergesagt}} = \frac{\text{\#beißt richtig vorhergesagt}}{\text{\#gesagte. beißen}} = \frac{\text{\#beißt richtig vorhergesagt}}{\text{\#beißt vorhergesagt}}$$

Hier berechnen wir gemeinsam ihren Verdreh, bis wir auch die Markierungen.

Deine Ergebnisse:

Modul KI-B4 - Von Daten und Bäumen zuletzt aktualisiert am 25.06.21 Seite 2 von 2

**Arbeitsmaterial KI-B4.2.2b**

d) Mit dem Tree-Widget kannst du den Baum aber noch nicht ansehen, sondern nur Einstellungen fürs Training festlegen.

Um den gelernten Baum anzuzeigen, ziehe ein Tree Viewer-Widget aus dem Bereich Visualize auf die Leinwand und verbinde es mit dem Tree-Widget. Mit einem Doppelklick kannst du dir den Baum ansehen (siehe auch Hilfskarte 1).

e) Im Folgenden gilt es, den gelernten Entscheidungsbaum zu testen. Dazu stehen Testdaten bereit, auf die du nun den Baum anwendest.

Ziehe dazu ein Predictions-Widget aus dem Bereich Evaluate auf die Leinwand, das das zur Verfügung gestellte Modell auf die zu Verfügung gestellten Daten anwendet. Du hast bereits beides. Verbinde das Widget mit den Testdaten und dem Entscheidungsbaum. Ein Doppelklick zeigt an, wie das Modell die Daten klassifiziert hat (siehe auch Hilfskarte 2).

f) Schluss untersuchst du, wie gut das vom Computer erstellte Modell ist. Bei dieser Einschätzung hilft dir die Konfusionsmatrix.

Ziehe das Confusion Matrix-Widget aus dem Bereich Evaluate auf die Leinwand (Hilfskarte 3) und verbinde es mit dem Predictions-Widget. Ein Doppelklick auf das Confusion Matrix-Widget zeigt die Konfusionsmatrix an.

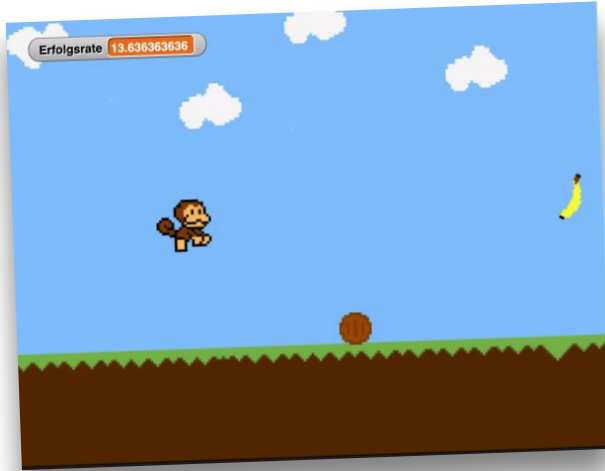
Berechne Genauigkeit, Trefferquote und Präzision deiner Vorhersage gemäß der Formel aus der letzten Stunde. Die Felder der hier angezeigten Konfusionsmatrix entsprechen auch den Feldern aus deinem Arbeitsmaterial von letzter Stunde.

Modul KI-B4 - Von Daten und Bäumen zuletzt aktualisiert am 25.06.21 Seite 2 von 4

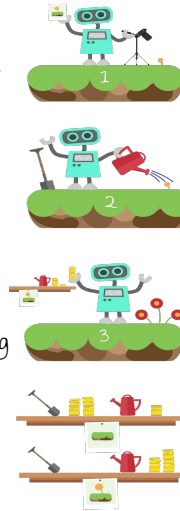




# Die Bananenjagd - Computer selbst lernen lassen

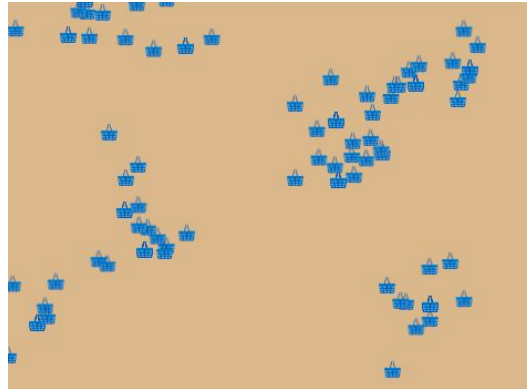
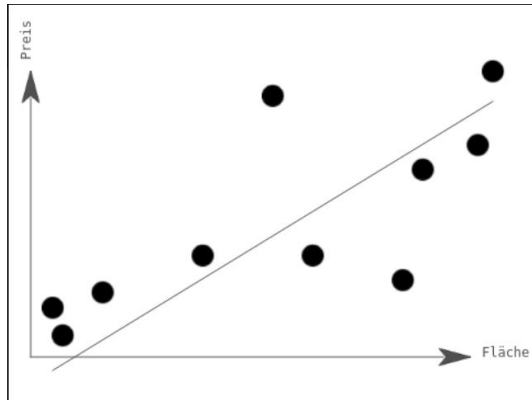


1. Zustand erfassen
2. Aktion wählen und durchführen
3. Belohnung oder Bestrafung erhalten
4. Strategie anpassen

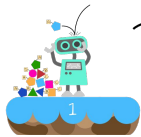


```
fortlaufend
setze Zustand auf x-Position von Fass / 10 gerundet
setze beste Aktion auf
  bestes Verhalten für Zustand: Zustand Modell: Modell
führe beste Aktion aus
setze Belohnung auf if berühre Fass ? then -10 else 1
Aktualisiere Modell: Modell alter Zustand: Zustand neuer Zustand
  x-Position von Fass / 10 gerundet Belohnung: Belohnung
gezeigtes Verhalten: beste Aktion
```

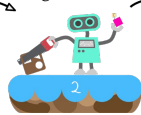
# Die Bananenjagd - Computer selbst lernen lassen



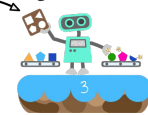
Beschriftete Eingaben erhalten



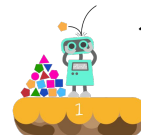
Regeln finden, die bekannte Eingaben richtig beschriften



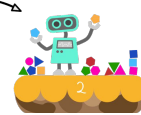
Neue Eingaben entsprechend der gefundenen Regeln beschriften



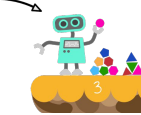
Unbeschriftete Eingaben erhalten



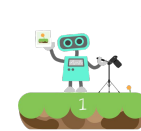
Ähnlichkeiten in den Eingaben erkennen und Muster finden



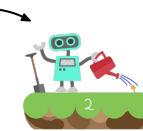
In der Eingabe Gruppen und Ausreißer identifizieren



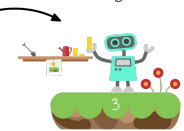
Zustand erfassen



Aktion wählen und durchführen



Belohnung oder Bestrafung erhalten



Strategie anpassen



OER-Materialien unter:

<https://computingeducation.de/proj-it2school/>



# Mein persönlicher Assistent



OER-Materialien unter:  
<https://computingeducation.de/proj-it2school/>

# Reise in die Quantenzeit

## QUANTEN- INFORMATIK

### REISE IN DIE QUANTENZEIT

Willkommen in der Welt im Jahr 2038! Es hat sich einiges geändert: Die Verschlüsselungssysteme, die wir bisher im Internet verwendet haben, waren nicht mehr sicher und mussten aktualisiert werden. Jetzt können persönliche Daten über Netzwerke wirklich abhörbarer übertragen werden, dank der Quanteninformatik hat aber auch die Arzneimittelentwicklung und -forschung ganz neue Dimensionen angenommen. Und die Suche nach Informationen dauert jetzt wirklich nur noch einen Wimpernschlag.

So oder so ähnlich könnte eine Welt mit Quantencomputern zukünftig aussehen. Aber noch ist diese Zukunft nicht geschrieben. Du kannst sie selbst aktiv mitgestalten und die Quanteninformatik könnte dabei ein wichtiger Baustein sein. Dieses Heft bietet dir eine Einführung in die Welt der Quanteninformatik.

Das Heft beinhaltet einerseits Erklärtexte und andererseits Aktivitäten, mit denen du selbst Konzepte der Quanteninformatik entdecken kannst. Es ist so gestaltet, dass du die Welt der Quanteninformatik alleine, gemeinsam mit einem Freund, einer Freundin oder deiner Lehrkraft erforschen kannst. Es ist dein Heft. Das heißt, du kannst deine Notizen direkt in Heft machen.

An vielen Stellen ergeben sich Verbindungen zu klassischen Themen der Informatik. Du brauchst aber keine Angst haben: Wir werden alle erforderlichen Begriffe und Grundlagen anschaulich erläutern. Auch wenn also Begriffe wie Bits, Schaltung oder Simulation noch neu für dich sind, wirst du mit diesem Material sehr viel Spaß haben!

Los geht's!

PS: An manchen Stellen in dieser Broschüre haben wir Anspielungen auf eine mögliche Zukunft gemacht. Alle Fakten und Anregungen sind frei erfunden. Wir stehen in keiner Verbindung mit den genannten Firmen oder Marken.



In Jahr 2021 haben Quantencomputer so aus (CC-BY-SA ZDF).

## LEICHTE & SCHWERE PROBLEME

Herkömmliche Computer (wie dein Notebook oder dein Smartphone, aber auch raumgroße Supercomputer in den Rechenzentren dieser Welt) erzielen beeindruckende Ergebnisse. Es gibt Programme, die die grafische Ausgabe von fiktionalistischen Computerspielen oder Animationsfilmen berechnen oder welche, die erkennen, was auf einem Bild abgebildet ist. Andere Programme führen komplexe, wissenschaftliche Berechnungen durch und treiben so den technischen Fortschritt voran.

Auch wenn es manchmal so wirkt, als könnten herkömmliche Computer jede Aufgabe mit Leichtigkeit lösen, sind auch für Computer manche Probleme schwieriger als andere und einige sogar nicht mal in akzeptabler Zeit zu lösen.

Probleme es selbst aus.

Auf den folgenden Seiten findest du unterschiedlich schwere Probleme.

Nimm dir jeweils 5 Minuten Zeit und versuche zunächst selbst eine Lösung zu finden.

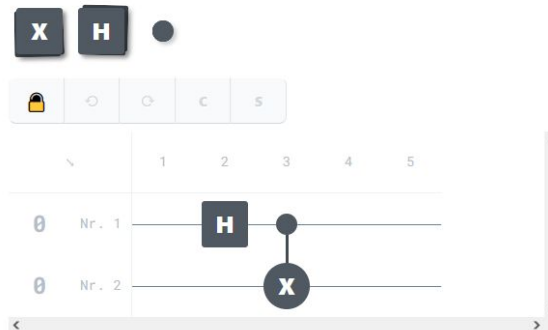
Bewerte anschließend: Wie schwierig würdest du das Problem auf einer Skala von 1 (kann doch jedes Kind) bis 10 (war in der Zeit gar nicht zu lösen) einstufen? Und warum?



Der Film „Die Hoch-Bumby“ wurde komplett von Computern berechnet (CC-BY-Bilder Foundation).

SEHE  
NÄCHSTE  
SEITE

! Den Wert welches Eingabe-Qubits müssen wir ändern, damit eine 50% Wahrscheinlichkeit besteht als Ausgabe **10** und eine 50% Wahrscheinlichkeit als Ausgabe **01** zu erhalten?  
Überprüfe Deine Vermutung!



## Auswertung

Nach 1000 Messungen zeigt sich folgende Ergebnisverteilung:

1	00⟩	50% Wahrscheinlichkeit
2	01⟩	0% Wahrscheinlichkeit
3	10⟩	0% Wahrscheinlichkeit
4	11⟩	50% Wahrscheinlichkeit

Lösung

OER-Materialien unter:

<https://computingeducation.de/proj-quanteninformatik/>

# Unterrichtsmaterial unter [Computingeducation.de](https://www.computingeducation.de)



**ComputingEducation**  
Computer science explained!

HOME ÜBER UNS PUBLIKATIONEN THEMEN

## OBTS NEWS

21. JAN 2028

### QUANTENINFORMATIK

#### Quanteninformatik

Quanteninformatik gilt als eine der vielversprechendsten neuen technologischen Entwicklungen - Forscherinnen und Forscher...

1 MIN. LESEZEIT

### Von DATEN und BÄUMEN

KI  
**IT2School - Module zu künstlicher Intelligenz**

Künstliche Intelligenz In Kooperation mit der Wissensfabrik haben wir im Rahmen des IT2School-Projekts vier Unterrichtsmodule zum Thema...

### Agile Schule

AGILE METHODEN  
**Agile Schule**

Agile Projektarbeit macht Schule! Die aus der Softwareentwicklung bekannte Vorgehensweise lässt sich gewinnbringend auf den...

### Schlag den Roboter – Maschinelles Lernen erfahren

KI  
**Schlag den Roboter – Maschinelles Lernen erfahren**

Schlag den Computer im Mini-Schach (online). Hinter den jüngsten Entwicklungen im Bereich künstliche Intelligenz stecken vor allem...



Informatische Bild

## Inform Schul

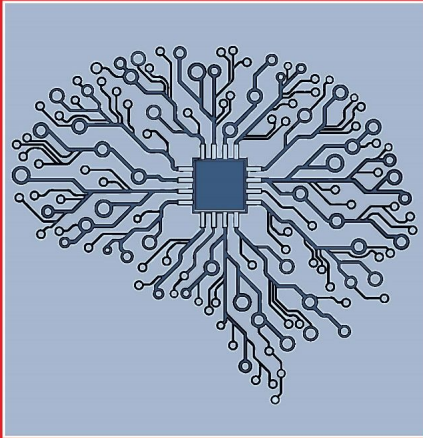
Zeitgemäße, gute  
der digitalen Welt  
Im Jahr 2022 ist di  
Bereich der inform  
diese wichtige Art  
Fachgruppe Didak

ARTIKEL SENDEN

<http://www.log-in-verlag.de/>

# LOG IN

Informatische Bildung und Computer in der Schule



Künstliche Intelligenz  
und Unterricht.  
Menschheitsträume.  
So lernen Maschinen!  
Nachdenken über KI.

Maschinelles Lernen,  
KI im Unterricht.  
IT-Ethik in der Schule.  
Robby lernt –  
aber nicht alles!

Nr. 193/194  
2020

LOG IN Verlag

Registrieren Einloggen



GESELLSCHAFT FÜR INFORMATIK  
FA INFORMATISCHE BILDUNG IN SCHULEN

Open Access  
Impressum | Datenschutz | Hinweis: Diese Webseite verwendet technisch notwendige Cookies!



GESELLSCHAFT FÜR INFORMATIK  
FG DIDAKTIK DER INFORMATIK



**Vielen Dank für die Aufmerksamkeit**