

Materialien zur Vorbereitung auf das  
1. Staatsexamen Informatik (Didaktik der  
Informatik) in Bayern

FSI Lehramt Informatik (Erlangen)

22. Juli 2024

# Lizenz

Dieses Werk ist unter einer Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License lizenziert. Es kann wie folgt referenziert werden:

Materialien zur Vorbereitung auf das 1. Staatsexamen Informatik (Didaktik der Informatik) in Bayern © 2024 by FSI Lehramt Informatik (Erlangen) is licensed under CC BY-NC-SA 4.0. To view a copy of this license, visit <https://creativecommons.org/licenses/by-nc-sa/4.0>.

# Inhaltsverzeichnis

<b>1</b>	<b>Vorwort</b>	<b>3</b>
1.1	Aufbau und Motivation . . . . .	3
1.2	Mitmachen . . . . .	3
<b>2</b>	<b>Themenbereiche</b>	<b>4</b>
<b>3</b>	<b>Lösungsversuche</b>	<b>5</b>
3.1	F21T3 . . . . .	5

# 1 Vorwort

## 1.1 Aufbau und Motivation

Ziel dieses Dokuments und des zugehörigen Repositorys ist es, eine studentische Sammlung an Lösungsvorschlägen zu den Aufgaben vergangener 1. Staatsexamen im Fach Informatik in Bayern zu erstellen. Wir wollen dadurch kooperative und nachhaltige Vorbereitung fördern. Da wir leider kein Urheberrecht an den Aufgabenstellungen haben, können wir diese jedoch nicht veröffentlichen. Wir bitten um Verständnis.

Als Ausgleich dafür soll diese Sammlung auch einen Überblick über die Themenbereiche der Informatik geben, die in den Staatsexamen abgefragt werden. Dabei soll idealerweise eine Sammlung an Definitionen, Algorithmen, Sätzen und Ähnlichem entstehen, die potenziell mit eigens erstellten Beispielen ergänzt werden.

## 1.2 Mitmachen

*von Max Mustermann*

E-Mail: `max@mustermann.de`

Wir freuen uns selbstverständlich über jede Art von Beitrag. Sei es das Hinzufügen von Lösungen, das Korrigieren von Fehlern oder das Ergänzen von Inhalten in den Themenbereichen. Ausführliche Informationen zum Mitwirken lassen sich auf Github finden.

Wir möchten auch die Möglichkeit geben, eigene Lösungsvorschläge für Aufgaben mit den eigenen Autor-Informationen zu versehen. Dies sieht man beispielsweise an diesem Unterkapitel. Dies ist aber komplett optional!

## 2 Themenbereiche

## 3 Lösungsversuche

### 3.1 F21T3

*von Max Ole Elliger*

E-Mail: [ole.elliger@fau.de](mailto:ole.elliger@fau.de)

#### Aufgabe 1

**Erlebnis für die SuS** Die Arbeit mit den Robotern bietet den Schülerinnen und Schülern (SuS) die Möglichkeit, erarbeitete Algorithmen direkt auszuprobieren, und das nicht nur am Bildschirm. Dies lässt sich anhand des EIS-Prinzips nach Bruner nachvollziehen: Während die reine Programmierarbeit eher symbolischen Charakter hat, bietet sich den SuS durch die Roboter die Möglichkeit, ihre Erarbeitung enaktiv auszuprobieren.

Im Vergleich dazu bietet eine Simulation eines Roboters diese Möglichkeit nicht direkt. Viel mehr findet das Ausprobieren (nach dem EIS-Prinzip) ikonisch statt.

**Praktische Umsetzung im Unterricht** Während eine simulierte Umgebung wie Robot Karol einfach am Computer bedient werden kann, benötigen die Roboter mehr physischen Platz im Klassenzimmer. Je nach Klassengröße und Platz pro Schreibtisch kann dies ein Problem darstellen.

**Anforderungen an die Programmierumgebung** Mit Blick auf die erste Kompetenzerwartung des Lehrplanauszugs sollte die Programmierumgebung Möglichkeiten bieten, diese auch ohne Programmierkenntnisse erforschen zu können, schließlich sollen die SuS Problemstellungen u. a. aus ihrer Erfahrungswelt analysieren und strukturieren. Dies wird beispielsweise in Robot Karol durch die Schaltflächen zur Navigation durch die Welt ermöglicht.

Abgesehen davon sollte die Umgebung typische Programmierbausteine beinhalten, wie z.B. Sequenz, Bedingte Anweisung, Wiederholung mit fester Anzahl oder Bedingte Wiederholung.

Zudem sollte die Bedienung intuitiv sein, sodass das Testen von Programmen keine allzu große Hürde bietet.

#### Aufgabe 2

Hier die geforderte Grobplanung:

Stunde	Beschreibung	Feinziel
1	Betrachtung von Algorithmen aus der Erfahrungswelt der SuS, Definition der Begriffe <i>Algorithmus</i> , <i>Anweisung</i>	Die SuS sind in der Lage, den Begriff <i>Algorithmus</i> anhand von Beispielen aus ihrer Lebenswirklichkeit zu verdeutlichen.
2	Einarbeiten in die Programmierungsumgebung des Roboters, Schreiben einfacher sequentieller Programme, Begriff <i>Sequenz</i>	Die SuS sind in der Lage, einfache sequentielle Programme in der Roboterprogrammierungsumgebung zu schreiben.
3	Programme mit bedingten Anweisungen	Die SuS sind in der Lage, <i>bedingte Anweisungen</i> gezielt in Programmen einzusetzen.
4	Wiederholung mit fester Anzahl	Die SuS sind in der Lage, <i>Wiederholungen mit fester Anzahl</i> gezielt in Programmen einzusetzen.
5	Bedingte Wiederholung	Die SuS sind in der Lage, <i>bedingte Wiederholungen</i> gezielt in Programmen einzusetzen.
6	Projektarbeit	Die SuS sind in der Lage, mithilfe der erlernten Programmierelemente größere Aufgabenstellungen zu bewältigen.

Tabelle 3.1: Grobplanung

### Aufgabe 3

#### Aufgabenstellung

1. Nenne die Definition des Begriffs *Algorithmus*! Verdeutliche den Begriff zudem an einem Beispiel aus dem Alltag!
2. Im folgenden ist ein Programmausschnitt dargestellt. Nenne die beiden verwendeten Programmierkonstrukte!

```

wiederhole 5 mal
    Schritt ()
    Drehen ()
endewiederhole

```

3. Programmiere folgende Situation für den Roboter: Wenn der Roboter vor einer Wand steht, dann soll er sich einmal nach links drehen und einen Schritt laufen. Ansonsten soll er einen Schritt nach vorne gehen.
4. Programmiere folgende Situation für den Roboter: Der Roboter soll sich dauerhaft abwechselnd einmal links drehen und anschließend nach rechts drehen.

## Lösungsvorschlag

1. Ein Algorithmus ist eine *endliche* und *eindeutige* Abfolge von Anweisungen. Ein Beispiel aus dem Alltag ist ein Backrezept: Eine Anweisung ist beispielsweise, man solle 200 Gramm Mehl in eine Schüssel geben. Bekannterweise besteht ein Rezept nur aus endlich vielen Anweisungen, und diese sollen in einer klar vorgegebenen eindeutigen Reihenfolge durchgeführt werden.
2. *Wiederholung mit fester Anzahl, Sequenz*
3.

```
wenn IstWand() dann
    LinksDrehen()
    Schritt()
sonst
    Schritt()
endewenn
```
4.

```
wiederhole solange True
    LinksDrehen()
    RechtsDrehen()
endewiederhole
```

## Aufgabe 4

Nach der üblichen Begrüßung und Behandlung organisatorischer Themen sollen alle SuS eine Anleitung zum Bau eines Papierfliegers niederschreiben. Sie dürfen dabei nicht selbst ein Exemplar bauen. Nach 10 Minuten tauschen sie Anleitungen mit einer festgelegten Partnerin oder Partner aus, die nicht direkt daneben sitzen. Anschließend sollen Papierflieger gemäß den übergebenen Anleitungen gebaut werden.

Diese vorbereitende Übung soll zu dem Ergebnis führen, dass Anleitungen leicht missverstanden werden können. Da davon auszugehen ist, dass die Anleitungen nicht ausreichend detailliert geschrieben werden, wird voraussichtlich kaum ein Papierflieger den Erwartungen der SuS entsprechen. Dies führt zu einem Unterrichtsgespräch, welche Eigenschaften gute Anleitungen haben müssen. Hierbei sollten folgende Eigenschaften gefunden werden:

- präzise
- möglichst kleinschrittig, Schritt für Schritt
- nachvollziehbar für die Leserin/den Leser

Mithilfe dieser Sammlung soll nun der Bezug zur Informatik im Unterrichtsgespräch thematisiert werden. Dabei soll der Unterschied zwischen Mensch und Computer besprochen werden: Computer haben im Gegensatz zu Menschen nicht die Möglichkeit, Anweisungen kreativ zu interpretieren.

Anschließend werden die Begriffe *Anweisung* und *Algorithmus* eingeführt und in Form eines Hefteintrages gesichert:

Einen eindeutigen und atomaren Einzelschritt nennen wir *Anweisung*.



Eine endliche und eindeutige Abfolge von Anweisungen nennen wir *Algorithmus*.

Nun sollen die SuS in Partnerarbeit Beispiele für Algorithmen aus dem Alltag finden. Nach 5 Minuten werden diese in Form eines Unterrichtsgespräches an der Tafel gesammelt. Dadurch bekommen die SuS Zeit, mit einem Peer über die neuen Begriffe zu reflektieren und nach Beispielen zu suchen, wobei sie diese im Anschluss überprüfen können. Zwecks Sicherung werden diese Beispiele dann auch im Heft festgehalten.

Zum Schluss sollen in Einzelarbeit die folgenden Beispiele dahingehend analysiert werden, ob Algorithmen dargestellt werden:

1. Zuerst fahren Sie zwei Kilometer geradeaus. Anschließend bie-  
gen Sie links ab und halten nach 200 Metern. Wenn Sie nun  
aussteigen, befinden Sie sich vor dem Bahnhof.
2. Addiere 1  
Addiere 1  
Addiere 1  
Addiere 1  
Addiere 1  
...
3. Schütte als Erstes 200 Gramm Mehl in die Schüssel. Anschlie-  
ßend kommen ...ähm... 50 Gramm Zucker oder 100 Gramm Salz  
dazu. Danach einfach kräftig umrühren und fertig ist der Ku-  
chen!

Während das erste Beispiel tatsächlich einen Algorithmus darstellt, stellen die anderen beiden Beispiele Gegenbeispiele dar, wobei Beispiel 2 auf die Endlichkeit und Beispiel 3 auf die Eindeutigkeit der Anweisungen hinweist. Dadurch können die SuS den Begriff besser abgrenzen.

Die Ergebnisse werden im Anschluss im Unterrichtsgespräch besprochen, um sicherzustellen, dass alle SuS die richtigen Ergebnisse haben.

## Aufgabe 5

**Einsatz in der 9. Klasse (Einstieg Programmierung)** Je nach konkreter zugehöriger Programmierungsumgebung lassen sich die Roboter auch für den Einstieg in die Programmierung in 9. Klasse verwenden. Die Roboter könnten zur Auffrischung des Wissens aus der 7. Klasse verwendet werden, um anschließend den Übergang zu einer neuen Programmiersprache (z.B. Python oder Java) zu erleichtern. Falls die zugehörige Programmierungsumgebung auch Möglichkeiten zur Objektorientierung bietet, könnte man auch diskutieren, ob man die Roboter weiterverwendet.

**Einsatz im Projekt in der 10. Klasse** In der 10. Klasse könnten die Roboter auch wieder eingesetzt werden, da diese verschiedene reale Problemstellungen ermöglichen. Beispielsweise könnte versucht werden, in Gruppen die Roboter auf ein Rennen vorzubereiten. Dafür sind diese natürlich auch geeignet.