

66118 Frühjahr 2021

Fachdidaktik (Gymnasium)

Aufgabenstellungen mit Lösungsvorschlägen

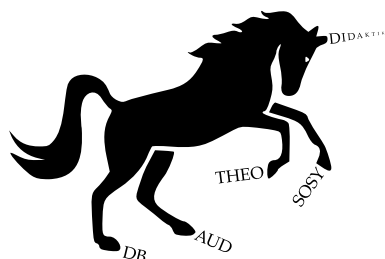


Die Bschlangaul-Sammlung

Hermine Bschlangaul and Friends

Aufgabenübersicht

Thema Nr. 2 [Modellierung, Programmierung, Fundamentale Ideen, Werkzeuge, Algorithmen]	3
Thema Nr. 3 [Klassensatz programmierbare Roboter]	5



Die Bschlangaul-Sammlung

Hermine Bschlangaul and Friends

Eine freie Aufgabensammlung mit Lösungen von Studierenden für Studierende zur Vorbereitung auf die 1. Staatsexamensprüfungen des Lehramts Informatik in Bayern.



Diese Materialsammlung unterliegt den Bestimmungen der Creative Commons Namensnennung-Nicht kommerziell-Share Alike 4.0 International-Lizenz.

Thema Nr. 2 [Modellierung, Programmierung, Fundamentale Ideen, Werkzeuge, Algorithmen]

Im LehrplanPLUS finden Sie für den „Lernbereich 3: Grundlagen der objektorientierten Modellierung und Programmierung (ca. 26 Stunden)“ für die 9. Jahrgangsstufe folgenden Text:

Kompetenzerwartungen:

Die Schülerinnen und Schüler...

- analysieren Objekte aus ihrer Erfahrungswelt (z. B. Fahrzeuge, Personen) hinsichtlich ihrer Eigenschaften (Attribute) und Fähigkeiten (Methoden) und abstrahieren sie zu Klassen. Sie stellen Objekte und Klassen als Grundlage einer möglichen Implementierung grafisch dar.
- deklarieren eine Klasse sowie die zugehörigen Attribute und Methoden in einer objektorientierten Programmiersprache.
- verwenden bei der Implementierung Wertzuweisungen, um Attributwerte zu ändern, und interpretieren diese als Zustandsänderung des zugehörigen Objekts.
- formulieren unter Verwendung der Kontrollstrukturen Algorithmen zu geeigneten Problemstellungen, u. a. durch grafische Darstellungen.
- implementieren Methoden auf der Grundlage gegebener Algorithmen objektorientiert, wobei sie sich des Unterschiedes zwischen Methodendefinition und Methodenaufruf bewusst sind. Dabei nutzen sie ggf. auch Methoden anderer Klassen.
- analysieren, interpretieren und modifizieren Algorithmen, wodurch sie die Fähigkeit erlangen, fremde Programme flexibel einzusetzen und kritisch zu bewerten.
- modellieren durch Klassendiagramme einfache Generalisierungshierarchien zu geeigneten Strukturen aus ihrer Erfahrungswelt.
- implementieren mithilfe einer objektorientierten Sprache einfache Generalisierungshierarchien; dabei nutzen sie das Konzept der Vererbung sowie die Möglichkeit, Methoden zu überschreiben

Inhalte zu den Kompetenzen:

- objektorientierte Konzepte, u. a. Objekt, Klasse, Attribut, Attributwert, Methode
- Variablenkonzept; Arten von Variablen: Parameter, lokale Variable und Attribute; Übergabewert
- Wertzuweisung zur Änderung von Variablenwerten

- Methoden: Methodenkopf, Methodenrumpf, Methodendefinition, Methodenauf-ruf, Übergabewert, Rückgabewert; Konstruktor als spezielle Methode; Standard-methoden zum Geben und Setzen von Attributwerten
- Algorithmus: Strukturelemente, grafische Darstellung, Pseudocode
- Datentypen: ganze Zahlen, Gleitkommazahlen, Wahrheitswerte, Zeichen, Zeichenketten
- Generalisierung und Spezialisierung: Ober- und Unterklasse, Vererbung von Attributen und Methoden an Unterklassen, Überschreiben von Methoden
- Fachbegriffe: Parameter, Übergabewert, Rückgabewert, lokale Variable, Wertzuweisung, Konstruktor, Methodenkopf, Methodenrumpf, Vererbung, Generalisierung, Spezialisierung, Oberklasse, Unterklasse

Aufgabe

- (a) In obigem Lehrplanabschnitt wird mehrfach „Modellierung“ und „Programmierung“ genannt.
 - (i) Grenzen Sie die beiden Begriffe gegeneinander ab und begründen Sie, weshalb man beides im Informatikunterricht benötigt.
 - (ii) Erläutern Sie das Konzept der „Fundamentalen Ideen“ nach Schwill! Nehmen Sie dabei Bezug auf „Modellierung“ und „Programmierung“.
- (b) Im Informatikunterricht ist sowohl der Einsatz einer blockbasierten als auch einer textbasierten Sprache denkbar.
 - (i) Gehen Sie jeweils auf Vor- und Nachteile der beiden Möglichkeiten anhand konkreter Beispiele ein.
 - (ii) Entscheiden Sie dann begründet, welche Wahl Sie in der 9. Jahrgangsstufe treffen würden.
- (c) Im zitierten Lehrplanabschnitt ist die Verwendung von Kontrollstrukturen zur Formulierung von Algorithmen vorgesehen. Es gibt aber keine konkrete Auflistung, welche Kontrollstrukturen besprochen werden sollen.
 - (i) Geben Sie einen kurzen Überblick über die Kontrollstrukturen imperativer bzw. objektorientierter Programmiersprachen an, die hier fachlich in Frage kommen könnten.
 - (ii) Im Hinblick darauf, dass für den gesamten Lehrplanabschnitt 26 Unterrichtsstunden zur Verfügung stehen, kann es notwendig sein, sich auf wenige Kontrollstrukturen beschränken zu müssen. Entscheiden Sie, welche Kontrollstrukturen Sie wählen würden und erklären Sie, in welcher Reihenfolge Sie diese in der 9. Jahrgangsstufe einführen würden! Begründen Sie Ihre Ausführungen.
- (d) Erstellen Sie für den zitierten Lehrplanabschnitt einen Sequenzplan mit 13 Doppelstunden! Geben Sie für jede dieser Doppelstunden eine kurze, nachvollziehbare Beschreibung der jeweiligen Zielsetzung an.

- (e) Erstellen Sie eine Feinplanung für eine Doppelstunde zur bedingten Wiederholung unter Verwendung der von Ihnen in Aufgabe 2b) gewählten Art der Programmiersprache.
- (i) Legen Sie zunächst drei beobachtbare Lernziele fest.
 - (ii) Skizzieren Sie eine Einführungsaufgabe, die die Schülerinnen und Schüler zu diesem Thema bearbeiten sollen.
 - (iii) Erläutern Sie anschließend den Unterrichtsfortgang nachvollziehbar (textuelle Beschreibung). Begründen Sie dabei Ihre fachdidaktischen Entscheidungen und gruppieren Sie Ihren Text nach Unterrichtsphasen.

Thema Nr. 3 [Klassensatz programmierbare Roboter]

Der LehrplanPLUS der 7. Jahrgangsstufe (Natur und Technik) des neunjährigen Gymnasiums enthält den folgenden Lehrplanpunkt:

NT7 2.3 Beschreibung von Abläufen durch Algorithmen (ca. 11 Std.)

Kompetenzerwartungen

Die Schülerinnen und Schüler ...

- analysieren und strukturieren geeignete Problemstellungen u. a. aus ihrer Erfahrungswelt (z. B. Bedienung eines Geräts), entwickeln Algorithmen zu deren Lösung und beschreiben diese unter effizienter Verwendung von Kontrollstrukturen.
- setzen unter sinnvoller Nutzung algorithmischer Bausteine einfache Algorithmen mithilfe geeigneter Programmierwerkzeuge um.

Inhalte zu den Kompetenzen:

- Algorithmus: Definition des Begriffs, Strukturelemente (Anweisung, Sequenz, ein- und zweiseitig bedingte Anweisung, Wiederholung mit fester Anzahl, Wiederholung mit Bedingung)
- Fachbegriffe: Algorithmus, Anweisung, Sequenz, ein- und zweiseitig bedingte Anweisung, Wiederholung mit fester Anzahl, Wiederholung mit Bedingung

Aufgabe

Gehen Sie bei den folgenden Aufgaben von folgendem Szenario aus:

Ihrer Schule wurde eine Sachspende über einen Klassensatz programmierbare Roboter angeboten. Dabei handelt es sich um fertig montierte, fahrbare Kleinroboter mit je zwei Abstandssensoren (siehe Abbildung).

Der Roboter verfügt über zwei unabhängige Motoren, deren Geschwindigkeit gesteuert werden kann. Kurven können gefahren werden, indem die Motoren unterschiedlich schnell laufen. Zur Programmierung steht eine einfache Umgebung zur Verfügung.

- (a) Der oben zitierte Lehrplanpunkt wird oftmals anhand einer Software mit einem simulierten, steuerbaren Roboter — z.B. Robot Karol — unterrichtet. Diskutieren Sie Unterschiede zwischen dieser Simulation und dem Einsatz der beschriebenen realen Roboter aus fachdidaktischer Sicht. Beschreiben Sie, welchen Ansprüchen die Programmierumgebung genügen sollte, damit ein Unterrichtseinsatz in Natur und Technik zu obigem Lehrplanpunkt möglich und sinnvoll ist.

Lösungsvorschlag

Der größte Unterschied ist sicherlich, dass der hier beschriebene Roboter real, also anfassbar ist. SuS können unmittelbar in der Realität erfahren, was ihre Programmierung auslöst. Dies ist vor allem für solche SuS vorteilhaft, die einen Computer und dessen Programme als eine Art Blackbox verstehen, d.h. die z.B. Robot Karol als Spielzeug ansehen, das keinen Bezug zur Lebenswelt hat, da er nur im Rechner existiert. Zusätzlich sehe ich den Roboter aus oben genanntem Grund als genderneutraler als Robot Karol, der wohl eher die computerbegeisterten Jungs in der Klasse ansprechen wird.

Darüber hinaus kann man anhand des Roboters Fragen aus den Bereich Technik aufnehmen, wie z.B. „Wie funktioniert eine Lenkung im Auto?“ im Gegensatz zur Frage „Wie kann man mit einer starren Achse wie z.B. bei einem Rollstuhl lenken?“. Solche Querverweise sind bei Robot Karol nicht möglich.

Für Robot Karol spricht hingegen, dass die Möglichkeiten des Programmierens vielfältiger sind. So kann der reale Roboter z.B. keine Steine „hinlegen“. Somit erscheint mir auf den ersten Blick Robot Karol abwechslungsreicher zu sein.

Ein weiterer Unterschied könnte darin liegen, dass Robot Karol eine textbasierte Programmierung erfordert, wohingegen der reale Roboter blockbasiert funktionieren könnte (je nach Hersteller). Der große Vorteil blockbasierter Programmiersprachen wie z.B. Snap! liegt beim Einstieg in das Thema „Programmierung“ darin, dass sich die SuS auf die Art und Weise, wie programmiert wird, konzentrieren können und sich keine Gedanken darüber machen müssen, wie nun der richtige Befehl lautet. Sie können sich also auf die Strukturen des Programmierens konzentrieren.

Ansprüche an die Programmierumgebung

Der Roboter sollte eine einfache, intuitiv strukturierte Benutzeroberfläche aufweisen. Vorteilhaft wäre hierfür eine App für Tablets, die sich über das W-LAN der Schule mit den Robotern verbinden kann. Die App sollte blockbasiert sein, wobei es für die einzelnen Fähigkeiten des Roboters vorgefertigte Bricks geben muss. Darüber hinaus müssen die im Lehrplan geforderten Sequenzen, bedingte Anweisungen und Wiederholungen möglich sein. Die erstellten Programme sollten für jede SuS speicherbar sein, so dass in der Folgestunde weitergearbeitet werden kann.

- (b) Geben Sie eine Grobplanung der Unterrichtssequenz für oben zitierten Lehrplanpunkt unter Nutzung der realen Roboter an. Gehen Sie dabei von insgesamt sechs Doppelstunden aus! Nennen Sie für jede Stunde ein beobachtbares Feinziel.

Anmerkung: Jede Doppelstunde besteht nach der in folgenden beschriebenen Wissensvermittlung per Lehrerinput aus ausreichend Aufgaben, die die SuS selbstständig lösen sollen. Die Schüleraktivierungsphasen werden nicht explizit in der folgenden Grobübersicht erwähnt, sind aber wesentlicher Bestandteil der Doppelstunde.

- 1.DS: Einführung in die Bedienung des Roboters** SuS erhalten einen Grobübersicht über die Funktionsmöglichkeiten des Roboters und die einzelnen Bricks zur Programmierung in der zugehörigen App -> Klärung der Fachbegriffe Anweisung und Sequenz Feinziel: SuS sind am Ende der Stunde in der Lage, einfache Bewegungen des Roboters zu programmieren und auszuführen.
- 2.DS: Wiederholung mit fester Anzahl** SuS lernen, mit Wiederholungen Programmsequenzen mehrfach hintereinander auszuführen. Auch eine Schachtelung von Wiederholungen werden gelernt und deren Sinn erarbeitet -> Klärung des Fachbegriffs Wiederholung mit fester Anzahl Feinziel: SuS sind am Ende der Stunde in der Lage sein, den Roboter mit Hilfe von Wiederholungen z.B. Quadrate und Rechtecke automatisch fahren zu lassen.
- 3.DS: Wiederholung mit Bedingung** SuS lernen mögliche Bedingungen kennen und die Art, wie man mit diesen programmieren kann. SuS erkennen den Vorteil gegenüber der Wiederholung mit fester Anzahl erarbeitet. SuS lernen den Begriff Algorithmus im Zusammenhang mit ihren Programmsequenzen kennen -> Klärung der Fachbegriffe Wiederholung mit Bedingung und Algorithmus Feinziel: SuS sind am Ende der Stunde in der Lage sein, den Roboter mit Hilfe von Wiederholungen mit Bedingungen zu programmieren.
- 4.DS: Aufgabenstunde** SuS erhalten Aufgaben (Schwierigkeit variabel), die sie mit den bisherigen Unterrichtsinhalten lösen können. Jede SuS wählt Anfangsschwierigkeit selbst aus. Feinziel: SuS sind am Ende der Stunde in der Lage sein, vorgefertigte Probleme mit einem eigenen Programm lösen zu können.
- 5.DS: bedingte Anweisung** SuS lernen die Möglichkeit kennen, dem Roboter mögliche Alternativen im Programmablauf zu programmieren erarbeitet -> Klärung des Fachbegriffs ein- und zweiseitig bedingte Anweisung Feinziel: SuS sind am Ende der Stunde in der Lage sein, zweiseitig bedingte Anweisungen programmieren zu können
- 6.DS: Zusammenfassende Aufgabenstunde** Vgl. 4. DS

- (c) Entwerfen Sie einen schriftlichen Leistungsnachweis mit Lösungsskizze für eine Bearbeitungszeit von 20 Minuten. Geben Sie an, an welcher Stelle der Unterrichtssequenz Sie ihn einsetzen würden.

Test zu Beginn der 6.DS zur Überprüfung der Programmierfähigkeiten der SuS

Aufgabe 1: Erkläre den Unterschied zwischen einer einseitig und einer zweiseitig bedingten Anweisung!

Aufgabe 2: Schreibe in Anlehnung an den Unterricht die Hilfsprogramme „Linksdrehen“ und „Umdrehen“ auf!

Aufgabe 3: Unser Roboter soll das Ende eines verwinkelten Ganges finden. Der Gang ist seitlich durch Mauern begrenzt und kann sich nur um 90° nach links bzw. nach rechts biegen (keine Verzweigungen!!!!). Das Ende ist durch eine Sackgasse gegeben. Mögliches Beispiel:

Schreibe ein Programm, mit dem der Roboter für alle möglichen Gänge dieser Art den Endpunkt findet. Verwende hierbei sowohl Wiederholungen mit Bedingungen als auch zweiseitig bedingte Anweisungen!

Lösungsskizze:

Zu 2: Linksdrehen soll den Roboter um 90° nach links drehen. Dabei bewegt sich das rechte Rad vorwärts, das linke rückwärts. Umdrehen ist ein zweimaliges Linksdrehen!

Zu 3: Die SuS können auch die aus dem Programm bekannten Bricks darstellen!

```
Wiederhole solange NichtIstWand
  vorwärts
endeWiederhole
linksdrehen
Wenn IstWand
  Dann Umdrehen
    Wenn IstWand
      Dann fertig
    Sonst vorwärts
  endeWenn
Sonst vorwärts
endeWenn
```

- (d) Geben Sie für den Einstieg in die Unterrichtssequenz (erste Doppelstunde) eine Feinplanung an. Beschreiben Sie dabei den geplanten Ablauf detailliert und skizzieren Sie Tafelbilder, Hefteinträge, Arbeitsblätter o. Ä. (ggf. mit einer Musterlösung).

Einstieg/Motivation (5Min) Zeigen eines Videos zum Thema „Roboter in der Wirtschaft“ (z.B. in der Autoproduktion) Vorführen des schuleigenen Roboters

Kennenlernphase (15Min) Benutzeroberfläche der App wird per Beamer an die Wand projiziert und die einzelnen grundlegenden Bestandteile der App wird im Lehrervortrag erklärt. Arbeitsblatt mit Screenshot der App und einzelnen Bestandteile zum Ausfüllen durch die Schüler (hier nicht möglich, den Screenshot anzugeben) Die Begriffe Anweisung und Sequenz werden erklärt -> Tafelbild und Hefteintrag

Ausprobierphase (25Min) Die SuS erhalten einen Roboter, verbinden diesen mit der Bedienoberfläche wie in der Kennenlernphase beschrieben und führen erste Anweisungen aus (von SuS frei wählbar). Arbeitsauftrag: Wie kann sich

der Roboter rechtsdrehen? -> Gruppen- bzw. Einzelarbeit (nach Anzahl der Roboter)

Erarbeitungsphase (30Min) SuS sollen Sequenzen zu folgenden Aufgaben schreiben: Mein Roboter fährt einen Kreis mit 70 cm Radius. Mein Roboter fährt ein Rechteck mit Kantenlängen 40 und 60 cm. Wettbewerb: Welche Gruppe setzt die Vorgaben am besten um?

Wissenssicherungs- und Aufräumphase (10Min) SuS geben folgende Inhalte wiederholend in eigenen Worten wieder: Anweisung – Sequenz- grobe Steuerelemente des Roboters Aufräumen der Roboter

- (e) Geben Sie weitere Stellen des bayerischen Lehrplans für das Fach Informatik am neun-jährigen Gymnasium (NTG) an, an denen Roboter dieser oder ähnlicher Art sinnvoll eingesetzt werden könnten. Bewerten Sie in jedem Fall knapp, ob die eingangs beschriebenen Roboter dafür geeignet wären und nennen Sie ggf. Alternativen.

Lösungsvorschlag

Der Einsatz eines Roboters wäre vielleicht noch in der 9. Jahrgangsstufe bei der Einführung in die objektorientierte Programmierung möglich. Dabei kann anhand des Roboters folgende Lehrplaninhalte eingeführt werden:

analysieren Objekte aus ihrer Erfahrungswelt (z. B. Fahrzeuge, Personen) hinsichtlich ihrer Eigenschaften (Attribute) und Fähigkeiten (Methoden) und abstrahieren sie zu Klassen. Sie stellen Objekte und Klassen als Grundlage einer möglichen Implementierung grafisch dar.

deklarieren eine Klasse sowie die zugehörigen Attribute und Methoden in einer objektorientierten Programmiersprache.

verwenden bei der Implementierung Wertzuweisungen, um Attributwerte zu ändern, und interpretieren diese als Zustandsänderung des zugehörigen Objekts.

Um kompliziertere Vorgänge beschreiben zu können, erscheint der in der Aufgabe beschriebene Roboter zu limitiert zu sein. So haben z.B. Lego Mindstorm-Roboter aufgrund ihrer vielfältigen Sensoren viel mehr Möglichkeiten, weshalb diese wohl viel ansprechender für SuS der 9. Klasse sein werden.