

2.1.1 Übersichtsraster Unterrichtsvorhaben

I) Einführungsphase

Kachel I
<p>Thema <i>Was macht Informatik? . Einführung in die Inhaltsfelder der Informatik</i></p> <p>Inhaltsfelder:</p> <p>1. <u>Informatiksysteme</u></p> <p>Informatiksysteme und ihr genereller Aufbau</p> <p>(a) Daten und ihre Strukturierung (b) Algorithmen (c) Informatiksysteme (hier zwei konkrete Szenarien, <i>Greenfoot</i> und <i>BlueJ</i>)</p> <p>Der kompetente Umgang mit dem Schulnetzwerk</p> <p>(a) Erstellen und Anlegen von Ordnerstrukturen (b) Sortieren von Dateien und Ordnern (c) Einzelrechner und Netzwerk (d) Sicherheit und Datenschutz</p> <p>Inhaltliche Schwerpunkte:</p> <p>- Einsatz, Nutzung und Aufbau von Informatiksystemen</p> <p>Zentrale Kompetenzen:</p> <p>- Kommunizieren und Kooperieren - Darstellen und Interpretieren - Argumentieren</p> <p>Die Schülerinnen und Schüler</p> <p>- nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D)</p>

Kachel II

Thema

Grundlagen der objektorientierten Analyse, Modellierung und Implementierung

Die didaktische Lernumgebung ist *Greenfoot* mit einem Roboter-Szenario.

Inhaltsfeld:

1. Daten und ihre Strukturierung

Identifikation von Objekten und Klassen

An einem lebensweltnahen Beispiel werden Objekte und Klassen im Sinne der objektorientierten Modellierung eingeführt. (konkrete Gegenstandsbereiche aus der Lebenswelt der Schülerinnen und Schüler analysiert und im Sinne des objektorientierten Paradigmas strukturiert)

Analyse von Objekten und Klassen im Greenfoot-Szenario

(a) Schritte der objektorientierten Analyse, Modellierung und Implementation

(b) Analyse und Erprobung der Objekte im Greenfoot-Szenario
Der Fokus hier liegt auf Grundlagen der Unterscheidung zwischen Klasse und Objekt, Attribute, Methoden, Objektidentität und Objektzustand.

Inhaltliche Schwerpunkte:

- Objekte und Klassen
- Syntax und Semantik einer Programmiersprache

Zentrale Kompetenzen:

- Modellieren
- Implementieren
- Kommunizieren und Kooperieren
- Darstellen und Interpretieren

Die Schülerinnen und Schüler

- ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften und ihre Operationen (M),
- stellen den Zustand eines Objekts dar (D),
- modellieren Klassen mit ihren Attributen und ihren Methoden (M),
- implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I),
- implementieren Klassen in einer Programmiersprache, auch unter Nutzung dokumentierter Klassenbibliotheken (I).

Klassen.notebook

Umgang mit Greenfoot.notebook

AB – Robita von Hand bewegen

AB – Zentrale Begriffe

AB – Die Methode act()

AB – Eigene Methoden

Softwareentwicklung mit Greenfoot und BlueJ 1

Schöningh 2010 (Buch 1)

S 23/1-3; S. 24/4,5 [27/4,5]

Kachel III

<p>Thema Einführung in die Objektorientierung, Programmierung in Greenfoot <i>Algorithmische Grundstrukturen in Java Grundlagen der objektorientierten Analyse und Programmierung anhand von Beispielkontexten und Roboterszenarien</i> Die didaktische Lernumgebung ist Greenfoot mit einem Roboter-Szenario.</p> <p>Inhaltsfelder: 1. <u>Algorithmen</u> (a) bedingte Anweisungen (IF) (b) Wiederholungen (While-Schleife, for-Schleife) (c) Verknüpfung von Bedingungen durch die logischen Funktionen UND, ODER und NICHT (d) Systematisierung des Vorgehens zur Entwicklung von Algorithmen zur Lösung komplexerer Probleme (Teilprobleme) (e) Fehlersuche - Debugger 2. <u>Variablen und Methoden</u> (a) Implementierung eigener Methoden mit Klassen- und lokalen Variablen, auch zur Realisierung einer Zählschleife (b) Implementierung eigener Methoden mit Parameterübergabe und/oder Rückgabewert (c) switch-case (d) Implementierung von Konstruktoren</p> <p>Inhaltliche Schwerpunkte: - Objekte und Klassen - Syntax und Semantik einer Programmiersprache - Quelltext einer Java-Klasse - Analyse, Entwurf und Implementierung einfacher Algorithmen</p> <p>Zentrale Kompetenzen: - Argumentieren - Modellieren - Implementieren - Kommunizieren und Kooperieren</p> <p>Die Schülerinnen und Schüler - analysieren und erläutern einfache Algorithmen und Programme (A), - entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M), - implementieren einfache Algorithmen un-</p>	<p>Bedingte Anweisung.notebook AB Die Bedingte Anweisung IF PAP Buch 1 – S. 39/3,4; S. 40/5,6 Logische Verknüpfung UND.notebook Logische Verknüpfung O- DER.notebook AB – Die logischen Verknüpfungen UND, ODER und NICHT Buch 1 – S. 45/1; S. 46/4,5; S. 47/6,7,8 WHILE und DO-WHILE.notebook AB – WHILE und DO-WHILE AB – WhileDoWhile-Aufgaben Buch 1 – S. 51/2,3; S. 52/4 – 7 FOR-Schleife.notebook AB – FOR-Schleife Buch 1 – S. 55/1; S. 56/3,4 AB - For-Schleife- WeiterführendeAufgaben Buch 1 – S. 63/PI, PII [PIII] Schachbrett.notebook AB – Schachbrett Variablen.notebook AB – Variablen Buch 1 – S. 84/2,3 Parameter.notebook Rückgabewert.notebook AB – Parameter, Rückgabewert Buch 1 – S.89/2,3,4,5; S. 90/7; S. 91/9 AB – Übersicht Variablen Buch 1 – S. 92/A2; S. 93/A3,A4 [A5]</p>
--	---

<p>ter Beachtung der Syntax und Semantik einer Programmiersprache,</p> <ul style="list-style-type: none">- modifizieren einfache Algorithmen und Programme,- testen Programme schrittweise anhand von Beispielen,- interpretieren Fehlermeldungen und korrigieren den Quellcode- ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen zu (M),- implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen	
---	--

Kachel IV

Thema

Das ist die digitale Welt! – Einführung in die Grundlagen, Anwendungsgebiete und Verarbeitung binärer Codierung

Wie werden binäre Informationen gespeichert und wie können sie davon ausgehend weiter verarbeitet werden?

Wie unterscheiden sich analoge Medien und Geräte von digitalen Medien und Geräten?

Wie ist der Grundaufbau einer digitalen Rechenmaschine?

Inhaltsfelder:

1. Analoge und digitale Aufbereitung und Verarbeitung von Daten

(a) Erarbeitung der Unterschiede von analog und digital

(b) Zusammenfassung und Bewertung der technischen Möglichkeiten von analog und digital

2. Der Umgang mit binärer Codierung von Informationen

(a) Das binäre (und hexadezimale) Zahlensystem

(b) Binäre Informationsspeicherung

(c) Binäre Verschlüsselung - ASCII

Inhaltliche

Schwerpunkte:

- Binäre Codierung und Verarbeitung
- Besondere Eigenschaften der digitalen Speicherung und Verarbeitung von Daten

Zentrale Kompetenzen:

- Argumentieren
- Kommunizieren und Kooperieren
- Darstellen und Interpretieren

Die Schülerinnen und Schüler

- bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A)

- stellen ganze Zahlen und Zeichen in Binärcodes dar (D),

- interpretieren Binärcodes als Zahlen und Zeichen (D)

- nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K)

- //implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I)

Binärcodierung.**notebook**

AB **Ë** Binärcodierung

Negative Binärzahlen.**notebook**

AB – Negative Binärzahlen Hexadezimalsystem.**notebook**

AB – Hexadezimalsystem

Informatik 1 Ë Schöningh 2014

(Buch 2)

S. 76/7

Kachel V

Thema: Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand lebensnaher Anforderungsbeispiele
Wie werden realistische Systeme anforderungsspezifisch reduziert, als Entwurf modelliert und implementiert?
Wie kommunizieren Objekte und wie wird dieses dargestellt und realisiert?
Eine mögliche didaktische Lernumgebung ist *BlueJ*.

Inhaltsfelder:

1. Umsetzung von Anforderungen in Entwurfsdiagramme

- (a) Aus Anforderungsbeschreibungen werden Objekte mit ihren Eigenschaften identifiziert
- (b) Gleichartige Objekte werden in Klassen (Entwurf) zusammengefasst und um Datentypen und Methoden erweitert
- (c) Java-Editor

2. Implementationsdiagramme als erster Schritt der Programmierung

- (a) Erweiterung des Entwurfsdiagramms um Konstruktoren
- (b) Festlegung von Datentypen in Java, sowie von Rückgaben und Parametern
- (c) Entwicklung von Klassendokumentationen
- (d) //Erstellung von Sequenzdiagrammen als Vorbereitung für die Programmierung

3. Programmierung anhand der Dokumentation und des Implementations- und Sequenzdiagrammes

- (a) Klassen werden in Java-Quell-code umgesetzt
- (b) Einzelne Klassen und das Gesamtsystem werden anhand der Anforderungen und Dokumentationen auf ihre Korrektheit überprüft.

4. Vererbungsbeziehungen

- (a) Das Grundprinzip der Vererbung wird erarbeitet
- (b) Die Vorteile der Vererbungsbeziehungen
- (c) Zugriffskontrollen und get- und set-Methoden
- (d) Vererbung wird implementiert

5. Objektkommunikation

- (a) Objekte als Attribute
- (b) Methoden an Objekten aufrufen

6. Softwareprojekt

- (a) Analyse und Dekonstruktion eines Spiels (Modelle, Quelltexte)
- (b) Erweiterung des Spiels um weitere

Fahrzeuge.**notebook**

AB- UML – Klassen-Entwurfsdiagramm

VererbungFahrzeuge.**notebook**

AB **Ë** Vererbung getter-setter.**notebook**

AB – getter/setter-Aufgaben

Objekte als Attribute.**notebook** (am Beispiel **Jugendlicher mit Führerschein**)

Methoden an Objekten aufrufen.**notebook**

(am Beispiel **Frauchen mit Hund**) Schiedsrichter.**notebook**

AB **Ë** Schiedsrichter **Buch 2** **Ë** S.114/2

Rollenspiele.**notebook**

AB – Projekt Rollenspiele

<p>Funktionalitäten</p> <p>(c) Modellierung eines Spiels aufgrund einer Anforderungsbeschreibung, inklusive einer grafischen Benutzeroberfläche</p> <p>(d) (arbeitsteilige) Implementation des Spiels</p> <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> - Objekte und Klassen - Syntax und Semantik einer Programmiersprache - Analyse, Entwurf und Implementierung einfacher Algorithmen <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Kommunizieren und Kooperieren - Darstellen und Interpretieren - Argumentieren - Modellieren - Implementieren <p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> - analysieren und erläutern eine objektorientierte Modellierung (A), - stellen die Kommunikation zwischen Objekten grafisch dar (M), - ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), - modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M), - ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M), - ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M), - modellieren Klassen unter Verwendung von Vererbung (M), - testen Programme schrittweise anhand von Beispielen (I), - interpretieren Fehlermeldungen und korrigieren den Quellcode (I), - analysieren und erläutern einfache Algorithmen und Programme (A) - modifizieren einfache Algorithmen und Programme (I), - entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M). - stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D), - dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D) 	
--	--

Kachel VI

Thema

Analyse-, Such- und Sortieralgorithmen mit quadratischer Laufzeit anhand einer kontextbezogenen und komplexen Problemstellung und deren Realisierung und Implementierung
 Wie können Objekte bzw. Daten effizient gesucht und sortiert werden?
 Eine mögliche didaktische Lernumgebung ist *BlueJ*.

Inhaltsfelder:

1. Modellierung und Implementation von Datenansammlungen
 - (a) Modellierung von Attributen als Felder (Arrays)
 - (b) Deklaration, Instanziierung und Zugriffe auf ein Feld
 - (c) Grafikoberfläche GUI
2. Explorative Erarbeitung von Suchverfahren
 - (a) Erkundung von Strategien für das Suchen auf unsortierten Daten, auf sortierten Daten und mithilfe einer Berechnungsfunktion.
 - (b) Vergleich der drei Verfahren durch intuitive Effizienzbetrachtungen.
3. Systematisierung von Algorithmen und Effizienzbetrachtungen
 - (a) Formulierung (falls selbst gefunden) oder Erläuterung von mehreren Algorithmen im Pseudocode
 - (b) Anwendung von Sortieralgorithmen auf verschiedene Beispiele
 - (c) Bewertung von Algorithmen anhand der Anzahl der nötigen Vergleiche
 - (d) Effizienzbetrachtungen an einem konkreten Beispiel bezüglich der Rechenzeit und des Speicherplatzbedarfs

Inhaltliche Schwerpunkte:

- Objekte und Klassen
- Algorithmen zum Suchen und Sortieren
- Analyse, Entwurf und Implementierung einfacher Algorithmen

Zentrale Kompetenzen:

- Modellieren
- Argumentieren
- Implementieren
- Kommunizieren und Kooperieren
- Darstellen und Interpretieren

Die Schülerinnen und Schüler

Arrays.notebook

AB **Ë** Einstieg Arrays
AB - Würfelsimulation

GUI - Anzahl, Tauschen, Maximum, Minimum, Summe von-bis, Wie Oft
Buch 2 – S.139/1, S140/2 [S.141/3]

AB - PotenzenProdukt
AB – FibonacciReihe SiebEratosthenes

[**AB** – CDs]

LineareSuche.notebook

Buch 2 **Ë** S. 146/1

BinäreSuche.notebook

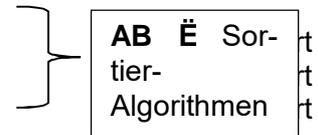
Buch 2 **Ë** S. 147/2
AB – Binäre Suche **Buch 2** – S. 148/3

Hashing.notebook

AB – Hashing **Buch 2** – S. 149/4

AB – SuchenSortieren **Buch 2** - S156/1

Selection
 Bubble
 Insertion



Buch 2 - S. 158/4

<ul style="list-style-type: none">- analysieren Such- und Sortieralgorithmen und wenden sie auf Beispiele an (D)- entwerfen einen weiteren Algorithmus zum Sortieren (M)- beurteilen die Effizienz von Algorithmen am Beispiel von Sortierverfahren hinsichtlich Zeit und Speicherplatzbedarf (A)- ordnen Attributen lineare Datenansammlungen zu (M)	
--	--