

Informatik

(Stand: 30.03.2015)

Inhalt

Seite

1 Die Fachgruppe Informatik des Hildegard-von-Bingen Gymnasiums 3

2 Entscheidungen zum Unterricht 5

2.1 Unterrichtsvorhaben 5

Da in den folgenden Unterrichtsvorhaben Inhalte in der Regel anhand von Problemstellungen in Anwendungskontexten bearbeitet werden, werden in einigen Unterrichtsvorhaben jeweils mehrere Inhaltsfelder angesprochen.

2.1.1 Übersichtsraster Unterrichtsvorhaben

I) Einführungsphase 6

II) Qualifikationsphase (Q1 und Q2) 10

2.1.2 Konkretisierte Unterrichtsvorhaben

I) Einführungsphase 15

II) Qualifikationsphase

2.2 Grundsätze der fachmethodischen und fachdidaktischen Arbeit

2.3 Grundsätze der Leistungsbewertung und Leistungsrückmeldung

3 Entscheidungen zu fach- und unterrichtsübergreifenden Fragen

4 Qualitätssicherung und Evaluation

1 Die Fachgruppe Informatik des Hildegard-von-Bingen-Gymnasiums, Köln

Beim Hildegard-von-Bingen-Gymnasium handelt es sich um eine vierzügige Schule im Zentrum von Köln mit zurzeit ca. 800 Schülerinnen und Schülern. Wie auch an den beiden Nachbarschulen wird das Fach Informatik schon seit vielen Jahren angeboten.

Das Fach Informatik wird am Hildegard-von-Bingen-Gymnasium ab der Jahrgangsstufe 8 im Wahlpflichtbereich II (WP II) dreistündig unterrichtet und von etwa einem Viertel der Schülerinnen und Schüler der jeweiligen Jahrgangsstufe besucht. In der zweijährigen Laufzeit dieser Kurse lernen die Schüler die Grundlagen der Algorithmik. Der Unterricht erfolgt dabei in enger Verzahnung mit Inhalten der Mathematik und Physik.

In der Sekundarstufe II bietet das Hildegard-von-Bingen-Gymnasium für die Schülerinnen und Schüler in allen Jahrgangsstufen jeweils einen Grundkurs in Informatik an.

Um insbesondere Schülerinnen und Schülern gerecht zu werden, die in der Sekundarstufe I keinen Informatikunterricht besucht haben, wird in Kursen der Einführungsphase besonderer Wert darauf gelegt, dass keine Vorkenntnisse aus der Sekundarstufe I zum erfolgreichen Durchlaufen des Kurses erforderlich sind.

Der Unterricht der Sekundarstufe II wird mit Hilfe der Programmiersprache Java in der Programmierumgebung BlueJ durchgeführt. In der EF kommt bei der Einführung in die objektorientierte Programmierung die Software Greenfoot zum Einsatz. Die Vorgehensweise orientiert sich in allen drei Jahrgangsstufen dabei am eingeführten Schulbuch:

| | |
|----------------|---|
| Informatik I | ISBN 978-3-14-037126-1 |
| Informatik II | ISBN 978-3-14-037122-3 und |
| Informatik III | ISBN 978-3-14-037125-4 Schöningh-Verlag |

Durch projektartiges Vorgehen, offene Aufgaben und Möglichkeiten, Problemlösungen zu verfeinern oder zu optimieren, entspricht der Informatikunterricht der Oberstufe in besonderem Maße den Erziehungszielen, Leistungsbereitschaft zu fördern, ohne zu überfordern.

Die gemeinsame Entwicklung von Materialien und Unterrichtsvorhaben, die Evaluation von Lehr- und Lernprozessen sowie die stetige Überprüfung und eventuelle Modifikation des schulinternen Curriculums durch die Fachkonferenz Informatik stellen einen wichtigen Beitrag zur Qualitätssicherung und -entwicklung des Unterrichts dar.

Zurzeit besteht die Fachschaft Informatik des Hildegard-von-Bingen-Gymnasiums aus zwei Lehrkräften, denen zwei Computerräume mit je ca 25 Computerarbeitsplätzen zur Verfügung stehen. Alle Arbeitsplätze sind an das schulinterne Rechnernetz angeschlossen, so dass alle Schülerinnen und Schüler über den zentralen Server der Schule Zugriff auf das Internet haben.

Der Unterricht erfolgt im 45-Minuten-Takt. Die Kursblockung sieht grundsätzlich für Grundkurse eine Doppelstunde und eine Einzelstunde vor.

2 Entscheidungen zum Unterricht

2.1 Unterrichtsvorhaben

Die Darstellung der Unterrichtsvorhaben im schulinternen Lehrplan besitzt den Anspruch, sämtliche im Kernlehrplan angeführten Kompetenzen abzudecken. Dies entspricht der Verpflichtung jeder Lehrkraft, Schülerinnen und Schülern Lerngelegenheiten zu ermöglichen, so dass alle Kompetenzerwartungen des Kernlehrplans von ihnen erfüllt werden können.

Die entsprechende Umsetzung erfolgt auf zwei Ebenen: der Übersichts- und der Konkretisierungsebene. Die in den Tabellen der Konkretisierungsebene angegebenen Kapitel und Materialien beziehen sich auf das im Unterricht verwendete Lehrwerk: Informatik, Lehrwerk für die gymnasiale Oberstufe, Schöningh

Im „Übersichtsraster Unterrichtsvorhaben“ (Kapitel 2.1.1) wird die für alle Lehrerinnen und Lehrer gemäß Fachkonferenzbeschluss verbindliche Verteilung der Unterrichtsvorhaben dargestellt. Das Übersichtsraster dient dazu, den Kolleginnen und Kollegen einen schnellen Überblick über die Zuordnung der Unterrichtsvorhaben zu den einzelnen Jahrgangsstufen sowie den im Kernlehrplan genannten Kompetenzen, Inhaltsfeldern und inhaltlichen Schwerpunkten zu verschaffen. Der ausgewiesene Zeitbedarf versteht sich als grobe Orientierungsgröße, die nach Bedarf über- oder unterschritten werden kann.

Während der Fachkonferenzbeschluss zum „Übersichtsraster Unterrichtsvorhaben“ zur Gewährleistung vergleichbarer Standards sowie zur Absicherung von Lerngruppenübertritten und Lehrkraftwechseln für alle Mitglieder der Fachkonferenz Bindekraft entfalten soll, beinhaltet die Ausweisung „konkretisierter Unterrichtsvorhaben“ (Kapitel 2.1.2) Beispiele und Materialien, die empfehlenden Charakter haben.

Da in den folgenden Unterrichtsvorhaben Inhalte in der Regel anhand von Problemstellungen in Anwendungskontexten bearbeitet werden, werden in einigen Unterrichtsvorhaben jeweils mehrere Inhaltsfelder angesprochen.

Didaktische Lernumgebung

Die Unterrichtsvorhaben in Informatik hängen in ihrer konkreten Umsetzung von der gewählten Programmierumgebung ab. In diesem Lehrplan wird zur Einführung in die objektorientierte Programmierung die didaktische Lernumgebung Greenfoot gewählt. Die Lehrtexte und Aufgaben des Lehrwerks „Informatik 1“ beziehen sich auch speziell auf ein bestimmtes Greenfoot-Szenario. Nach der Einarbeitungsphase sollte auf eine komplexere Lernumgebung umgestellt werden. Hier bieten sich verschiedene Möglichkeiten, z.B.: BlueJ, Eclipse, NetBeans etc. an.

2.1.1 Übersichtsraster Unterrichtsvorhaben

I) Einführungsphase

| Einführungsphase | |
|--|---|
| <p><u>Unterrichtsvorhaben E-I</u></p> <p>Thema: <i>Was macht Informatik? – Einführung in die Inhaltsfelder der Informatik</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none">• Kommunizieren und Kooperieren• Darstellen und Interpretieren• Argumentieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none">• Informatiksysteme• Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none">• Einsatz, Nutzung und Aufbau von Informatiksystemen• Wirkung der Automatisierung <p>Zeitbedarf: 5 Stunden</p> | <p><u>Unterrichtsvorhaben E-II</u></p> <p>Thema: <i>Grundlagen der objektorientierten Analyse, Modellierung und Implementierung</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none">• Modellieren• Implementieren• Darstellen und Interpretieren• Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none">• Daten und ihre Strukturierung• Formale Sprachen und Automaten <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none">• Objekte und Klassen• Syntax und Semantik einer Programmiersprache <p>Zeitbedarf: 10 Stunden</p> |

Einführungsphase

Unterrichtsvorhaben E-III

Thema:

Algorithmische Grundstrukturen in Java

Zentrale Kompetenzen:

- Argumentieren
- Modellieren
- Implementieren
- Kommunizieren und Kooperieren

Inhaltsfelder:

- Daten und ihre Strukturierung
- Algorithmen
- Formale Sprachen und Automaten

Inhaltliche Schwerpunkte:

- Objekte und Klassen
- Syntax und Semantik einer Programmiersprache
- Analyse, Entwurf und Implementierung einfacher Algorithmen

Zeitbedarf: 20 Stunden

Unterrichtsvorhaben E-V

Thema:

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

Zentrale Kompetenzen:

- Argumentieren
- Darstellen und Interpretieren
- Kommunizieren und Kooperieren

Inhaltsfelder:

- Informatiksysteme
- Informatik, Mensch und Gesellschaft

Inhaltliche Schwerpunkte:

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

Zeitbedarf: 8 Stunden

Einführungsphase

Unterrichtsvorhaben E-IV

Thema:

Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand lebensnaher Anforderungsbeispiele

Zentrale Kompetenzen:

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

Inhaltsfelder:

- Daten und ihre Strukturierung
- Algorithmen
- Formale Sprachen und Automaten

Inhaltliche Schwerpunkte:

- Objekte und Klassen
- Syntax und Semantik einer Programmiersprache
- Analyse, Entwurf und Implementierung einfacher Algorithmen

Zeitbedarf: 20 Stunden

Unterrichtsvorhaben E-VI

Thema:

Such- und Sortieralgorithmen

Zentrale Kompetenzen:

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

Inhaltsfelder:

- Algorithmen
- Daten und ihre Strukturierung

Inhaltliche Schwerpunkte:

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

Zeitbedarf: 9 Stunden

Einführungsphase

Unterrichtsvorhaben E-VII

Thema:

Leben in der digitalen Welt – Immer mehr Möglichkeiten und immer mehr Gefahren!?

Zentrale Kompetenzen:

- Argumentieren
- Darstellen und Interpretieren
- Kommunizieren und Kooperieren

Inhaltsfelder:

- Informatiksysteme
- Informatik, Mensch und Gesellschaft

Inhaltliche Schwerpunkte:

- Geschichte der automatischen Datenverarbeitung
- Wirkung der Automatisierung
- Dateisysteme

Zeitbedarf: 5 Stunden

Summe Einführungsphase: 77

II) Qualifikationsphase (Q1 und Q2) - GRUNDKURS

| Qualifikationsphase 1 | |
|--|---|
| <p><u>Unterrichtsvorhaben Q1-I</u></p> <p>Thema: <i>Wiederholung der objektorientierten Modellierung und Programmierung anhand einer kontextbezogenen Problemstellung</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none">• Argumentieren• Modellieren• Implementieren• Darstellen und Interpretieren• Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none">• Daten und ihre Strukturierung• Algorithmen• Formale Sprachen und Automaten• Informatiksysteme <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none">• Objekte und Klassen• Analyse, Entwurf und Implementierung von Algorithmen• Syntax und Semantik einer Programmiersprache• Nutzung von Informatiksystemen <p>Zeitbedarf: 8 Stunden</p> | <p><u>Unterrichtsvorhaben Q1-II</u></p> <p>Thema: <i>Modellierung und Implementierung von Anwendungen mit dynamischen, linearen Datenstrukturen</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none">• Argumentieren• Modellieren• Implementieren• Darstellen und Interpretieren• Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none">• Daten und ihre Strukturierung• Algorithmen• Formale Sprachen und Automaten <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none">• Objekte und Klassen• Analyse, Entwurf und Implementierung von Algorithmen• Algorithmen in ausgewählten informatischen Kontexten• Syntax und Semantik einer Programmiersprache <p>Zeitbedarf: 15 Stunden</p> |

Qualifikationsphase 1

Unterrichtsvorhaben Q1-III

Thema:

Suchen und Sortieren auf linearen Datenstrukturen

Zentrale Kompetenzen:

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

Inhaltsfelder:

- Algorithmen
- Formale Sprachen und Automaten

Inhaltliche Schwerpunkte:

- Analyse, Entwurf und Implementierung von Algorithmen
- Algorithmen in ausgewählten informatischen Kontexten
- Syntax und Semantik einer Programmiersprache

Zeitbedarf: 12 Stunden

Unterrichtsvorhaben Q1-IV

Thema:

Modellierung und Implementierung von Anwendungen mit dynamischen, nichtlinearen Datenstrukturen

Zentrale Kompetenzen:

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

Inhaltsfelder:

- Daten und ihre Strukturierung
- Algorithmen
- Formale Sprachen und Automaten

Inhaltliche Schwerpunkte:

- Objekte und Klassen
- Analyse, Entwurf und Implementierung von Algorithmen
- Algorithmen in ausgewählten informatischen Kontexten
- Syntax und Semantik einer Programmiersprache

Zeitbedarf: 12 Stunden

Qualifikationsphase 1

Unterrichtsvorhaben Q1-V

Thema:

Graphen

Zentrale Kompetenzen:

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

Inhaltsfelder:

- Algorithmen
- Formale Sprachen und Automaten

Inhaltliche Schwerpunkte:

- Analyse, Entwurf und Implementierung von Algorithmen
- Algorithmen in ausgewählten informatischen Kontexten
- Syntax und Semantik einer Programmiersprache

Zeitbedarf: 5 Stunden

Unterrichtsvorhaben Q1-VI

Thema:

Vielfalt beim Einsatz von Klassen durch Abstrakte Klassen, Polymorphie und MVC

Zentrale Kompetenzen:

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

Inhaltsfelder:

- Daten und ihre Strukturierung
- Algorithmen
- Formale Sprachen und Automaten
- Informatiksysteme

Inhaltliche Schwerpunkte:

- Objekte und Klassen
- Analyse, Entwurf und Implementierung von Algorithmen
- Syntax und Semantik einer Programmiersprache
- Nutzung von Informatiksystemen

Zeitbedarf: 12 Stunden

Qualifikationsphase 1

Unterrichtsvorhaben Q1-VII

Thema:

Kommunikation in und Aufbau von Netzwerken

Zentrale Kompetenzen:

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

Inhaltsfelder:

- Algorithmen
- Formale Sprachen und Automaten
- Informatiksysteme
- Informatik, Mensch und Gesellschaft

Inhaltliche Schwerpunkte:

- Analyse, Entwurf und Implementierung von Algorithmen
- Syntax und Semantik einer Programmiersprache
- Einzelrechner und Rechnernetzwerke
- Nutzung von Informatiksystemen
- Wirkung der Automatisierung

Zeitbedarf: 15 Stunden

Summe Qualifikationsphase 1: 79

Qualifikationsphase 2

Unterrichtsvorhaben Q2-I

Thema:

Modellierung und Nutzung von relationalen Datenbanken in Anwendungskontexten

Zentrale Kompetenzen:

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

Inhaltsfelder:

- Daten und ihre Strukturierung
- Algorithmen
- Formale Sprachen und Automaten
- Informatik, Mensch und Gesellschaft

Inhaltliche Schwerpunkte:

- Datenbanken
- Algorithmen in ausgewählten informatischen Kontexten
- Syntax und Semantik einer Programmiersprache
- Sicherheit

Zeitbedarf: 15 Stunden

Unterrichtsvorhaben Q2-II

Thema:

Endliche Automaten und formale Sprachen

Zentrale Kompetenzen:

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

Inhaltsfelder:

- Endliche Automaten und formale Sprachen

Inhaltliche Schwerpunkte:

- Endliche Automaten
- Grammatiken regulärer Sprachen
- Möglichkeiten und Grenzen von Automaten und formalen Sprachen

Zeitbedarf: 20 Stunden

Qualifikationsphase 2

Unterrichtsvorhaben Q2-III

Thema:

Prinzipielle Arbeitsweise eines Computers und Grenzen der Automatisierbarkeit

Zentrale Kompetenzen:

- Argumentieren
- Kommunizieren und Kooperieren

Inhaltsfelder:

- Informatiksysteme
- Informatik, Mensch und Gesellschaft

Inhaltliche Schwerpunkte:

- Einzelrechner und Rechnernetzwerke
- Grenzen der Automatisierung

Zeitbedarf: 12 Stunden

Unterrichtsvorhaben Q2-IV

Thema:

Sicherheit in der Informatik: Verschlüsselung und ihre Folgen

Zentrale Kompetenzen:

- Argumentieren
- Darstellen und Interpretieren
- Kommunizieren und Kooperieren

Inhaltsfelder:

- Algorithmen
- Formale Sprachen und Automaten
- Informatiksysteme
- Informatik, Mensch und Gesellschaft

Inhaltliche Schwerpunkte:

- Analyse, Entwurf und Implementierung von Algorithmen
- Syntax und Semantik einer Programmiersprache
- Nutzung von Informatiksystemen
- Wirkung der Automatisierung
- Grenzen der Automatisierung

Zeitbedarf: 12 Stunden

Summe Qualifikationsphase 2: 59 Stunden

2.1.2 Konkretisierte Unterrichtsvorhaben

Im Folgenden sollen die im *Unterkapitel 2.1.1* aufgeführten Unterrichtsvorhaben konkretisiert werden.

In der Einführungsphase werden die Entwicklungsumgebungen Greenfoot und BlueJ verwendet. Die folgenden Installationspakete und Dokumentationen stehen zur Verfügung:

- Installationspaket (Windows) ([Download](#))
- Installationspaket (MacOS) ([Download](#))
- Installationspaket (Linux) ([Download](#))
- Dokumentationen ([Download](#))

In der Qualifikationsphase werden die Unterrichtsvorhaben unter Berücksichtigung der Vorgaben für das Zentralabitur Informatik in NRW konkretisiert. Diese sind zu beziehen unter der Adresse

<http://www.standardsicherung.schulministerium.nrw.de/abitur-gost/fach.php?fach=15> (abgerufen: 14. 08. 2014)

I) Einführungsphase

Die folgenden Kompetenzen aus dem Bereich *Kommunizieren und Kooperieren* werden in allen Unterrichtsvorhaben der Einführungsphase vertieft und sollen aus Gründen der Lesbarkeit nicht in jedem Unterrichtsvorhaben separat aufgeführt werden:

Die Schülerinnen und Schüler

- verwenden Fachausdrücke bei der Kommunikation über informatische Sachverhalte (K),
- präsentieren Arbeitsabläufe und -ergebnisse (K),
- kommunizieren und kooperieren in Gruppen und in Partnerarbeit (K),
- nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung und gemeinsamen Verwendung von Daten unter Berücksichtigung der Rechteverwaltung (K).

Unterrichtsvorhaben EF-I

Thema: Was macht Informatik? – Einführung in die Inhaltsfelder der Informatik

Leitfragen: *Was macht Informatik? Welche fundamentalen Konzepte müssen Informatikerinnen und Informatiker in ihre Arbeit einbeziehen, damit informatische Systeme effizient und zuverlässig arbeiten können? Wo lassen sich diese Konzepte (in Ansätzen) in dem schuleigenen Netzwerk und Computersystem wiederfinden?*

Vorhabenbezogene Konkretisierung:

Das erste Unterrichtsvorhaben stellt eine allgemeine Einführung in das Fach Informatik dar. Dabei ist zu berücksichtigen, dass für manche Schülerinnen und Schüler in der Einführungsphase der erste Kontakt mit dem Unterrichtsfach Informatik stattfindet, so dass zu Beginn Grundlagen des Fachs behandelt werden müssen.

Im ersten Unterrichtsvorhaben werden die fünf Inhaltsfelder des Faches Informatik beispielhaft an einem Informatiksystem erarbeitet. Das Unterrichtsvorhaben ist so strukturiert, dass die Schülerinnen und Schüler anhand bekannter Alltagstechnik die Grundideen fundamentaler informatischer Konzepte (Inhaltsfelder) größtenteils selbstständig erarbeiten und nachvollziehen. Ausgehend von dem bekannten Bedienungs- und Funktionalitätswissen eines Navigationsgerätes werden die Strukturierung von Daten, das Prinzip der Algorithmik, die Eigenheit formaler Sprachen, die Kommunikationsfähigkeit von Informatiksystemen und die positiven und negativen Auswirkungen auf Mensch und

Gesellschaft thematisiert. Das am Beispiel Navigationsgerät erworbene Wissen kann auf weitere den Schülerinnen und Schülern bekannte Informatiksysteme übertragen werden. In einem letzten Schritt kann ausgehend von den Inhaltsfeldern das Schulnetzwerk in Ansätzen so analysiert werden, dass ein kompetenter Umgang mit diesem ermöglicht wird.

Zeitbedarf: 5 Stunden

Sequenzierung des Unterrichtsvorhabens:

| Unterrichtssequenzen | Zu entwickelnde Kompetenzen | Beispiele, Medien, Materialien |
|--|---|---|
| <p>1. Informatiksysteme und ihr genereller Aufbau</p> <p>(a) Daten und ihre Strukturierung (b) Algorithmen (c) Formale Sprachen und Automaten (d) Informatiksysteme (e) Informatik, Mensch und Gesellschaft</p> | <p>Die SuS</p> <ul style="list-style-type: none"> • bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A) • nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D) | <p>Kapitel 1 - Was macht Informatik Als Anschauungsmaterial bieten sich Informatiksysteme aus der Lebenswelt der Schüler sowie Navigationsgeräte und Landkarten an.</p> |
| <p>2. Der kompetente Umgang mit dem Schulnetzwerk</p> <p>(a) Erstellen und Anlegen von Ordnerstrukturen (b) Sortieren von Dateien und Ordnern (c) Eingabe von Befehlen über Eingabeaufforderung (d) Einzelrechner und Netzwerk (e) Sicherheit und Datenschutz</p> | | <p>Kapitel 1 - Was macht Informatik Schulnetzwerk, Benutzer- und Datenschutzbestimmungen der Schule</p> |

Unterrichtsvorhaben EF-II

Thema: Grundlagen der objektorientierten Analyse, Modellierung und Implementierung

Leitfrage: *Wie lassen sich Gegenstandsbereiche informatisch modellieren und in einem Greenfoot-Szenario informatisch realisieren?*

Vorhabenbezogene Konkretisierung:

Ein zentraler Bestandteil des Informatikunterrichts der Einführungsphase ist die Objektorientierte Programmierung. Dieses Unterrichtsvorhaben führt in die Grundlagen der Analyse, Modellierung und Implementierung in diesem Kontext ein.

Dazu werden zunächst konkrete Gegenstandsbereiche aus der Lebenswelt der Schülerinnen und Schüler analysiert und im Sinne des objektorientierten Paradigmas strukturiert. Dabei werden die grundlegenden Begriffe der Objektorientierung und Modellierungswerkzeuge wie Objektdiagramme und Klassendiagramme eingeführt.

Im Anschluss wird die objektorientierte Analyse für das Greenfoot-Szenario Planetenerkundung durchgeführt. Die vom Szenario vorgegebenen Klassen werden von Schülerinnen und Schülern in Teilen analysiert und entsprechende Objekte anhand einfacher Problemstellungen erprobt. Die Lernenden implementieren und testen einfache Programme. Die Greenfoot-Umgebung ermöglicht es, Beziehungen zwischen Klassen zu einem späteren Zeitpunkt (Kapitel 5) zu thematisieren. So kann der Fokus hier auf Grundlagen wie der Unterscheidung zwischen Klasse und Objekt, Attribute, Methoden, Objektidentität und Objektzustand gelegt werden.

Da in Kapitel 2 zudem auf die Verwendung von Kontrollstrukturen verzichtet wird und der Quellcode aus einer rein linearen Sequenz besteht, ist auf diese Weise eine Fokussierung auf die Grundlagen der Objektorientierung möglich, ohne dass algorithmische Probleme ablenken. Natürlich kann die Arbeit an diesen Projekten unmittelbar zum nächsten Unterrichtsvorhaben (Kapitel 3) führen. Dort stehen Kontrollstrukturen im Mittelpunkt.

Zeitbedarf: 10 Stunden

Sequenzierung des Unterrichtsvorhabens:

| Unterrichtssequenzen | Zu entwickelnde Kompetenzen | Beispiele, Medien, Materialien |
|--|--|---|
| <p>3. Identifikation von Objekten</p> <p>(a) An einem lebensweltnahen Beispiel werden Objekte und Klassen im Sinne der objektorientierten Modellierung eingeführt.</p> <p>(b) Objekte werden durch Objektdiagramme, Klassen durch Klassendiagramme dargestellt.</p> <p>(c) Die Modellierungen werden einem konkreten Anwendungsfall entsprechend angepasst.</p> | <p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • 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). | <p>Kapitel 2 - Einführung in die Objektorientierung</p> <p>2.1 Objektorientierte Modellierung</p> |
| <p>4. Analyse von Objekten und Klassen im Greenfoot-Szenario</p> <p>(a) Schritte der objektorientierten Analyse, Modellierung und Implementation</p> <p>(b) Analyse und Erprobung der Objekte im Greenfoot-Szenario</p> | | <p>Kapitel 2 - Einführung in die Objektorientierung</p> <p>2.2 Das Greenfoot-Szenario „Planetenerkundung“</p> <ul style="list-style-type: none"> • Von der Realität zu Objekten • Von den Objekten zu Klassen • Klassendokumentation • Objekte inspizieren • Methoden aufrufen • Objektidentität und Objektzustand |
| <p>5. Implementierung einfacher Aktionen in Greenfoot</p> <p>(a) Quelltext einer Java-Klasse</p> <p>(b) Implementation eigener Methoden, Dokumentation mit JavaDoc</p> <p>(c) Programme übersetzen und testen</p> | | <p>Kapitel 2 Einführung in die Objektorientierung</p> <p>2.3 Programmierung in Greenfoot</p> <ul style="list-style-type: none"> • Methoden schreiben • Programme übersetzen und testen |

Unterrichtsvorhaben EF-III

Thema: Algorithmische Grundstrukturen in Java

Leitfragen: *Wie lassen sich Aktionen von Objekten flexibel realisieren?*

Vorhabenbezogene Konkretisierung:

Das Ziel dieses Unterrichtsvorhabens besteht darin, das Verhalten von Objekten flexibel zu programmieren. Ein erster Schwerpunkt liegt dabei auf der Erarbeitung von Kontrollstrukturen. Die Strukturen Wiederholung und bedingte Anweisung werden an einfachen Beispielen eingeführt und anschließend anhand komplexerer Problemstellungen erprobt. Da die zu entwickelnden Algorithmen zunehmend umfangreicher werden, werden systematische Vorgehensweisen zur Entwicklung von Algorithmen thematisiert.

Ein zweiter Schwerpunkt des Unterrichtsvorhabens liegt auf dem Einsatz von Variablen. Beginnend mit lokalen Variablen, die in Methoden und Zählschleifen zum Einsatz kommen, über Variablen in Form von Parametern und Rückgabewerten von Methoden, bis hin zu Variablen, die die Attribute einer Klasse realisieren, lernen die Schülerinnen und Schüler die unterschiedlichen Einsatzmöglichkeiten des Variablenkonzepts anzuwenden.

Zeitbedarf: 20 Stunden

Sequenzierung des Unterrichtsvorhabens:

| Unterrichtssequenzen | Zu entwickelnde Kompetenzen | Beispiele, Medien, Materialien |
|--|--|---|
| 1. Algorithmen (a) Wiederholungen (While-Schleife, Zählschleife) (b) bedingte Anweisungen (c) Verknüpfung von Bedingungen durch die logischen Funktionen UND, ODER und NICHT (d) Systematisierung des Vorgehens zur Entwicklung von Algorithmen zur Lösung komplexerer Probleme | Die Schülerinnen und Schüler <ul style="list-style-type: none">• analysieren und erläutern einfache Algorithmen und Programme (A),• entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M),• ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen zu (M),• modifizieren einfache Algorithmen und | Kapitel 3 - Algorithmen 3.1 Kontrollstrukturen 3.2 Wiederholungen 3.3 Zählschleifen 3.4 Bedingte Anweisungen 3.5 Logische Operationen 3.6 Algorithmen entwickeln |

| | | |
|--|---|---|
| <p>2. Variablen und Methoden</p> <ul style="list-style-type: none"> (a) Implementierung eigener Methoden mit lokalen Variablen, auch zur Realisierung einer Zählschleife (b) Implementierung eigener Methoden mit Parameterübergabe und/oder Rückgabewert (c) Implementierung von Konstruktoren (d) Realisierung von Attributen | <p>Programme (I),</p> <ul style="list-style-type: none"> • implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I), • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I), • implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I), • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I), • testen Programme schrittweise anhand von Beispielen (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I). | <p>Kapitel 4 - Variablen und Methoden</p> <p>4.1 Variablen</p> <p>4.2 Methoden</p> |
|--|---|---|

Unterrichtsvorhaben EF-IV

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

Leitfragen: *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?*

Vorhabenbezogene Konkretisierung:

Das Unterrichtsvorhaben hat die binäre Speicherung und Verarbeitung sowie deren Besonderheiten zum Inhalt.

Im ersten Schritt erarbeiten die Schülerinnen und Schüler anhand ihnen bekannter technischer Gegenstände die Gemeinsamkeiten, Unterschiede und Besonderheiten der jeweiligen analogen und digitalen Version. Nach dieser ersten grundlegenden Einordnung des digitalen Prinzips wenden die Schülerinnen und Schüler das Binäre als Zahlensystem mit arithmetischen und logischen Operationen an und codieren Zeichen binär.

Zum Abschluss soll der grundlegende Aufbau eines Rechnersystems im Sinne der Von-Neumann-Architektur erarbeitet werden.

Zeitbedarf: 8 Stunden

Sequenzierung des Unterrichtsvorhabens:

| Unterrichtssequenzen | Zu entwickelnde Kompetenzen | Beispiele, Medien, Materialien |
|---|---|---|
| 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 | Die Schülerinnen und Schüler <ul style="list-style-type: none">• 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),• beschreiben und erläutern den strukturellen Aufbau und die Arbeitswei- | <i>Die digitale Welt 001 – Von analog zu digital</i> |
| 2. Der Umgang mit binärer Codierung von Informationen (a) Das binäre (und hexadezimale) Zah- | | <i>Die digitale Welt 001 – Von analog zu digital</i> |

| | | |
|--|--|--|
| <p>lensystem (b) Binäre Informationsspeicherung (c) Binäre Verschlüsselung (d) Implementation eines Binärumrechners</p> | <p>se singulärer Rechner am Beispiel der „Von-NeumannArchitektur“ (A),</p> <ul style="list-style-type: none"> • nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K), • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I). | |
| <p>3. Aufbau informatischer Systeme (a) Identifikation des EVA-Prinzips als grundlegende Arbeitsweise informatischer Systemen (b) Nachvollziehen der von-Neumann-Architektur als relevantes Modell der Umsetzung des EVA-Prinzips</p> | | <p><i>Die digitale Welt 001 – Von analog zu digital</i></p> |

Unterrichtsvorhaben EF-V

Thema: Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand lebensnaher Anforderungsbeispiele

Leitfrage: *Wie werden realistische Systeme anforderungsspezifisch reduziert, als Entwurf modelliert und implementiert? Wie kommunizieren Objekte und wie wird dieses dargestellt und realisiert?*

Vorhabenbezogene Konkretisierung:

Das Unterrichtsvorhaben hat die Entwicklung von Objekt- und Klassenbeziehungen zum Schwerpunkt. Dazu werden, ausgehend von der Realität, über Objektidentifizierung und Entwurf bis hin zur Implementation kleine Softwareprodukte in Teilen oder ganzheitlich erstellt.

Zuerst identifizieren die Schülerinnen und Schüler Objekte und stellen diese dar. Aus diesen Objekten werden Klassen und ihre Beziehungen in Entwurfsdiagrammen erstellt.

Nach diesem ersten Modellierungsschritt werden über Klassendokumentationen und der Darstellung von Objektkommunikationen anhand von Sequenzdiagrammen Implementationsdiagramme entwickelt. Danach werden die Implementationsdiagramme unter Berücksichtigung der Klassendokumentationen in Javaklassen programmiert. In einem letzten Schritt wird das Konzept der Vererbung sowie seiner Vorteile erarbeitet.

Schließlich sind die Schülerinnen und Schüler in der Lage, eigene kleine Softwareprojekte zu entwickeln. Ausgehend von der Dekonstruktion und Erweiterung eines Spiels wird ein weiteres Projekt von Grund auf modelliert und implementiert. Dabei können arbeitsteilige Vorgehensweisen zum Einsatz kommen. In diesem Zusammenhang wird auch das Erstellen von graphischen Benutzeroberflächen eingeführt.

Zeitbedarf: 20 Stunden

Sequenzierung des Unterrichtsvorhabens:

| Unterrichtssequenzen | Zu entwickelnde Kompetenzen | Beispiele, Medien, Materialien |
|---|---|--|
| 1. Umsetzung von Anforderungen in Entwurfsdiagramme (a) Aus Anforderungsbeschreibungen werden Objekte mit ihren Eigenschaften identifiziert | Die Schülerinnen und Schüler <ul style="list-style-type: none">analysieren und erläutern eine objektorientierte Modellierung (A),stellen die Kommunikation zwischen Objek- | Kapitel 5 - Klassenentwurf 5.1 Von der Realität zum Programm 5.2 Objekte 5.3 Klassen und Beziehungen entwerfen |

| | | |
|--|---|---|
| <p>(b) Gleichartige Objekte werden in Klassen (Entwurf) zusammengefasst und um Datentypen und Methoden erweitert</p> | <p>ten grafisch dar (M),</p> <ul style="list-style-type: none"> • ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), | |
| <p>2. Implementationsdiagramme als erster Schritt der Programmierung</p> <p>(a) Erweiterung des Entwurfsdiagramms um Konstruktoren und get- und set-Methoden</p> <p>(b) Festlegung von Datentypen in Java, sowie von Rückgaben und Parametern</p> <p>(c) Entwicklung von Klassendokumentationen</p> <p>(d) Erstellung von Sequenzdiagrammen als Vorbereitung für die Programmierung</p> | <ul style="list-style-type: none"> • 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), • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I), • testen Programme schrittweise anhand von Beispielen (I), | <p>Kapitel 5 - Klassenentwurf 5.4 Klassen und Beziehungen implementieren</p> |
| <p>3. Programmierung anhand der Dokumentation und des Implementations- und Sequenzdiagrammes</p> <p>(a) Klassen werden in Java-Quellcode umgesetzt.</p> <p>(b) Das Geheimnisprinzip wird umgesetzt.</p> <p>(c) Einzelne Klassen und das Gesamtsystem werden anhand der Anforderungen und Dokumentationen auf ihre Korrektheit überprüft.</p> | <ul style="list-style-type: none"> • 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). | <p>Kapitel 5 - Klassenentwurf 5.4 Klassen und Beziehungen implementieren</p> |
| | | |

| | | |
|--|--|--|
| <p>4. Vererbungsbeziehungen</p> <ul style="list-style-type: none"> (a) Das Grundprinzip der Vererbung wird erarbeitet. (b) Die Vorteile der Vererbungsbeziehungen (c) Vererbung wird implementiert | | <p>Kapitel 5 - Klassenentwurf 5.5 Vererbung</p> |
| <p>5. Softwareprojekt</p> <ul style="list-style-type: none"> (a) Analyse und Dekonstruktion eines Spiels (Modelle, Quelltexte) (b) Erweiterung des Spiels um weitere Funktionalitäten (c) Modellierung eines Spiels aufgrund einer Anforderungsbeschreibung, inklusive einer grafischen Benutzeroberfläche (d) (arbeitsteilige) Implementation des Spiels | | <p>Kapitel 7 - Softwareprojekte 7.1 Softwareentwicklung 7.2 Oberflächen</p> |

Unterrichtsvorhaben EF-VI

Thema: Such- und Sortieralgorithmen

Leitfrage: *Wie können Objekte bzw. Daten effizient gesucht und sortiert werden?*

Vorhabenbezogene Konkretisierung:

Dieses Unterrichtsvorhaben beschäftigt sich mit der Erarbeitung von Such- und Sortieralgorithmen. Der Schwerpunkt des Vorhabens liegt dabei auf den Algorithmen selbst und nicht auf deren Implementierung in einer Programmiersprache, auf die in diesem Vorhaben vollständig verzichtet werden soll.

Zunächst lernen die Schülerinnen und Schüler das Feld als eine erste Datensammlung kennen. Alle Algorithmen dieses Kapitels arbeiten auf einem Feld. Die Schülerinnen und Schüler lernen zunächst Strategien des Suchens (lineare Suche, binäre Suche, Hashing) und dann des Sortierens (Selection Sort, Insertion Sort, Bubble Sort) kennen.

Daran anschließend lernen die Schülerinnen und Schüler zunächst Strategien des Suchens (lineare Suche, binäre Suche, Hashing) und dann des Sortierens (Selection Sort, Insertion Sort, Bubble Sort) kennen. Die Projekteinstiege dienen dazu, die jeweiligen Strategien handlungsorientiert zu erkunden und intuitive Effizienzbetrachtungen der Suchalgorithmen vorzunehmen.

Schließlich wird die Effizienz unterschiedlicher Sortierverfahren beurteilt.

Zeitbedarf: 9 Stunden

Sequenzierung des Unterrichtsvorhabens:

| Unterrichtssequenzen | Zu entwickelnde Kompetenzen | Beispiele, Medien, Materialien |
|--|---|---|
| <p>1. Modellierung und Implementation von Datenansammlungen</p> <p>(a) Modellierung von Attributen als Felder (b) Deklaration, Instanziierung und Zugriffe auf ein Feld</p> | <p>Die Schülerinnen und Schüler</p> <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). | <p>Kapitel 6 - Suchen und Sortieren 6.1 Speichern mit Struktur – Arrays</p> |
| <p>2. Explorative Erarbeitung von Suchverfahren</p> <p>(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</p> | | <p>Kapitel 6 - Suchen und Sortieren Projekteinstieg 1: Suchen 6.2 Suchen mit System Lineare Suche Binäre Suche Hashing</p> |
| <p>3. Systematisierung von Algorithmen und Effizienzbetrachtungen</p> <p>(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) Analyse eines weiteren Sortieralgorithmus (sofern nicht in (a) bereits geschehen)</p> | | <p>Kapitel 6 - Suchen und Sortieren Projekteinstieg 2: Sortieren 6.3 Ordnung ist das halbe Leben!? – Sortieren Sortieren Selection Sort Insertion Sort Bubble Sort</p> |

Unterrichtsvorhaben EF-VII

Thema: Leben in der digitalen Welt – Immer mehr Möglichkeiten und immer mehr Gefahren!?

Leitfrage: *Welche Entwicklungen, Ideen und Erfindungen haben zur heutigen Informatik geführt? Welche Auswirkungen hat die Informatik für das Leben des modernen Menschen?*

Vorhabenbezogene Konkretisierung:

Das Unterrichtsvorhaben stellt die verschiedenen Entwicklungsstränge der Informatik in den Fokus. Darüber hinaus wird beispielhaft analysiert und bewertet, welche Möglichkeiten und Gefahren die moderne Informationsverarbeitung mit sich bringt.

Im ersten Schritt des Unterrichtsvorhabens wird anhand von Themenkomplexen entscheidende Entwicklungen der Informatik erarbeitet. Dabei werden auch übergeordnete Tendenzen identifiziert.

Ausgehend von dieser Betrachtung kann die aktuelle Informatik hinsichtlich ihrer Leistungsfähigkeit analysiert werden. Dabei soll herausgestellt werden, welche positiven und negativen Folgen Informatiksysteme mit sich bringen können.

Zeitbedarf: 5 Stunden

Sequenzierung des Unterrichtsvorhabens:

| Unterrichtssequenzen | Zu entwickelnde Kompetenzen | Beispiele, Medien, Materialien |
|--|---|--|
| 1. Schriftzeichen, Rechenmaschine, Computer (a) Anhand von Schwerpunkten, wie z.B. Datenspeicherung, Maschinen, Vernetzung sollen wichtige Entwicklungen der Informatik vorgestellt werden. (b) Anhand der unterschiedlichen Schwerpunkte sollen universelle Ten- | Die Schülerinnen und Schüler <ul style="list-style-type: none">• bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A),• erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (A), | <i>Die digitale Welt 100 – Von der Schrift zum Smartphone</i> |

| | | |
|--|---|--|
| <p>denzen der Entwicklung der Informationsverarbeitung erarbeitet werden</p> | <ul style="list-style-type: none"> • nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K) | |
| <p>2. Die Informationsverarbeitung und ihre Möglichkeiten und Gefahren</p> <p>(a) Ausgehend von 1. werden Tendenzen der Entwicklung der Informatik erarbeitet</p> <p>(b) Informatik wird als Hilfswissenschaft klassifiziert, die weit über ihren originären Bereich hinaus Effizienz- und Leistungssteigerungen erzeugt</p> <p>(c) Anhand von Fallbeispielen werden technische und organisatorische Vorteile, sowie deren datenschutzrechtlichen Nachteile betrachtet.</p> | | <p><i>Die digitale Welt 100 – Von der Schrift zum Smartphone</i></p> |

II) Qualifikationsphase

Die folgenden Kompetenzen aus dem Bereich *Kommunizieren und Kooperieren* werden in allen Unterrichtsvorhaben der Qualifikationsphase vertieft und sollen aus Gründen der Lesbarkeit nicht in jedem Unterrichtsvorhaben separat aufgeführt werden:

Die Schülerinnen und Schüler

- verwenden die Fachsprache bei der Kommunikation über informatische Sachverhalte (K),
- nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung von Dateien unter Berücksichtigung der Rechteverwaltung (K),
- organisieren und koordinieren kooperatives und eigenverantwortliches Arbeiten (K),
- strukturieren den Arbeitsprozess, vereinbaren Schnittstellen und führen Ergebnisse zusammen (K),
- beurteilen Arbeitsorganisation, Arbeitsabläufe und Ergebnisse (K),
- präsentieren Arbeitsabläufe und -ergebnisse adressatengerecht (K).

Unterrichtsvorhaben Q1-I:

Thema: Wiederholung der objektorientierten Modellierung und Programmierung

Leitfragen: *Wie modelliert und implementiert man zu einer Problemstellung in einem geeigneten Anwendungskontext Java-Klassen inklusive ihrer Attribute, Methoden und Beziehungen? Wie kann man die Modellierung und die Funktionsweise der Anwendung grafisch darstellen?*

Vorhabenbezogenen Konkretisierung:

Zu einer Problemstellung in einem Anwendungskontext soll eine Java-Anwendung entwickelt werden. Die Problemstellung soll so gewählt sein, dass für diese Anwendung die Verwendung einer abstrakten Oberklasse als Generalisierung verschiedener Unterklassen sinnvoll erscheint und eine Klasse durch eine Unterklasse spezialisiert werden kann. Um die Aufgabe einzugrenzen, können (nach der ersten Problemanalyse) einige Teile (Modellierungen oder Teile von Java-Klassen) vorgegeben werden.

Die Schülerinnen und Schülern erläutern und modifizieren den ersten Entwurf und modellieren sowie implementieren weitere Klassen und Methoden für eine entsprechende Anwendung. Klassen und ihre Beziehungen werden in einem Implementationsdiagramm dargestellt. Da-

bei werden Sichtbarkeitsbereiche zugeordnet. Exemplarisch wird eine Klasse dokumentiert. Der Nachrichtenaustausch zwischen verschiedenen Objekten wird verdeutlicht, indem die Kommunikation zwischen zwei ausgewählten Objekten grafisch dargestellt wird. In diesem Zusammenhang wird das Nachrichtenkonzept der objektorientierten Programmierung wiederholt.

Zeitbedarf: 8 Stunden

Sequenzierung des Unterrichtsvorhabens:

| Unterrichtssequenzen | Zu entwickelnde Kompetenzen | Beispiele, Medien, Materialien |
|--|---|--|
| <p>1. Wiederholung und Erweiterung der objektorientierten Modellierung und Programmierung durch Analyse und Erweiterung eines kontextbezogenen Beispiels</p> <p>(a) Analyse der Problemstellung (b) Analyse der Modellierung (Implementationsdiagramm) (c) Erweiterung der Modellierung im Implementationsdiagramm (Vererbung, abstrakte Klasse) (d) Kommunikation zwischen mindestens zwei Objekten (grafische Darstellung) (e) Dokumentation von Klassen (f) Implementierung der Anwendung oder von Teilen der Anwendung</p> | <p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • analysieren und erläutern objektorientierte Modellierungen (A), • beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), • modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M), • ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M), • modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M), • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I), • nutzen die Syntax und Semantik einer Programmiersprache bei der Imple- | <p>Kapitel 2 – Konzepte des objektorientierten Modellierens</p> <p>Projekteinstieg 2.1 Die Welt ist voller Objekte 2.2 Gut geplant – Klassenentwurf 2.3 Hierarchien machen's einfacher – Vererbung</p> |

| | | |
|--|---|--|
| | <p>mentierung und zur Analyse von Programmen (I),</p> <ul style="list-style-type: none">• wenden eine didaktisch orientierte Entwicklungsumgebung zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an (I),• interpretieren Fehlermeldungen und korrigieren den Quellcode (I),• stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D),• dokumentieren Klassen (D),• stellen die Kommunikation zwischen Objekten grafisch dar (D). | |
|--|---|--|

Unterrichtsvorhaben Q1-II:

Thema: Modellierung und Implementierung von Anwendungen mit dynamischen, linearen Datenstrukturen

Leitfrage: *Wie können beliebig viele linear angeordnete Daten im Anwendungskontext verwaltet werden?*

Vorhabenbezogene Konkretisierung:

Nach Analyse einer Problemstellung in einem geeigneten Anwendungskontext, in dem Daten nach dem First-In-First-Out-Prinzip verwaltet werden, werden der Aufbau von Schlangen am Beispiel dargestellt und die Operationen der Klasse Queue erläutert. Anschließend werden für die Anwendung notwendige Klassen modelliert und implementiert. Eine Klasse für eine den Anforderungen der Anwendung entsprechende Oberfläche sowie die Klasse Queue wird dabei von der Lehrkraft vorgegeben. Anschließend wird die Anwendung modifiziert, um den Umgang mit der Datenstruktur zu üben. Anhand einer Anwendung, in der Daten nach dem Last-In-First-Out-Prinzip verwaltet werden, werden Unterschiede zwischen den Datenstrukturen Schlange und Stapel erarbeitet. Um einfacher an Objekte zu gelangen, die zwischen anderen gespeichert sind, wird die Klasse List eingeführt und in einem Anwendungskontext verwendet. In mindestens einem weiteren Anwendungskontext wird die Verwaltung von Daten in Schlangen, Stapeln oder Listen vertieft. Modellierungen werden dabei in Entwurfs- und Implementationsdiagrammen dargestellt.

Zeitbedarf: 15 Stunden

Sequenzierung des Unterrichtsvorhabens:

| Unterrichtssequenzen | Zu entwickelnde Kompetenzen | Beispiele, Medien, Materialien |
|---|---|---|
| <p>1. Die Datenstruktur Schlange im Anwendungskontext unter Nutzung der Klasse Queue</p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <p>(b) Erarbeitung der Funktionalität der Klasse Queue</p> <p>(c) Modellierung und Implementierung der Anwendung unter Verwendung eines oder mehrerer Objekte der Klasse</p> | <p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none">• erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A),• analysieren und erläutern Algorithmen und Programme (A),• beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),• ordnen Attributen, Parametern und Rück- | <p>Kapitel 3 – Lineare Datenstrukturen</p> <p>Projekteinstieg – Wartende Helden</p> <p>3.3 Wer zuerst kommt, ... - Schlangen</p> |

| | | |
|---|--|---|
| Queue | | |
| <p>2. Die Datenstruktur Stapel im Anwendungskontext unter Nutzung der Klasse Stack</p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <p>(b) Erarbeitung der Funktionalität der Klasse Stack</p> <p>(c) Modellierung und Implementierung der Anwendung unter Verwendung eines oder mehrerer Objekte der Klasse Stack</p> | <p>gaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nicht-lineare Datensammlungen zu (M),</p> <ul style="list-style-type: none"> • ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), • modifizieren Algorithmen und Programme (I), • implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), • nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I), • testen Programme systematisch anhand von Beispielen (I), • stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D). | <p>Kapitel 3 – Lineare Datenstrukturen</p> <p>3.4 Daten gut abgelegt – Stapel</p> <p>Mögliche Projekte:</p> <ul style="list-style-type: none"> • Palindromprüfer • Terminterpreter |
| <p>3. Die Datenstruktur lineare Liste im Anwendungskontext unter Nutzung der Klasse List</p> <p>(a) Erarbeitung der Vorteile der Klasse List im Gegensatz zu den bereits bekannten linearen Strukturen</p> <p>(b) Modellierung und Implementierung einer kontextbezogenen Anwendung unter Verwendung der Klasse List.</p> | | <p>Kapitel 3 – Lineare Datenstrukturen</p> <p>3.5 Flexibel für alle Fälle – lineare Listen</p> <p>Mögliche Projekte:</p> <ul style="list-style-type: none"> • Ranglisten bei Sportereignissen • Vokabeltrainer |

Unterrichtsvorhaben Q1-III:

Thema: Suchen und Sortieren auf linearen Datenstrukturen

Leitfrage: *Wie kann man gespeicherte Informationen günstig (wieder-)finden?*

Vorhabenbezogene Konkretisierung:

In einem Anwendungskontext werden zunächst Informationen in einer linearen Liste bzw. einem Feld gesucht. Hierzu werden Verfahren entwickelt und implementiert bzw. analysiert und erläutert, wobei neben einem iterativen auch ein rekursives Verfahren thematisiert wird und mindestens ein Verfahren selbst entwickelt und implementiert wird. Die verschiedenen Verfahren werden hinsichtlich Speicherbedarf und Zahl der Vergleichsoperationen miteinander verglichen.

Anschließend werden Sortierverfahren entwickelt und implementiert (ebenfalls für lineare Listen und Felder). Hierbei soll auch ein rekursives Sortierverfahren entwickelt werden. Die Implementationen von Quicksort sowie dem Sortieren durch Einfügen werden analysiert und erläutert. Falls diese Verfahren vorher schon entdeckt wurden, sollen sie hier wiedererkannt werden. Die rekursive Abarbeitung eines Methodenaufrufs von Quicksort wird grafisch dargestellt.

Abschließend werden verschiedene Sortierverfahren hinsichtlich der Anzahl der benötigten Vergleichsoperationen und des Speicherbedarfs beurteilt.

Zeitbedarf: 12 Stunden

Sequenzierung des Unterrichtsvorhabens:

| Unterrichtssequenzen | Zu entwickelnde Kompetenzen | Beispiele, Medien, Materialien |
|---|---|--|
| 1. Suchen von Daten in Listen und Arrays (a) Lineare Suche in Listen und in Arrays (b) Binäre Suche in Arrays als Beispiel für rekursives Problemlösen (c) Untersuchung der beiden Suchverfahren hinsichtlich ihrer Effizienz (Laufzeitverhalten, Speicherbedarf) | Die Schülerinnen und Schüler <ul style="list-style-type: none">• analysieren und erläutern Algorithmen und Programme (A),• beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),• beurteilen die Effizienz von Algorithmen | Kapitel 4 – Algorithmen Projekteinstieg 1: Suchen 4.2 Suchen mit Systemen Mögliche Projekte: <ul style="list-style-type: none">• Karteiverwaltung• Adressbuch |

| | | |
|--|---|---|
| <p>2. Sortieren in Listen und Arrays - Entwicklung und Implementierung von iterativen und rekursiven Sortierverfahren</p> <p>(a) Entwicklung und Implementierung eines einfachen Sortierverfahrens für eine Liste</p> <p>(b) Implementierung eines einfachen Sortierverfahrens für ein Feld</p> <p>(c) Entwicklung eines rekursiven Sortierverfahrens für ein Feld (z.B. Sortieren durch Mischen)</p> | <p>unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (A),</p> <ul style="list-style-type: none"> • entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“ (M), • modifizieren Algorithmen und Programme (I), • implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), • implementieren und erläutern iterative und rekursive Such- und Sortierverfahren (I), | <p>Kapitel 4 – Algorithmen</p> <p>Projekteinstieg 2: Sortieren</p> <p>4.3 Ordnung ist das halbe Leben!? - Sortieren</p> <p>4.4 Teile die Arbeit – Rekursive Algorithmen</p> <p>Mögliche Projekte:</p> <ul style="list-style-type: none"> • Karteiverwaltung • Adressbuch |
| <p>3. Untersuchung der Effizienz der Sortierverfahren „Sortieren durch direktes Einfügen“ und „Quicksort“ auf linearen Listen</p> <p>(a) Grafische Veranschaulichung der Sortierverfahren</p> <p>(b) Untersuchung der Anzahl der Vergleichsoperationen und des Speicherbedarf bei beiden Sortierverfahren</p> <p>(c) Beurteilung der Effizienz der beiden Sortierverfahren</p> | <ul style="list-style-type: none"> • nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I), • testen Programme systematisch anhand von Beispielen (I), • stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D). | <p>Mögliche Projekte:</p> <ul style="list-style-type: none"> • Karteiverwaltung • Adressbuch |

Unterrichtsvorhaben Q1-IV:

Thema: Modellierung und Implementierung von Anwendungen mit dynamischen, nichtlinearen Datenstrukturen

Leitfragen: *Wie können Daten im Anwendungskontext mit Hilfe binärer Baumstrukturen verwaltet werden? Wie kann dabei der rekursive Aufbau der Baumstruktur genutzt werden? Welche Vor- und Nachteile haben Suchbäume für die geordnete Verwaltung von Daten?*

Vorhabenbezogene Konkretisierung:

Anhand von Beispielen für Baumstrukturen werden grundlegende Begriffe eingeführt und der rekursive Aufbau binärer Bäume dargestellt. Anschließend werden für eine Problemstellung in einem der Anwendungskontexte Klassen modelliert und implementiert. Dabei werden die Operationen der Datenstruktur Binärbaum thematisiert und die entsprechende Klasse BinaryTree (der Materialien für das Zentralabitur in NRW) der Vorgaben für das Zentralabitur NRW verwendet. Klassen und ihre Beziehungen werden in Entwurfs- und Implementationsdiagrammen dargestellt. Die Funktionsweise von Methoden wird anhand grafischer Darstellungen von Binärbäumen erläutert.

Unter anderem sollen die verschiedenen Baumtraversierungen (Pre-, Post- und Inorder) implementiert werden. Unterschiede bezüglich der Möglichkeit, den Baum anhand der Ausgabe der Baum Inhalte via Pre-, In- oder Postorder-Traversierung zu rekonstruieren, werden dabei ebenfalls angesprochen, indem die fehlende Umkehrbarkeit der Zuordnung Binärbaum → Inorder-Ausgabe an einem Beispiel verdeutlicht wird.

Eine Tiefensuche wird verwendet, um einen in der Baumstruktur gespeicherten Inhalt zu suchen.

Zu einer Problemstellung in einem entsprechenden Anwendungskontext werden die Operationen der Datenstruktur Suchbaum thematisiert und unter der Verwendung der Klasse BinarySearchTree (der Materialien für das Zentralabitur in NRW) weitere Klassen oder Methoden in diesem Anwendungskontext modelliert und implementiert. Auch in diesem Kontext werden grafische Darstellungen der Bäume verwendet.

Die Verwendung von binären Bäumen und Suchbäumen wird anhand weiterer Problemstellungen oder anderen Kontexten weiter geübt.

Zeitbedarf: 12 Stunden

Sequenzierung des Unterrichtsvorhabens:

| Unterrichtssequenzen | Zu entwickelnde Kompetenzen | Beispiele, Medien, Materialien |
|----------------------|-----------------------------|--------------------------------|
| | | |

| | | |
|--|---|--|
| <p>1. Analyse von Baumstrukturen in verschiedenen Kontexten</p> <p>(a) Grundlegende Begriffe (Grad, Tiefe, Höhe, Blatt, Inhalt, Teilbaum, Ebene, Vollständigkeit)</p> <p>(b) Aufbau und Darstellung von binären Bäumen anhand von Baumstrukturen in verschiedenen Kontexten</p> | <p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A), • analysieren und erläutern Algorithmen und Programme (A), • beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), • ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M), • modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M), • verwenden bei der Modellierung geeigneter Problemstellungen die Möglichkeiten der Polymorphie (M), • entwickeln iterative und rekursive Algorithmen unter Nutzung der Konstruktionsstrategien „Modularisierung“ und „Teilen und Herrschen“ (M), • implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), • modifizieren Algorithmen und Pro- | <p>Kapitel 5 – Nicht-lineare Datenstrukturen</p> <p>Projekteinstieg 1: Spielbäume</p> <p>Projekteinstieg 2: Binäre Suchbäume</p> <p>5.1 Spielen mit Struktur – Baumstrukturen</p> <p>5.2 Zwei Nachfolger sind genug! - Binäre Bäume</p> |
| <p>2. Die Datenstruktur Binärbaum im Anwendungskontext unter Nutzung der Klasse BinaryTree</p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen im Anwendungskontext</p> <p>(b) Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramms</p> <p>(c) Erarbeitung der Klasse BinaryTree und beispielhafte Anwendung der Operationen</p> <p>(d) Implementierung der Anwendung oder von Teilen der Anwendung</p> <p>(e) Traversierung eines Binärbaums im Pre-, In- und Postorderdurchlauf</p> | <p>(M),</p> <p>• ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M),</p> <p>• modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M),</p> <p>• verwenden bei der Modellierung geeigneter Problemstellungen die Möglichkeiten der Polymorphie (M),</p> <p>• entwickeln iterative und rekursive Algorithmen unter Nutzung der Konstruktionsstrategien „Modularisierung“ und „Teilen und Herrschen“ (M),</p> <p>• implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I),</p> <p>• modifizieren Algorithmen und Pro-</p> | <p>Kapitel 5 – Nicht-lineare Datenstrukturen</p> <p>5.2 Zwei Nachfolger sind genug! - Binäre Bäume</p> <p>Mögliche Projekte:</p> <ul style="list-style-type: none"> • Ahnenbaum • Morsecodierung |
| <p>3. Die Datenstruktur binärer Suchbaum im Anwendungskontext unter Verwendung der Klasse BinarySearchTree</p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften</p> | <p>• modifizieren Algorithmen und Pro-</p> | <p>Kapitel 5 – Nicht-lineare Datenstrukturen</p> <p>5.3 Wer Ordnung hält, spart Zeit beim Suchen – Binäre Suchbäume</p> |

| | | |
|---|--|---|
| <p>ten und Operationen</p> <p>(b) Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramm, grafische Darstellung eines binären Suchbaums und Erarbeitung der Struktureigenschaften</p> <p>(c) Erarbeitung der Klasse BinarySearchTree und Einführung des Interface Item zur Realisierung einer geeigneten Ordnungsrelation</p> <p>(d) Implementierung der Anwendung oder von Teilen der Anwendung inklusive einer sortierten Ausgabe des Baums</p> | <p>gramme (I),</p> <ul style="list-style-type: none"> • nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I), • testen Programme systematisch anhand von Beispielen (I), • stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D), • stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D). | |
| <p>4. Übung und Vertiefungen der Verwendung von Binärbäumen oder binären Suchbäumen anhand weiterer Problemstellungen</p> | | <p>Mögliche Projekte:</p> <ul style="list-style-type: none"> • Ahnenbaum • Morsecodierung |

Unterrichtsvorhaben Q1-V:

Thema: Graphen

Leitfragen: *Wie arbeiten Navigationsgeräte? Wie können Wegenetze, Beziehungsgeflechte oder Optimierungsprobleme dargestellt werden? Wie kommt die Internetseite auf den Bildschirm, wenn die Adresse eingegeben wird?*

Vorhabenbezogene Konkretisierung:

Anhand eines Graphen für ein Verkehrsnetz werden die Begriffe Knoten, Kanten, gerichteter Graph, ungerichteter Graph, Eingangsgrad eines Knotens und Ausgangsgrad eines Knotens herausgearbeitet. Als nächstes wird betrachtet, wie man einen Graphen speichern kann. Dabei kommen drei Varianten in Frage:

- Kantenliste,
- Adjazenzliste und
- Adjazenzmatrix.

Für jede dieser Speichervarianten wird untersucht, wie die Laufzeit für die verschiedenen Methoden ist, die man auf einem Graphen ausführen kann. Dabei wird deutlich, dass je nach Verwendungszweck des Graphen eine bestimmte Art der Speicherung der Graphen die geeignete darstellt. Daraufhin werden die Tiefen- und Breitensuche als erste Traversierungsarten auf Graphen besprochen. Die Bestimmung von Abläufen kann gut anhand des topologischen Sortierens verdeutlicht werden; hierzu muss ein gerichteter azyklischer Graph vorliegen. Anhand von Flugrouten kann das Thema kürzeste Wege besprochen werden und der Algorithmus von Dijkstra erarbeitet werden. Eine weitere zu thematisierende Problematik ist die der minimalen Kosten beim Aufbau eines Netzes. Hier wird das Begriff des Minimalen Spannbauemes und der Algorithmus von Kruskal zu thematisieren sein

Zeitbedarf: 5 Stunden

Sequenzierung des Unterrichtsvorhabens:

| Unterrichtssequenzen | Zu entwickelnde Kompetenzen | Beispiele, Medien, Materialien |
|--|---|---|
| 1. Der Einsatz von Graphen im Kontext der Navigationsgeräte (a) Grundlegende Begriffe (Knoten, Kante, gerichteter Graph, ungerichteter | Die Schülerinnen und Schüler <ul style="list-style-type: none">• stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren | Kapitel 5 – Nicht-lineare Datenstrukturen 5.4 Navigieren mit Struktur - Graphen |

| | | |
|---|---|--|
| <p>Graph, Gewicht, Grad eines Knotens, Pfad oder Weg in einem Graphen) (b) Übersetzung eines Anwendungskontextes in einen Graphen.</p> | <p>Aufbau (D),</p> <ul style="list-style-type: none"> • stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D), • entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien »Modularisierung«, »Teile und Herrsche« und »Backtracking« (M), • implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), • erläutern Operationen dynamischer (linearer und nicht-linearer) Datenstrukturen (A), • implementieren Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (I), • beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (A). analysieren und erläutern objektorientierte Modellierungen (IF1, A) | |
| <p>2. Modellierung von Graphen (a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen im Anwendungskontext (b) Varianten der Speicherung eines Graphen. (c) Tiefen- und Breitensuche als Traversierungsarten auf Graphen</p> | | <p><i>Kapitel 5 – Nicht-lineare Datenstrukturen</i> 5.5 Modellierung von Graphen</p> |
| <p>3. Der Algorithmus von Dijkstra (a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen (b) Erarbeitung des Algorithmuses von Dijkstra (c) Durchspielen des Algorithmuses anhand ausgewählter Beispiele</p> | | <p><i>Kapitel 5 – Nicht-lineare Datenstrukturen</i> 5.6 Kürzester Weg – Der Algorithmus von Dijkstra</p> |

Unterrichtsvorhaben Q1-VI:

Thema: Vielfalt beim Einsatz von Klassen durch Abstrakte Klassen, Polymorphie und MVC

Leitfragen: *Wie können verschiedene Oberflächen für ein Projekt realisiert werden? Wie kann ein bereits bestehendes Projekt um eine weitere, andere Oberfläche ergänzt werden?*

Vorhabenbezogene Konkretisierung:

Mit dem Prinzip der Modularisierung respektive Trennung von Teilbereichen haben sich die Schülerinnen und Schülern im Unterrichtsvorhaben über die linearen Datenstrukturen beschäftigt. Dort wird die Struktur vom Inhalt getrennt. Auf einem höheren Niveau werden nun komplette Klassen Kategorien zugeordnet. Die Einordnung in Klassen, die die Welt modellieren, Klassen, die für die Interaktion mit dem Benutzer zuständig sind und Klassen, die das Bindeglied darstellen und die Programmlogik übernehmen, stellt eine solche konzeptionelle Modularisierung dar.

Generell bietet sich hierbei an, einen Ansatz zu wählen, bei dem verschiedene Benutzerschnittstellen von den Schülerinnen und Schülern entwickelt werden. Dies können verschiedene grafische Benutzeroberflächen oder aber auch eine Konsole sein, über die der Benutzer interagiert. So wird klar, dass die Programmlogik gleich bleibt, die Ausgabe aber verschieden – z. B. je nach System – gestaltet werden kann.

Zeitbedarf: 12 Stunden

Sequenzierung des Unterrichtsvorhabens:

| Unterrichtssequenzen | Zu entwickelnde Kompetenzen | Beispiele, Medien, Materialien |
|---|--|--|
| 1. Analyse eines bereits erstellten Programmes, das nicht konsequent nach dem MVC-Prinzip erstellt wurde (a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen im Anwendungskontext (b) Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramm | Die Schülerinnen und Schüler <ul style="list-style-type: none">• analysieren und erläutern objektorientierte Modellierungen (A) ,• implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (M),• analysieren und erläutern Algorithmen und Programme (A), | Bereits erarbeitetes Projekt – dabei bieten sich an: <ul style="list-style-type: none">• Vokabeltrainer,• Karteiverwaltung,• Adressbuch usw. |

| | | |
|--|--|---|
| <p>2. Entwicklung verschiedener Oberflächen zu einem bereits bestehenden Projekt</p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen im Anwendungskontext</p> <p>(b) Implementierung der Anwendung oder von Teilen der Anwendung</p> | <ul style="list-style-type: none"> • modifizieren Algorithmen und Programme (I) | <p>Bereits erarbeitetes Projekt – dabei bieten sich an:</p> <ul style="list-style-type: none"> • Vokabeltrainer, • Karteiverwaltung, • Adressbuch usw. |
|--|--|---|

Unterrichtsvorhaben Q1-VII:

Thema: Kommunikation in und Aufbau von Netzwerken

Leitfragen: *Welche Netzwerktypen gibt es? Wie sprechen Rechner mit einander?*

Vorhabenbezogene Konkretisierung:

Vernetzte Informatiksysteme bestimmen die heutige IT-Infrastruktur und damit haben sie, denkt man zum Beispiel an die zahlreichen Internetanwendungen, auch einen nicht unerheblichen Einfluss auf unsere Lebenswelt. Zunächst sollten die Netzwerktechnischen Grundlagen betrachtet werden. Dabei erscheint es sinnvoll folgende Aspekte zu betrachten:

- Definition eines Netzwerkes
- Netzwerktypen nach Ausdehnung (LAN, MAN, WAN, GAN)
- Peer-to-Peer vs. Client-Server-Netzwerk
- OSI-Schichtenmodell und die Funktion der Schichten
- Frames
- Netzwerktopologien (Ring-, Stern-, Bus-, Baum- und Maschen-Netz)
- Arten der Verkabelung
- Nachrichtentechnische Grundlagen (Digitalisierung, Modulationsverfahren, Übertragungsmedien)
- Übertragungsprotokolle (CSMA/CD und Token Passing)
- Netzwerkhardware

In Folge wird am Beispiel des TCP/IP-Protokolls der Protokollbegriff vertiefend betrachtet. Dies kann sinnvoller Weise in Form von Schülerreferaten erfolgen. Dabei erscheinen die folgenden Themen sinnvoll:

- Wesen eines Protokolls und Low-Level-Protokolle (MAC, LLC, SAP und SNAP)
- IP, ICMP, ARP, RARP, Routing-Protokolle und TCP
- Adressierung und Subnetze IPv4
- Dynamic Host Configuration Protokoll DHCP
- Routing Grundlagen / Routing Information Protokoll (RIP) / OSPF (Open Shortest Path First)
- Namensauflösung Host-Datei / Domain Name System (DNS)
- File Transfer Protokoll (FTP)
- Hypertext Transfer Protokoll (HTTP)
- E-Mail (SMTP/POP3/IMPAP4)
- Weiterentwicklung IPv6

Zeitbedarf: 15 Stunden

Sequenzierung des Unterrichtsvorhabens

| Unterrichtssequenzen | Zu entwickelnde Kompetenzen | Beispiele, Medien, Materialien |
|--|---|---|
| <p>1. Netzwerke (a) Definition eines Netzwerkes und Netzwerktypen (b) Protokolle (c) Client-Server-Netzwerke</p> | <p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • beschreiben und erläutern Netzwerk-Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken (A) • untersuchen und bewerten anhand von Fallbeispielen Auswirkungen des Einsatzes von Informatiksystemen sowie Aspekte der Sicherheit von Informatiksystemen, des Datenschutzes und des Urheberrechts (A) | <p>Informatik 3 – Kapitel 1 Projekteinstieg 1.1 Menschen kommunizieren – ohne und mit Technik 1.2 Ohne Protokoll läuft nichts – Netzwerke 1.3 Einer für alle – Client-Server-Netzwerke</p> |
| <p>2. Vertiefung des Protokollbegriffs an ausgewählten Anwendungen</p> | <ul style="list-style-type: none"> • untersuchen und bewerten Problemlagen, die sich aus dem Einsatz von Informatiksystemen ergeben, hinsichtlich rechtlicher Vorgaben, ethischer Aspekte und gesellschaftlicher Werte unter Berücksichtigung unterschiedlicher Interessenlagen (A) • analysieren und erläutern Algorithmen und Methoden zur Client-Server-Kommunikation (A), • analysieren und erläutern Protokolle zur Kommunikation in einem Client-Server-Netzwerk (A), • entwickeln und erweitern Protokolle zur Kommunikation in einem Client-Server-Netzwerk (M) | |

Unterrichtsvorhaben Q2-I:

Thema: Modellierung und Nutzung von relationalen Datenbanken in Anwendungskontexten

Leitfragen: *Wie können Fragestellungen mit Hilfe einer Datenbank beantwortet werden? Wie entwickelt man selbst eine Datenbank für einen Anwendungskontext?*

Vorhabenbezogene Konkretisierung:

Ausgehend von einer vorhandenen Datenbank entwickeln Schülerinnen und Schüler für sie relevante Fragestellungen, die mit dem vorhandenen Datenbestand beantwortet werden sollen. Zur Beantwortung dieser Fragestellungen wird die vorgegebene Datenbank von den Schülerinnen und Schülern analysiert und die notwendigen Grundbegriffe für Datenbanksysteme sowie die erforderlichen SQL-Abfragen werden erarbeitet.

In anderen Anwendungskontexten müssen Datenbanken erst noch entwickelt werden, um Daten zu speichern und Informationen für die Beantwortung von möglicherweise auftretenden Fragen zur Verfügung zu stellen. Dafür ermitteln Schülerinnen und Schüler in den Anwendungssituationen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten und stellen diese in Entity-Relationship-Modellen dar. Entity-Relationship-Modelle werden interpretiert und erläutert, modifiziert und in Datenbankschemata überführt. Mit Hilfe von SQL-Anweisungen können anschließend im Kontext relevante Informationen aus der Datenbank extrahiert werden.

Ein Entity-Relationship-Diagramm kann auch verwendet werden, um die Entitäten inklusive ihrer Attribute und Relationen in einem vorgegebenen Datenbankschema darzustellen.

An einem Beispiel wird verdeutlicht, dass in Datenbanken Redundanzen unerwünscht sind und Konsistenz gewährleistet sein sollte. Die 1. bis 3. Normalform wird als Gütekriterium für Datenbankentwürfe eingeführt. Datenbankschemata werden hinsichtlich der 1. bis 3. Normalform untersucht und (soweit nötig) normalisiert.

Zeitbedarf: 15 Stunden

Sequenzierung des Unterrichtsvorhabens

| Unterrichtssequenzen | Zu entwickelnde Kompetenzen | Beispiele, Medien, Materialien |
|---|---|--|
| <p>1. Nutzung von relationalen Datenbanken</p> <p>(a) Aufbau von Datenbanken und Grundbegriffe</p> <ul style="list-style-type: none"> Entwicklung von Fragestellungen zur vorhandenen Datenbank Analyse der Struktur der vorgegebenen Datenbank und Erarbeitung der Begriffe Tabelle, Attribut, Datensatz, Datentyp, Primärschlüssel, Fremdschlüssel, Datenbankschema <p>(b) SQL-Abfragen</p> <ul style="list-style-type: none"> Analyse vorgegebener SQL-Abfragen und Erarbeitung der Sprachelemente von SQL (SELECT (DISTINCT) ...FROM, WHERE, AND, OR, NOT) auf einer Tabelle Analyse und Erarbeitung von SQL-Abfragen auf einer und mehrerer Tabelle zur Beantwortung der Fragestellungen (JOIN, UNION, AS, GROUP BY, ORDER BY, ASC, DESC, COUNT, MAX, MIN, SUM, Arithmetische Operatoren: +, -, *, /, (...), Vergleichsoperatoren: =, <>, >, <, >=, <=, LIKE, BETWEEN, IN, IS NULL) <p>(c) Vertiefung an einem weiteren Datenbankbeispiel</p> | <p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> erläutern die Eigenschaften und den Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nutzung (A), analysieren und erläutern die Syntax und Semantik einer Datenbankabfrage (A), analysieren und erläutern eine Datenbankmodellierung (A), erläutern die Eigenschaften normalisierter Datenbankschemata (A), bestimmen Primär- und Sekundärschlüssel (M), ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten (M), modifizieren eine Datenbankmodellierung (M), modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema (M), bestimmen Primär- und Sekundärschlüssel (M), überführen Datenbankschemata in vorgegebene Normalformen (M), verwenden die Syntax und Semantik einer Datenbankabfragesprache, um Informationen aus einem Datenbanksystem zu extrahieren (I), | <p>Informatik 3 Kapitel 3 – Datenbanken Projekteinstieg 3.1 Wissen speichern und verwalten – Datenbanksysteme 3.2 Daten anordnen mit Tabellen 3.3 Operationen auf Tabellen 3.4 Komplexe Filter</p> |
| <p>2. Modellierung von relationalen Datenbanken</p> | | <p>Informatik 3 Kapitel 3 – Datenbanken</p> |

| | | |
|---|---|--|
| <p>(a) Entity-Relationship-Diagramm</p> <ul style="list-style-type: none"> • Ermittlung von Entitäten, zugehörigen Attributen, Relationen und Kardinalitäten in Anwendungssituationen und Modellierung eines Datenbankentwurfs in Form eines Entity-Relationship-Diagramms • Erläuterung und Modifizierung einer Datenbankmodellierung <p>(b) Entwicklung einer Datenbank aus einem Datenbankentwurf</p> <ul style="list-style-type: none"> • Modellierung eines relationalen Datenbankschemas zu einem Entity-Relationship-Diagramm inklusive der Bestimmung von Primär- und Sekundärschlüsseln <p>(c) Redundanz, Konsistenz und Normalformen</p> <ul style="list-style-type: none"> • Untersuchung einer Datenbank hinsichtlich Konsistenz und Redundanz in einer Anwendungssituation • Überprüfung von Datenbankschemata hinsichtlich der 1. bis 3. Normalform und Normalisierung (um Redundanzen zu vermeiden und Konsistenz zu gewährleisten) | <ul style="list-style-type: none"> • ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen (D), • stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten in einem Entity-Relationship-Diagramm grafisch dar (D), • überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D). | <p>3.5 Datenbankentwurf</p> <p>3.6 Umsetzung des ER-Modells</p> <p>3.7 Datenbanken verbessern durch Normalformen</p> <p>3.8 Training</p> |
|---|---|--|

Unterrichtsvorhaben Q2-II:

Thema: Endliche Automaten und formale Sprachen

Leitfragen: *Wie kann man (endliche) Automaten genau beschreiben? Wie können endliche Automaten (in alltäglichen Kontexten oder zu informatischen Problemstellungen) modelliert werden? Wie können Sprachen durch Grammatiken beschrieben werden? Welche Zusammenhänge gibt es zwischen formalen Sprachen, endlichen Automaten und regulären Grammatiken?*

Vorhabenbezogene Konkretisierung:

Anhand kontextbezogener Beispiele werden endliche Automaten entwickelt, untersucht und modifiziert. Dabei werden verschiedene Darstellungsformen für endliche Automaten ineinander überführt und die akzeptierten Sprachen endlicher Automaten ermittelt. An einem Beispiel wird ein nichtdeterministischer Akzeptor eingeführt als Alternative gegenüber einem entsprechenden deterministischen Akzeptor.

Anhand kontextbezogener Beispiele werden Grammatiken regulärer Sprachen entwickelt, untersucht und modifiziert. Der Zusammenhang zwischen regulären Grammatiken und endlichen Automaten wird verdeutlicht durch die Entwicklung von allgemeinen Verfahren zur Erstellung einer regulären Grammatik für die Sprache eines gegebenen endlichen Automaten bzw. zur Entwicklung eines endlichen Automaten, der genau die Sprache einer gegebenen regulären Grammatik akzeptiert.

Auch andere Grammatiken werden untersucht, entwickelt oder modifiziert. An einem Beispiel werden die Grenzen endlicher Automaten ausgelotet.

Zeitbedarf: 20 Stunden

Sequenzierung des Unterrichtsvorhabens:

| Unterrichtssequenzen | Zu entwickelnde Kompetenzen | Beispiele, Medien oder Materialien |
|--|---|---|
| <p>1. Endliche Automaten</p> <p>(a) Vom Automaten in den Schülerinnen und Schülern bekannten Kontexten zur formalen Beschreibung eines endlichen Automaten</p> <p>(b) Untersuchung, Darstellung und Entwicklung endlicher Automaten</p> | <p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • analysieren und erläutern die Eigenschaften endlicher Automaten einschließlich ihres Verhaltens auf bestimmte Eingaben (A), • analysieren und erläutern Grammatiken regulärer Sprachen (A), | <p>Informatik 3 Kapitel 4 – Endliche Automaten und Formale Sprachen Projekteinstieg 2 4.2 Automatentheorie</p> |
| <p>2. Untersuchung und Entwicklung von Grammatiken regulärer Sprachen</p> <p>(a) Erarbeitung der formalen Darstellung regulärer Grammatiken</p> <p>(b) Untersuchung, Modifikation und Entwicklung von Grammatiken</p> <p>(c) Entwicklung von endlichen Automaten zum Erkennen regulärer Sprachen die durch Grammatiken gegeben werden</p> <p>(d) Entwicklung regulärer Grammatiken zu endlichen Automaten</p> | <ul style="list-style-type: none"> • zeigen die Grenzen endlicher Automaten und regulärer Grammatiken im Anwendungszusammenhang auf (A), • ermitteln die formale Sprache, die durch eine Grammatik erzeugt wird (A), • entwickeln und modifizieren zu einer Problemstellung endliche Automaten (M), • entwickeln und modifizieren zu einer Problemstellung endliche Automaten (M), • entwickeln zur akzeptierten Sprache eines Automaten die zugehörige Grammatik (M), | <p>Informatik 3 Kapitel 4 – Endliche Automaten und Formale Sprachen Projekteinstieg 2 4.3 Formale Sprachen 4.4 Parser</p> |
| <p>3. Grenzen endlicher Automaten</p> | <ul style="list-style-type: none"> • entwickeln zur Grammatik einer regulären Sprache einen zugehörigen endlichen Automaten (M), • modifizieren Grammatiken regulärer Sprachen (M), • entwickeln zu einer regulären Sprache eine Grammatik, die die Sprache erzeugt (M), • stellen endliche Automaten in Tabellen oder Graphen dar und überführen sie in | <p><i>Beispiele:</i> Klammerausdrücke, $a^n b^n$ im Vergleich zu $(ab)^n$</p> |

| | | |
|--|---|--|
| | <p>die jeweils andere Darstellungsform (D),</p> <ul style="list-style-type: none">• ermitteln die Sprache, die ein endlicher Automat akzeptiert (D).• beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken (D). | |
|--|---|--|

Unterrichtsvorhaben Q2-III:

Thema: Prinzipielle Arbeitsweise eines Computers und Grenzen der Automatisierbarkeit

Leitfragen: Was sind die strukturellen Hauptbestandteile eines Computers und wie kann man sich die Ausführung eines maschinennahen Programms mit diesen Komponenten vorstellen? Welche Möglichkeiten bieten Informatiksysteme und wo liegen ihre Grenzen?

Vorhabenbezogene Konkretisierung:

Anhand einer von-Neumann-Architektur und einem maschinennahen Programm wird die prinzipielle Arbeitsweise von Computern verdeutlicht.

Ausgehend von den prinzipiellen Grenzen endlicher Automaten liegt die Frage nach den Grenzen von Computern bzw. nach Grenzen der Automatisierbarkeit nahe. Mit Hilfe einer entsprechenden Java-Methode wird plausibel, dass es unmöglich ist, ein Informatiksystem zu entwickeln, das für jedes beliebige Computerprogramm und jede beliebige Eingabe entscheidet ob das Programm mit der Eingabe terminiert oder nicht (Halteproblem). Anschließend werden Vor- und Nachteile der Grenzen der Automatisierbarkeit angesprochen und der Einsatz von Informatiksystemen hinsichtlich prinzipieller Möglichkeiten und prinzipieller Grenzen beurteilt.

Zeitbedarf: 12 Stunden

Sequenzierung des Unterrichtsvorhabens:

| Unterrichtssequenzen | Zu entwickelnde Kompetenzen | Beispiele, Medien oder Materialien |
|---|--|---|
| <p>1. Von-Neumann-Architektur und die Ausführung maschinennaher Programme</p> <p>a) prinzipieller Aufbau einer von Neumann-Architektur mit CPU, Rechenwerk, Steuerwerk, Register und Hauptspeicher</p> <p>b) einige maschinennahe Befehle und ihre Repräsentation in einem Binär-Code, der in einem Register gespeichert werden kann</p> | <p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none">• erläutern die Ausführung eines einfachen maschinennahen Programms sowie die Datenspeicherung auf einer „Von-Neumann-Architektur“ (A),• untersuchen und beurteilen Grenzen des Problemlösens mit Informatiksystemen (A). | <p><i>Beispiel:</i> Addition von 4 zu einer eingegebenen Zahl mit einem Rechnermodell</p> <p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.3 –Von-Neumann-Architektur und maschinennahe Programmierung</p> |

| | | |
|---|--|---|
| <p>c) Analyse und Erläuterung der Funktionsweise eines einfachen maschinennahen Programms</p> | | |
| <p>2. Grenzen der Automatisierbarkeit</p> <p>a) Vorstellung des Halteproblems</p> <p>b) Unlösbarkeit des Halteproblems</p> <p>c) Beurteilung des Einsatzes von Informatiksystemen hinsichtlich prinzipieller Möglichkeiten und prinzipieller Grenzen</p> | | <p><i>Beispiel:</i> Halteproblem</p> <p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.3 - Halteproblem</p> |

Unterrichtsvorhaben Q2-IV:

Thema: Sicherheit in der Informatik: Verschlüsselung und ihre Folgen

Leitfragen: *Wie sicher ist „das Internet“? Wo liegen die Angriffspunkte der Verschlüsselungsverfahren? Gibt es eine „sichere“ Verschlüsselung?*

Vorhabenbezogene Konkretisierung:

Im Zeitalter vernetzter Informationssysteme ist der sichere Austausch von Daten über ein Netzwerk ein wichtiger Aspekt. Man denke an Homebanking-Anwendungen oder das Versenden von E-Mails die persönliche Daten beinhalten und für Dritte nicht lesbar sein sollen. Bei der Betrachtung von Verschlüsselungsverfahren sollen symmetrische und asymmetrische, mono- und polyalphabetische Verschlüsselungsverfahren thematisiert werden.

In diesem Zusammenhang scheint eine genetische Vorgehensweise sinnvoll, auch wenn diese historischen Verfahren heute aus Sicherheitsgründen in der praktischen Anwendung nicht mehr zum Einsatz kommen. Ein möglicher Ablauf könnte dabei folgende Verfahren umfassen:

- Caesar-Chiffre
- Vigenere-Verfahren
- RSA-Verschlüsselung
- mit PGP/GPG verschlüsselte Daten

Das Caesar- und das Vigenere-Verfahren können auf jeden Fall implementiert werden. Die Implementation von RSA sollte erfolgen, wenn die mathematischen und implementationstechnischen Fähigkeiten der Schüler des Grundkurses dies hergeben.

Ergänzend ist -falls Zeit bleibt – die Betrachtung weiterer kryptologischer Verfahren und grundlegender Themen sinnvoll, z. B.:

- Codierungen (ASCII, Freimaurer-Alphabet)
- Hash-Verfahren (SHA1, MD5)
- Steganographie

Schlüsselaustausch (Diffie-Hellman)

Einwegverfahren wie die Hashcodes MD5 und SHA1 sollten nach Möglichkeit betrachtet werden, da diese häufig bei Datenbanksystemen zum Einsatz kommen. Hier kann durch Recherche einiges über die Sicherheit der Verfahren herausgefunden werden. Der Schlüsselaustausch stellt ein wesentliches Element zur Anwendung der Verfahren in der Praxis dar, so dass zumindest eine anschauliche Betrachtung (vgl. CS unplugged) im Unterricht erfolgen sollte.

Zeitbedarf: 12 Stunden

Sequenzierung des Unterrichtsvorhabens:

| Unterrichtssequenzen | Zu entwickelnde Kompetenzen | Beispiele, Medien oder Materialien |
|---|--|------------------------------------|
| <p>1. Von der Caesar-Chiffre zu PGP</p> <ul style="list-style-type: none"> a) Prinzipien von Verschlüsselungsverfahren b) Die Mathematik hinter RSA c) Grundideen von PGP u.ä. d) Implementation einfacher Verschlüsselungsverfahren | <p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • analysieren und erläutern Eigenschaften, Funktionsweisen und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren (A) • implementieren einfache Verschlüsselungsverfahren (I) | <p>Internetrecherche</p> |
| <p>2. Grenzen der Verschlüsselungsverfahren</p> <ul style="list-style-type: none"> a) Vergleich der betrachteten Verschlüsselungsverfahren b) Ideen und Angriffspunkte von Verschlüsselungsverfahren c) Beurteilung des Einsatzes von Verschlüsselungsverfahren | <ul style="list-style-type: none"> • untersuchen und beurteilen Grenzen des Problemlösens mit Informatiksystemen (A). | |

Unterrichtsvorhaben Q2-V:

Wiederholung und Vertiefung ausgewählter Kompetenzen und Inhalte des ersten Jahrs der Qualifikationsphase

2.2 Grundsätze der fachmethodischen und fachdidaktischen Arbeit

In Absprache mit der Lehrerkonferenz sowie unter Berücksichtigung des Schulprogramms hat die Fachkonferenz Informatik des Konrad-Zuse-Gymnasiums die folgenden fachmethodischen und fachdidaktischen Grundsätze beschlossen. In diesem Zusammenhang beziehen sich die Grundsätze 1 bis 14 auf fächerübergreifende Aspekte, die auch Gegenstand der Qualitätsanalyse sind, die Grundsätze 15 bis 21 sind fachspezifisch angelegt.

Überfachliche Grundsätze:

- 1) Geeignete Problemstellungen zeichnen die Ziele des Unterrichts vor und bestimmen die Struktur der Lernprozesse.
- 2) Inhalt und Anforderungsniveau des Unterrichts entsprechen dem Leistungsvermögen der Schüler/innen.
- 3) Die Unterrichtsgestaltung ist auf die Ziele und Inhalte abgestimmt.
- 4) Medien und Arbeitsmittel sind schülernah gewählt.
- 5) Die Schüler/innen erreichen einen Lernzuwachs.
- 6) Der Unterricht fördert eine aktive Teilnahme der Schüler/innen.
- 7) Der Unterricht fördert die Zusammenarbeit zwischen den Schülern/innen und bietet ihnen Möglichkeiten zu eigenen Lösungen.
- 8) Der Unterricht berücksichtigt die individuellen Lernwege der einzelnen Schüler/innen.
- 9) Die Schüler/innen erhalten Gelegenheit zu selbstständiger Arbeit und werden dabei unterstützt.
- 10) Der Unterricht fördert strukturierte und funktionale Partner- bzw. Gruppenarbeit.
- 11) Der Unterricht fördert strukturierte und funktionale Arbeit im Plenum.
- 12) Die Lernumgebung ist vorbereitet; der Ordnungsrahmen wird eingehalten.
- 13) Die Lehr- und Lernzeit wird intensiv für Unterrichtszwecke genutzt.
- 14) Es herrscht ein positives pädagogisches Klima im Unterricht.

Fachliche Grundsätze:

- 15) Der Unterricht unterliegt der Wissenschaftsorientierung und ist dementsprechend eng verzahnt mit seiner Bezugswissenschaft.
- 16) Der Unterricht ist problemorientiert und soll von realen Problemen ausgehen und sich auf solche rückbeziehen.
- 17) Der Unterricht folgt dem Prinzip der Exemplarizität und soll ermöglichen, informatische Strukturen und Gesetzmäßigkeiten in den ausgewählten Problemen und Projekten zu erkennen.
- 18) Der Unterricht ist anschaulich sowie gegenwarts- und zukunftsorientiert und gewinnt dadurch für die Schülerinnen und Schüler an Bedeutsamkeit.
- 19) Der Unterricht ist handlungsorientiert, d.h. projekt- und produktorientiert angelegt.
- 20) Im Unterricht werden sowohl für die Schule didaktisch reduzierte als auch reale Informatiksysteme aus der Wissenschafts-, Berufs- und Lebenswelt eingesetzt.
- 21) Der Unterricht beinhaltet reale Begegnung mit Informatiksystemen.

2.3 Grundsätze der Leistungsbewertung und Leistungsrückmeldung

2.3.1 Allgemeines

Die Grundsätze der Leistungsbewertung ergeben sich aus dem Schulgesetz, das Verfahren aus der Verordnung über den Bildungsgang und die Abiturprüfung in der gymnasialen Oberstufe (APOGOST).

Bewertet werden der Umfang der Kenntnisse, die methodische Selbstständigkeit in ihrer Anwendung sowie die sachgemäße schriftliche und mündliche Darstellung.

Bei der schriftlichen und mündlichen Darstellung wird auf sachliche und sprachliche Richtigkeit, auf fachsprachliche Korrektheit, auf gedankliche Klarheit und auf eine der Aufgabenstellung angemessene Ausdrucksweise geachtet.

Bei Gruppenarbeiten muss die individuelle Leistung des einzelnen Schülers erkennbar und bewertbar sein.

Dem Beurteilungsbereich „Sonstige Mitarbeit“ kommt der gleiche Stellenwert zu wie dem Beurteilungsbereich „Klausuren“. Die Teilnoten in den beiden Beurteilungsbereichen werden unabhängig voneinander gebildet. Die Gesamtnote wird nicht rein arithmetisch, sondern nach pädagogischen Gesichtspunkten gebildet. Dabei ist die Entwicklung der Schülerin bzw. des Schülers über einen längeren Zeitraum zu beachten.

Anzahl und Dauer der Klausuren

| Stufe | EF | EF | Q1.1 | Q1.2 | Q2.1 | Q2.2 | (Abitur) |
|-------------|----|----|------|------|------|------|----------|
| Anzahl | 1 | 1 | 2 | 2 | 2 | 1 | 1 |
| Dauer / min | 90 | 90 | 90 | 90 | 135 | 180 | 180 |

2.3.2 Beurteilungsbereich Klausuren

Klausuren dienen der schriftlichen Überprüfung der Lernergebnisse in einem Kursabschnitt. Sie sollen darüber Aufschluss geben, inwieweit im laufenden Kursabschnitt gesetzte Ziele erreicht worden sind. Sie werden nach den Vorgaben der Richtlinien Informatik für die SII korrigiert und beurteilt. Sie bereiten auf die komplexen Anforderungen in der Abiturprüfung vor.

Im zweiten Halbjahr des ersten Jahres der Qualifikationsphase (Q1) kann eine Klausur durch eine Facharbeit ersetzt werden. Die Kriterien zur Bewertung der Facharbeit sind in den Richtlinien Informatik der SII festgelegt.

Die Bewertung der Klausuren erfolgt nach dem folgenden Schema:

Einführungsphase

| Note | sehr gut (1) | gut (2) | befriedigend (3) | ausreichend (4) | mangelhaft (5) | ungenügend (6) |
|--------|--------------|---------|------------------|-----------------|----------------|----------------|
| ab ca. | 86% | 71% | 56% | 41% | 21% | 0% |

Eine Angabe von Notentendenzen (plus/ minus) ist bei der Benotung der Klausuren möglich.

Qualifikationsphase

| | | | | | | | | | | | | | | | | |
|--------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|
| Punkte | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Note | 1+ | 1 | 1- | 2+ | 2 | 2- | 3+ | 3 | 3- | 4+ | 4 | 4- | 5+ | 5 | 5- | 6 |
| ab ca. | 96 | 91 | 86 | 81 | 76 | 71 | 66 | 61 | 56 | 51 | 46 | 41 | 34 | 27 | 20 | 0 |

Der defizitäre Bereich beginnt bei der Note „ausreichend minus“. Formalfehler und Mängel in der Darstellung werden mit bis zu 15% der Gesamtpunktzahl gewichtet.

2.3.3. Beurteilungsbereich „Sonstige Mitarbeit“

Im Beurteilungsbereich „Sonstige Mitarbeit“ sind alle Leistungen zu werten, die eine Schülerin oder ein Schüler im Zusammenhang mit dem Unterricht mit Ausnahme der Klausuren und der Facharbeit erbringt.

Dazu gehören Beiträge zum Unterrichtsgespräch, beim selbstständigen Arbeiten, in Gruppenarbeit, bei der Mitarbeit in Projekten sowie bei der Präsentation von Arbeitsergebnissen.

Die Schülerinnen und Schüler werden durch die verschiedenen Formen der „Sonstigen Mitarbeit“ auf die mündliche Abiturprüfung vorbereitet und lernen deren Struktur und Beurteilungskriterien kennen.

Die in 1.4.3 für die SI genannten Teilbereiche der sonstigen Mitarbeit gelten auch für die Oberstufe. Nach den Richtlinien Informatik für die SII sind die Beiträge der Schülerinnen und Schüler zum Unterrichtsgespräch Basis der Leistungsbewertung in diesem Beurteilungsbereich. Zudem hat das Vortragen der Hausaufgaben im Fach Informatik einen beachtlichen Stellenwert.

Verstärkt sollen jedoch in der Oberstufe Selbstständiges Arbeiten (auch und insbesondere in Vertretungsstunden) sowie Arbeiten in Gruppen und Projekten gefordert und gefördert werden. Bei der Leistungsbewertung in diesen Arbeitsformen können Gesichtspunkte sein, wie und in welchem Umfang die Schülerinnen und Schüler

- Beiträge zur Arbeit leisten
- Beiträge anderer aufnehmen und weiterentwickeln
- sich in die Denkweisen anderer einfinden
- Aufgaben wie Gesprächsleitung, Protokollführung, Programmorganisation übernehmen
- Informationen beschaffen und erschließen
- ihre Gruppenarbeit organisieren und durchführen, auch in arbeitsteiligen Verfahren
- systematische und heuristische Vorgehensweisen nutzen
- ihre Arbeitsschritte überprüfen, diskutieren und dokumentieren

Bei der selbstständigen Arbeit kann darüber hinaus mit bewertet werden, inwieweit eine Schülerin oder ein Schüler in der Lage ist

- das eigene Lernen zielbewusst zu planen und zu steuern
- den eigenen Lernerfolg zu überprüfen

- daraus Rückschlüsse zu ziehen für das weitere Lernen

3 Entscheidungen zu fach- und unterrichtsübergreifenden Fragen

Die Fachkonferenz Informatik hat sich im Rahmen des Schulprogramms für folgende zentrale Schwerpunkte entschieden:

Zusammenarbeit mit anderen Fächern

Im Informatikunterricht werden Kompetenzen anhand informatischer Inhalte in verschiedenen Anwendungskontexten erworben, in denen Schülerinnen und Schülern aus anderen Fächern Kenntnisse mitbringen können. Diese können insbesondere bei der Auswahl und Bearbeitung von Softwareprojekten berücksichtigt werden und in einem hinsichtlich der informatischen Problemstellung angemessenem Maß in den Unterricht Eingang finden.

Insbesondere stellen die Fächer Mathematik, Physik und Sozialwissenschaften eine Fülle von Problemen bereit, deren Behandlung den Informatikunterricht bereichern. Genannt seien hier:

Mathematik: Primzahlbehandlung, Sieb des Eratosthenes, Gauß-Algorithmus zur Lösung von Gleichungssystemen, Behandlung von Wachstumsprozessen, Grafische Auswertung von statistischem Datenmaterial, Berechnung von Ausgleichsgeraden, Zahlensysteme (Binärsystem, Hexadezimalsystem), Auswertung mathematischer Terme durch Syntaxanalyse, Funktionsplotter zur grafischen Darstellung von Schaubildern, numerische (näherungsweise) Lösung von Gleichungen, Kurvendiskussion

Physik: Simulation physikalischer Vorgänge, z.B. schiefer Wurf mit Luftreibung

Sozialwissenschaften: Probleme des Schutzes personenbezogener Daten, Datenschutzgesetz, Möglichkeiten zur Identitätsermittlung aus wenigen Daten

4 Qualitätssicherung und Evaluation

Durch Diskussion der Aufgabenstellung von Klausuren in Fachdienstbesprechungen und eine regelmäßige Erörterung der Ergebnisse von Leistungsüberprüfungen wird ein hohes Maß an fachlicher Qualitätssicherung erreicht.

Das schulinterne Curriculum (siehe 2.1) ist zunächst bis 2017 für den ersten Durchgang durch die gymnasiale Oberstufe nach Erlass des Kernlehrplanes verbindlich. Erstmals nach Ende der Einführungsphase im Sommer 2015, werden in einer Sitzung der Fachkonferenz Erfahrungen ausgetauscht und ggf. Änderungen für den nächsten Durchgang der Einführungsphase beschlossen, um erkannten ungünstigen Entscheidungen schnellstmöglich entgegenwirken zu können.

Nach Abschluss des Abiturs 2017 wird die Fachkonferenz Informatik auf der Grundlage ihrer Unterrichtserfahrungen eine Gesamtsicht des schulinternen Curriculums vornehmen und ggf. eine Beschlussvorlage für die erste Fachkonferenz des folgenden Schuljahres erstellen.

5 Anhang

Die Tabelle auf den folgenden Seiten gibt brauchbare Kriterien für die Bildung der Note im Bereich „Sonstige Mitarbeit“.

| Note/ Punkte | Unterrichts- gespräch und Sachkompe- tenz | Hausaufga- ben | Methoden- kompetenz | Verhalten bei Gruppenar- beit, Sozial- kompetenz | Selbstkompe- tenz |
|--------------------------------|---|--|---|---|---|
| sehr gut/ 13-15 | gleichmäßig hohe, konzentrierte und äußerst qualitätsvolle Mitarbeit im Unterricht; Erkennen eines Problems und dessen Einordnung und eigenständige Lösung in einen größeren Zusammenhang; sachgerechte und ausgewogene Beurteilung; angemessene, klare sprachliche Darstellung und richtige Verwendung von Fachbegriffen, Algorithmen und Strukturen | HA und Projekte regelmäßig, differenziert und gründlich mit herausragenden Ergebnissen | methodische Vielfalt; zielsicheres Beschaffen von Informationen und deren Verarbeitung; überzeugende Präsentation auch von Teilergebnissen; besonders strukturierte und inhaltliche Qualität; sachgerechter, souveräner Einsatz fachspezifischer Arbeitstechniken | bringt bei Schwierigkeiten die ganze Gruppe voran; übernimmt Verantwortung für die Gruppe, unterstützt die anderen hervorragend | zeigt eine ausgeprägte Bereitschaft, sich auf Fragestellungen des Unterrichts einzulassen und sich mit diesen korrelativ sowie kritisch auseinanderzusetzen; anstrengungsbereit, intrinsisch motiviert; selbstbewusst, nicht leicht zu entmutigen |
| gut/ 10-12 | hohe Mitarbeit im Unterricht; Verständnis schwieriger Sachverhalte und deren Einordnung in den Gesamtzusammenhang des Themas; Erkennen des Problems; Kenntnisse über die Unterrichtsreihe hinaus; Verwendung von Fachbegriffen; vermittelte Fachkenntnis- | HA und Projekte regelmäßig und differenziert mit guten Leistungen | selbstständige Informationsbeschaffung; Kenntnis und souveräne Anwendung verschiedener Arbeits- und Präsentationstechniken; sachgerechter, souveräner Einsatz fachspezifischer Arbeitstechniken | aktiv an zeitökonomischer Planung und Durchführung der GA beteiligt, wirkt bei Schwierigkeiten aktiv an Problemlösung mit, geht aktiv auf Meinungen anderer ein | zeigt eine ausgeprägte Bereitschaft, sich auf Fragestellungen des Unterrichts einzulassen und sich mit diesen korrelativ auseinanderzusetzen; traut sich auch schwierige Aufgaben zu; Gedankengänge werden selbstständig weiterentwickelt und klar dargestellt und argumentativ |

| | | | | | |
|------------------------------|--|--|--|--|---|
| | se werden beherrscht; unterrichtsfördernde Beiträge; weitgehend vollständige weiterführende Ausführungen; sicherer Umgang auch mit anspruchsvollen Problemen und sachgerechte Anwendung von Fachtermini; | | | | vertreten |
| befriedigend/ 7-9 | insgesamt regelmäßig freiwillige Mitarbeit im Unterricht; im Wesentlichen richtige Wiedergabe einfacher Fakten und Zusammenhänge aus unmittelbar behandeltem Stoff; vermittelte Fachkenntnisse werden überwiegend beherrscht; weitgehend vollständige, differenzierte, sachbezogene Ausführungen; mittelschwere Algorithmen werden sachgerecht verstanden und bearbeitet; Darstellungen in der Programmiersprache sind weitgehend brauchbar; be- | HA und Projekte regelmäßig und mit befriedigenden Leistungen | kann projektdienliche Informationen einbringen, zugeeilte Inhalte erfassen und dokumentieren; fachspezifische Arbeitstechniken werden meist sachgerecht ausgewählt und eingesetzt; Verwendung adäquater Arbeitsmaterialien | erkennbare Mitverantwortung für das gemeinsame Projekt; sorgt mit für störungsfreies Miteinander; zeigt Bereitschaft, eigene und Gruppenergebnisse zu präsentieren | zeigt Bereitschaft, sich auf Fragestellungen des Unterrichtes einzulassen und sich mit diesen korrelativ auseinanderzusetzen; lässt sich nicht leicht entmutigen; greift gelegentlich Beiträge anderer auf und führt diese fort; verständliche Ausdrucksweise und leicht nachvollziehbare Programmteile mit hinreichenden Kommentaren |

| | | | | | |
|-------------------------|--|--|--|---|---|
| | grenztes Problembewusstsein | | | | |
| ausreichend/ 4-6 | nur gelegentlich freiwillige Mitarbeit im Unterricht; Äußerungen beschränken sich auf die Wiedergabe einfacher Fakten und Zusammenhänge aus dem unmittelbar behandelten Stoffgebiet und sind im Wesentlichen richtig; vermittelte Fachkenntnisse werden mit Einschränkungen beherrscht; leichtere Programmstücke werden dem Sinn nach richtig erfasst; teilweise nachlässige Darstellungen in der Programmiersprache | HA und Projekte nicht regelmäßig oder nur oberflächlich erledigt | Schwierigkeiten, Arbeitsabläufe folgerichtig und zeitökonomisch zu planen, fachspezifische Arbeitstechniken (z.B. Programmierung; Umsetzung v. Algorithmen; Programmanalyse) werden eingesetzt | Kommunikationsfähigkeit und -bereitschaft nur in Ansätzen; verlässt sich lieber auf die anderen Gruppenmitglieder; beteiligt sich zuweilen aktiv an Entscheidungsprozessen in der Gruppe; selten Präsentation von Ergebnissen | zeigt Bereitschaft, sich auf Fragestellungen des Unterrichtes einzulassen; neigt bei auftretenden Schwierigkeiten zu ausweichendem Verhalten, ist auf Nachfrage in der Lage, sich zu den Beiträgen anderer zu äußern; folgt den Beiträgen anderer noch aufmerksam |
| mangelhaft/ 1-3 | überwiegend überpassives Verhalten im Unterricht; Äußerungen nach Aufforderung sind einsilbig, unstrukturiert und nur teilweise richtig; Fehlleistungen, auch nach Vorbereitung; Schwierigkeiten bei sachbezogener Verarbeitung | HA oder Projekte häufig nicht vorhanden oder nur lückenhaft erledigt; zu vorbereiteten Themen meist keine sachbezogenen Äußerungen | kaum in der Lage, mit den Lerngegenständen sachgerecht und systematisch umzugehen; legt lediglich unverarbeitetes Material (z.B. Internetausdrucke oder abgeschriebene Programmteile) vor | wenig projektdienliche Mitarbeit; wenig zuverlässig; nicht auf Gruppenarbeit vorbereitet; übernimmt keine Mitverantwortung; beschränkt Rolle am liebsten auf „Programmtester“ für die Gruppe | kaum Bereitschaft, sich auf Fragestellungen des Unterrichtes einzulassen |

| | | | | | |
|--------------------------------------|---|--|---|---|---|
| | von Wissen und der Verknüpfung von Zusammenhängen; oft mangelndes Verständnis der eigentlichen Probleme | | | | |
| unge- nü- gend/ 0 | keine freiwillige Mitarbeit im Unterricht; Äußerungen nach Aufforderung sind falsch; Fachwissen nicht erkennbar | sehr häufig keine HA, fehlende Projektarbeit | keine Arbeitsplanung; nicht in der Lage, mit den Lerngegenständen sachgerecht umzugehen; nicht in der Lage, Informationen einzuholen und darzustellen | kein situationsangemessenes Gesprächsverhalten; nicht kooperativ; bei Gruppenarbeit kein Interesse an eigenem Arbeitsanteil und an Arbeitskontakten zu Mitschülern; hält sich nicht an Regeln | keine Bereitschaft, sich auf Fragestellungen des Unterrichtes einzulassen; fehlende Selbständigkeit im Arbeiten; Abschreiben der Arbeitsergebnisse der Mitschüler; wiederholtes unentschuldigtes Fehlen |