



Clemens-Brentano-Gymnasium, Dülmen

Schulinterner Lehrplan

Informatik

Sekundarstufe II

Stand: 16.05.2022 auf Basis der am 07.12.2021 durch die FK Informatik verabschiedeten Version.
(nur redaktionelle Änderungen, wie gendergerechte Ansprache)

Inhaltsverzeichnis

| | |
|--|----|
| 1. Rahmenbedingungen der fachlichen Arbeit | 3 |
| 2. Entscheidungen zum Unterricht | 5 |
| 2.1 Unterrichtsvorhaben | 6 |
| 2.2 Konkretisierungen | 15 |
| 2.3 Grundsätze der fachmethodischen und –didaktischen Arbeit..... | 48 |
| 2.4 Grundsätze der Leistungsbewertung und Leistungsrückmeldung | 49 |
| 2.5 Lehr- und Lernmittel..... | 50 |
| 3. Entscheidungen zu fach- oder unterrichtsübergreifenden Fragen | 51 |
| 4. Qualitätssicherung und Evaluation | 52 |
| 5. Anhang | 53 |

1. Rahmenbedingungen der fachlichen Arbeit

Lage und Rahmenbedingungen der Schule

Das Clemens-Brentano-Gymnasium liegt am Rand des Stadtzentrums der Kleinstadt Dülmen mit etwas weniger als 50.000 Einwohnern, die zum Kreis Coesfeld gehört. Die Umgebung der Stadt ist ländlich durch die Lage innerhalb des Münsterlandes geprägt. Viele Einwohnerinnen und Einwohner pendeln einerseits in Richtung des Ruhrgebiets, andererseits in das Umfeld der Stadt Münster. Auch hinsichtlich ihrer Freizeitgestaltung orientiert man sich in der Stadt in beide Richtungen.

Das Einzugsgebiet des Gymnasiums ist verhältnismäßig groß, aufgrund zahlreicher entfernt liegender Bauerschaften und eingemeindeten Orten ist ein nicht unerheblicher Teil der Schülerschaft auf die Beförderung mit Schulbussen angewiesen.

Ein größeres Waldgebiet ist von der Schule aus zu Fuß erreichbar, ebenso die typischen städtischen Einrichtungen, wie bspw. die Stadtbücherei, das Stadtarchiv oder die Stadtverwaltung.

Das Clemens-Brentano-Gymnasium ist das ältere der beiden Gymnasien der Stadt und existiert seit dem großen Stadtjubiläum 1911. In der Nachbarschaft angesiedelt ist das 1999 gegründete Annette-von-Droste-Hülshoff-Gymnasium, mit dem man in der gymnasialen Oberstufe in einer Reihe von Kursen kooperiert, um den Schüler*innen ein breites Fächerangebot unterbreiten zu können.

Seit dem Jahre 2011 ist an der Schule der gebundene Ganzttag eingeführt. Im Jahre 2014 wurde die Mensa eingeweiht.

Schwerpunkt der weiteren Schulentwicklung ist die Einrichtung von Tablet-Klassen ab Jahrgangsstufe 5.2, beginnend im Schuljahr 2018/19.

Es besuchen ca. 700 Schüler*innen das Clemens-Brentano-Gymnasium und werden dabei von ca. 70 Lehrkräften unterrichtet. In den letzten Jahren gibt es im Schnitt drei bis vier Klassen pro Jahrgangsstufe in der Sekundarstufe I und ca. 60 bis 100 Schüler*innen pro Jahrgangsstufe in der Sekundarstufe II.

Rahmenbedingungen des Faches an der Schule

Das Fach Informatik wird beginnend für die Jahrgangsstufe 5 (SJ 21/22) im Schuljahr 2022/23 als zweistündiges Fach in der Jahrgangsstufen 6 sowie traditionell im Wahlpflichtbereich II in den Jahrgangsstufen 9 und 10 dreistündig angeboten. Zusätzlich gibt es noch eine Stunde „Lernen Medien“ in der Jahrgangsstufe 5.

Vor der Umstellung auf den doppelstündigen Unterricht in Jahrgangsstufe 6, wurde Informatik am CBG jeweils einstündig in den Jahrgangsstufen 5 und 8 unterrichtet. Die Jahrgänge, die bisher nur ein einstündiges Angebot in der 5 hatten, werden im Übergang weiterhin einstündig in der Jahrgangsstufe 8 unterrichtet.

In der Sekundarstufe II besteht das Angebot aus einem dreistündigen Unterricht in der EF, sowie als dreistündiger Grundkurs in der Q1 und Q2. Diese Kurse werden teilweise im Rahmen der Kooperation der beiden städtischen Gymnasien angeboten.

Das Fach Informatik kann als drittes oder als viertes Abiturfach belegt werden.

Die Fachschaft Informatik bietet in Kooperationen mit den Fachschaften Mathematik und Physik einen Projektkurs in der Q1 an, der aber nur selten zu Stande kommt.

Es besteht die Möglichkeit im Bereich Informatik eine besondere Lernleistung abzulegen. Von dieser Möglichkeit machen einzelne Schüler*innen in manchen Jahren Gebrauch.

Es werden AGs in der Sek I angeboten, so z.B. traditionell die Lego-Roboter AG in der Erprobungsstufe oder die Arduino-AG im Bereich der Mittelstufe. Letztere in Kooperation mit der Fachschaft Physik.

Die Teilnahme an Informatikwettbewerben wie dem Informatik-Biber und dem Bundeswettbewerb Informatik wird durch die Fachschaft gefördert und begleitet.

Die Fachschaft Informatik begleitet mit ihrem Knowhow die Digitalisierung der Schule. Sie stellt eine Verzahnung der „Lernen Medien“-Stunden mit dem Informatikunterricht sicher. Die vermittelte informatische Grundbildung steht allen Fächern als Basis zur Verfügung, die entlang des Medienkompetenzrahmens von diesen weiter ausgebaut wird.

In der Oberstufe ist das Lehrwerk „Informatik“ des Schöningh-Verlags eingeführt. In den anderen Jahrgangsstufen wird mit Arbeitsblättern und Onlineangeboten gearbeitet. Für die Jahrgangsstufe 6 wird voraussichtlich ein Lehrwerk zum Schuljahr 2022/23 eingeführt.

Das CBG verfügt über zwei Computerräume mit jeweils 17 windowsbasierten PC-Arbeitsplätzen, in denen der Informatikunterricht stattfindet. Darüber hinaus steht den Schüler*innen eine Mediothek mit 9 windowsbasierten PC-Arbeitsplätzen zur Verfügung, wobei den Sek-I-Jahrgängen der Zugang nur möglich ist, wenn eine Mediotheksaufsicht anwesend ist.

Die Schule besitzt zwei Leihtablet-Klassensätze mit jeweils 16 Tablets. Nach und nach sollen alle Schüler*innen ein eigenes Tablet erhalten. Gestartet wurde mit der Jahrgangsstufe 5 im Schuljahr 2018/19. Mit Stand Ende des Schuljahres 2021/22 war nur noch die aktuelle Q1 nicht mit durch die Schule verwaltete Tablets versorgt.

2. Entscheidungen zum Unterricht

2.1 Unterrichtsvorhaben

| Jahrgangsstufe EF | |
|---|---|
| <p><u>Unterrichtsvorhaben I:</u> Thema/Kontext: Einführung in das Rechnersystem Kompetenzen: <ul style="list-style-type: none"> • Darstellen und Interpretieren Inhaltsfeld: Betriebssystem und Einsatz von Informatiksystemen Inhaltliche Schwerpunkte: Einsatz von Informatiksystemen Zeitbedarf: Min. 1 Woche</p> | <p><u>Unterrichtsvorhaben II:</u> Thema/Kontext: Grundlagen der objektorientierten Analyse, Modellierung und Implementierung (anhand der Analyse eines Simulationsbeispiels) Kompetenzen: <ul style="list-style-type: none"> • Modellieren Inhaltsfeld: Daten und ihre Strukturierung Inhaltliche Schwerpunkte: Objekte und Klassen Zeitbedarf: Min. 2 Wochen</p> |
| <p><u>Unterrichtsvorhaben III:</u> Thema/Kontext: Grundlagen der objektorientierten Programmierung in Java anhand einer Simulation oder eines Spiels Kompetenzen: <ul style="list-style-type: none"> • Modellieren • Implementieren Inhaltsfeld: Formale Sprachen und Automaten Inhaltliche Schwerpunkte: Syntax und Semantik der Programmiersprache Java Zeitbedarf: Min. 6 Wochen</p> | <p><u>Unterrichtsvorhaben IV:</u> Thema/Kontext: Grundlagen der Objektorientierten Programmierung in Java unter besonderer Berücksichtigung algorithmischer Grundstrukturen Kompetenzen: <ul style="list-style-type: none"> • Modellieren • Implementieren • Argumentieren Inhaltsfeld: Algorithmen Inhaltliche Schwerpunkte: Analyse, Entwurf und Implementierung einfacher Algorithmen Syntax und Semantik der Programmier-sprache Java Zeitbedarf: Min. 6 Wochen</p> |

| | |
|--|---|
| <p><u>Unterrichtsvorhaben V:</u> Thema/Kontext: Vorbereitung und Präsentation von Referaten zu:</p> <ul style="list-style-type: none"> • Geschichte der Digitalen Datenverarbeitung • Von-Neumann-Architektur • Darstellung von Zeichen und Zeichen in Binärcodes <p>Kompetenzen:</p> <ul style="list-style-type: none"> • Kommunizieren • Kooperieren <p>Inhaltsfelder: Informatiksysteme Informatik, Mensch und Gesellschaft</p> <p>Inhaltliche Schwerpunkte: Geschichte der automatischen Datenverarbeitung, Internet, Einzelrechner, Digitalisierung</p> <p>Zeitbedarf: Min. 2 Wochen</p> | <p><u>Unterrichtsvorhaben VI:</u> Thema/Kontext: Exemplarische Planung und Durchführung eines Softwareprojekts</p> <p>Kompetenzen:</p> <ul style="list-style-type: none"> • Modellieren • Implementieren • Kommunizieren • Kooperieren <p>Inhaltsfeld: Daten und ihre Strukturierung</p> <p>Inhaltliche Schwerpunkte: Objekte und Klassen, Lineare Datenstruktur Array</p> <p>Zeitbedarf: Min. 6 Wochen</p> |
| <p><u>Unterrichtsvorhaben VII:</u> Thema/Kontext: Suchen und Sortieren von z.B. Zahlen oder Buchstaben</p> <p>Kompetenzen:</p> <ul style="list-style-type: none"> • Modellieren • Kommunizieren • Kooperieren <p>Inhaltsfeld: Algorithmen</p> <p>Inhaltliche Schwerpunkte: Algorithmen zum Suchen und Sortieren</p> <p>Zeitbedarf: Min. 1 Woche</p> | <p><u>Unterrichtsvorhaben VIII:</u> Thema/Kontext: Auswirkungen des Einsatzes von Informatiksystemen am Beispiel eines aktuellen Themas</p> <p>Kompetenzen:</p> <ul style="list-style-type: none"> • Argumentieren <p>Inhaltsfeld: Informatik, Mensch und Gesellschaft</p> <p>Inhaltliche Schwerpunkte: Wirkungen der Automatisierung</p> <p>Zeitbedarf: Min. 1 Woche</p> |
| <p>Summe der Stunden in der Jahrgangsstufe: min. 25 Wochen</p> | |

| Anbindung an die Rahmenvorgaben Verbraucherbildung | Anbindungen an das Medienkonzept | Anbindungen an das StuBo-Konzept: | Anbindungen an das Methodenkonzept |
|--|---|--|------------------------------------|
| | <p>Die Mediennutzung und deren Betrachtung sind in der Informatik omnipräsent. In dieser Jahrgangsstufe werden verstärkt folgende Gebiete adressiert:</p> <p>Bedienen und Anwenden</p> <ul style="list-style-type: none"> • Medianausstattung • Digitale Werkzeuge • Datenorganisation • Datenschutz und Informationssicherheit <p>Kommunikations- u. Kooperationsprozesse</p> <ul style="list-style-type: none"> • Kommunikations- und Kooperationsprozesse • Kommunikation und Kooperation in der Gesellschaft <p>Produzieren und Präsentieren</p> <ul style="list-style-type: none"> • Medienproduktion und Präsentation • Gestaltungsmittel • Quellendokumentation • Rechtliche Grundlagen <p>Analysieren und Reflektieren</p> <ul style="list-style-type: none"> • Selbstregulierte Mediennutzung <p>Problemlösen und Modellieren</p> <ul style="list-style-type: none"> • Prinzipien der digitalen Welt • Algorithmen erkennen • Modellieren und Programmieren • Bedeutung von Algorithmen | <p>UV VIII: Veränderung der Arbeitswelt durch Automatisierung</p> <p>jahrgangsstufenübergreifend: Ehemalige S*, die ein Studium oder eine Ausbildung im IT-Bereich begonnen haben, stellen ihr Studienfach bzw. ihren Ausbildungsgang vor. (siehe Q1-VI)</p> | |

Jahrgangsstufe Q1

Unterrichtsvorhaben I:

Thema/Kontext:

Wiederholung der objektorientierten Modellierung und Programmierung anhand einer kontextbezogenen Problemstellung

Kompetenzen:

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

Inhaltsfeld:

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

Inhaltliche Schwerpunkte:

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

Klausur:

Über UV I/II, klausurterminabhängig

Zeitbedarf:

Min. 3 Woche

Unterrichtsvorhaben II:

Thema/Kontext:

Modellierung und Implementierung von Anwendungen mit dynamischen, linearen Datenstrukturen

Kompetenzen:

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

Inhaltsfeld:

- 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

Klausur:

Über UV I/II, klausurterminabhängig

Zeitbedarf:

Min. 6 Wochen

| | |
|--|---|
| <p><u>Unterrichtsvorhaben III:</u></p> <p>Thema/Kontext: Suchen und Sortieren auf linearen Datenstrukturen</p> <p>Kompetenzen:</p> <ul style="list-style-type: none"> • Argumentieren • Modellieren • Implementieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren <p>Inhaltsfeld:</p> <ul style="list-style-type: none"> • Algorithmen • Formale Sprachen und Automaten <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Analyse, Entwurf und Implementierung von Algorithmen • Algorithmen in ausgewählten informatischen Kontexten • Syntax und Semantik einer Programmiersprache <p>Klausur: Über UV II/III, klausurterminabhängig</p> <p>Zeitbedarf: Min. 6 Wochen</p> | <p><u>Unterrichtsvorhaben IV:</u></p> <p>Thema/Kontext: Modellierung und Nutzung von relationalen Datenbanken in Anwendungskontexten</p> <p>Kompetenzen:</p> <ul style="list-style-type: none"> • Argumentieren • Modellieren • Implementieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren <p>Inhaltsfeld:</p> <ul style="list-style-type: none"> • Daten und ihre Strukturierung • Algorithmen • Formale Sprachen und Automaten • Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Datenbanken • Algorithmen in ausgewählten informatischen Kontexten • Syntax und Semantik einer Programmiersprache • Sicherheit <p>Klausur: Über UV III/IV, klausurterminabhängig</p> <p>Zeitbedarf: Min. 6 Wochen</p> |
|--|---|

| | |
|---|---|
| <p><u>Unterrichtsvorhaben V:</u> Thema/Kontext: Sicherheit und Datenschutz in Netzstrukturen Kompetenzen:</p> <ul style="list-style-type: none"> • Argumentieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> • Informatiksysteme • Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Einzelrechner und Rechnernetzwerke • Sicherheit • Nutzung von Informatiksystemen, Wirkungen der Automatisierung <p>Klausur: Über UV IV/V, klausurterminabhängig Zeitbedarf: Min. 3 Wochen</p> | <p><u>Unterrichtsvorhaben VI (flexible Terminierung):</u> Thema/Kontext: <i>Studien- und Berufsorientierung:</i> Vorstellung von Ausbildungsberufen und Studiengängen im Bereich IT Kompetenzen:</p> <ul style="list-style-type: none"> • Argumentieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> • <i>Studien- und Berufsorientierung</i> • Informatik, Mensch und Gesellschaft <p>Zeitbedarf: 2-4 UStd. flexible Terminierung, da abhängig von Externen (siehe Maßnahmenbeschreibung im Anhang)</p> <p><u>Unterrichtsvorhaben VII = UV I der Q2:</u></p> <p>In langen Schuljahren sollte bereits mit dem Unterrichtsvorhaben I der Q2 vor den Sommerferien gestartet werden.</p> <p>Je nach Klausurplanung der Oberstufenkoordination findet die 1. Klausur der Q2 bereits nach wenigen Unterrichtsstunden statt. Die schon in der Q1 behandelten Inhalte müssen dann zu Beginn der Q2 wiederholt werden.</p> |
| <p>Summe der Stunden in der Jahrgangsstufe: min. 25 Wochen</p> | |

| Anbindung an die Rahmenvorgaben Verbraucherbildung | Anbindungen an das Medienkonzept | Anbindungen an das StuBo-Konzept: | Anbindungen an das Methodenkonzept |
|--|--|---|------------------------------------|
| | Die Mediennutzung und deren Betrachtung sind in der Informatik omnipräsent. In dieser Jahrgangsstufe werden verstärkt folgende Gebiete adressiert: Bedienen und Anwenden <ul style="list-style-type: none"> • Medienausstattung • Digitale Werkzeuge • Datenorganisation | jahrgangsstufenübergreifend: Ehemalige S*, die ein Studium oder eine Ausbildung im IT-Bereich begonnen haben, stellen ihr Studienfach bzw. ihren Ausbildungsgang vor. (siehe Q1-VI) | |

| | | | |
|--|---|--|--|
| | <ul style="list-style-type: none">• Datenschutz und Informationssicherheit <p>Problemlösen und Modellieren</p> <ul style="list-style-type: none">• Prinzipien der digitalen Welt• Algorithmen erkennen• Modellieren und Programmieren• Bedeutung von Algorithmen | | |
|--|---|--|--|

| Jahrgangsstufe Q2 | |
|---|--|
| <p><u>Unterrichtsvorhaben I:</u> Thema/Kontext: Modellierung und Implementierung von Anwendungen mit dynamischen, nichtlinearen Datenstrukturen</p> <p>Kompetenzen:</p> <ul style="list-style-type: none"> • Argumentieren • Modellieren • Implementieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren <p>Inhaltsfeld:</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>Klausur: Über UV I, klausurterminabhängig</p> <p>Zeitbedarf: Min. 7 Woche</p> | <p><u>Unterrichtsvorhaben II:</u> Thema/Kontext: Endliche Automaten und formale Sprachen</p> <p>Kompetenzen:</p> <ul style="list-style-type: none"> • Argumentieren • Modellieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren <p>Inhaltsfeld: Endliche Automaten und formale Sprachen</p> <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Endliche Automaten • Grammatiken regulärer Sprachen • Möglichkeiten und Grenzen von Automaten und formalen Sprachen <p>Klausur: Über UV II/III, klausurterminabhängig</p> <p>Zeitbedarf: Min. 7 Wochen</p> |
| <p><u>Unterrichtsvorhaben III:</u> Thema/Kontext: Prinzipielle Arbeitsweise eines Computers und Grenzen der Automatisierbarkeit</p> <p>Kompetenzen:</p> <ul style="list-style-type: none"> • Argumentieren • Kommunizieren und Kooperieren <p>Inhaltsfeld:</p> <ul style="list-style-type: none"> • Informatiksysteme • Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Einzelrechner und Rechnernetzwerke • Grenzen der Automatisierung <p>Klausur: Über UV II/III, klausurterminabhängig</p> <p>Zeitbedarf: Min. 4 Wochen</p> | <p><u>Unterrichtsvorhaben IV:</u> Thema/Kontext: Wiederholung und Vertiefung ausgewählter Kompetenzen und Inhalte der Qualifikationsphase</p> |
| <p>Summe der Stunden in der Jahrgangsstufe: min. 18 Wochen</p> | |

| Anbindung an die Rahmenvorgaben Verbraucherbildung | Anbindungen an das Medienkonzept | Anbindungen an das StuBo-Konzept: | Anbindungen an das Methodenkonzept |
|--|---|--|------------------------------------|
| | <p>Die Mediennutzung und deren Betrachtung sind in der Informatik omnipräsent. In dieser Jahrgangsstufe werden verstärkt folgende Gebiete adressiert:</p> <p>Bedienen und Anwenden</p> <ul style="list-style-type: none"> • Medienausstattung • Digitale Werkzeuge • Datenorganisation • Datenschutz und Informationssicherheit <p>Problemlösen und Modellieren</p> <ul style="list-style-type: none"> • Prinzipien der digitalen Welt • Algorithmen erkennen • Modellieren und Programmieren • Bedeutung von Algorithmen | <p>jahrgangsstufenübergreifend: Ehemalige S*, die ein Studium oder eine Ausbildung im IT-Bereich begonnen haben, stellen ihr Studienfach bzw. ihren Ausbildungsgang vor. (siehe Q1-VI)</p> | |

2.2 Konkretisierungen

| Klasse /Stufe | Inhaltliche Schwerpunkte | Inhaltliche Konkretisierung am CBG (Mindeststundenzahl) | Fachmethoden u. prozessbezogene Kompetenzen | Inhaltsbezogene Kompetenzen | Kompetenz / Leistungsüberprüfung | Anbindung an ein mögliches Projekt | Individuelle Förderung | Mögliche Vernetzung mit anderen Fächern |
|---------------|---|--|---|-------------------------------|--|---|------------------------|---|
| EF.1. | <p>Unterrichtsvorhaben 1:</p> <p>Einsatz von Informatiksystemen</p> | <p>Einführung in das Rechnersystem</p> <ul style="list-style-type: none"> Die S* werden in das Rechnersystem eingeführt und nutzen dies selbstständig, sicher, zielführend und verantwortungsbewusst zur strukturierten Verwaltung und gemeinsamen Verwendung von Daten. <p>Min. 1 Woche</p> | Darstellen und Interpretieren | Informatiksysteme | | | | |
| | <p>Unterrichtsvorhaben 2:</p> <p>Objekte und Klassen</p> | <p>Analyse einer Simulation</p> <ul style="list-style-type: none"> Objekte, ihre Eigenschaften, Methoden und Beziehungen werden auf Basis einer Simulation ermittelt und in einem Zustandsdiagramm dargestellt | Modellieren | Daten und ihre Strukturierung | Präsentation und Diskussion von Modellierungen | <p>Projekt Autosteuerung durch Hindernisparcours;</p> <p>Objekte vom Typ: Auto, Baum,...;</p> <p>arbeitsteiliges Arbeiten</p> | | |

| | | | | | | | | |
|--|--|--|---|--|--|---|--|--|
| | | <ul style="list-style-type: none"> Hieraus wird eine allgemeine Modellierung der Klassen erstellt Beziehungen der Klassen untereinander werden bestimmt (Ist-Beziehung; Assoziation) <p>Min. 2 Wochen</p> | | | | <p>Eine Oberklasse Auto mit der Steuerung wird zur Verfügung gestellt.</p> <p>Übung: Allgemein: S. 18-19; Greenfoot; Roboterprojekt (S. 19-23)</p> | | |
| | <p>Unterrichtsvorhaben 3:</p> <p>Syntax und Semantik der Programmiersprache Java</p> | <p>Entwicklung einer Simulation oder eines Spiels</p> <ul style="list-style-type: none"> Die S* erzeugen Objekte, nutzen dokumentierte Klassenbibliotheken, um den erstellten Objekten Aufträge zu erteilen, analysieren und erläutern einfache Algorithmen modifizieren einfache Algorithmen erweitern Klassen um Aufträge und dokumentieren ihre Erweiterungen nutzen globale Variablen und ordnen ihnen sinnvolle Datentypen zu speichern Anfragen von Objekten in Variablen <p>Min. 6 Wochen</p> | <p>Modellieren, Implementieren, Helfersystem</p> | <p>Formale Sprachen und Automaten</p> | <p>Präsentationen der Ergebnisse, Protokolle, Referate, Tests</p> | <p>Problem: Objekte lassen sich nicht ausrichten. Bsp.: Wie kann ich einen Zylinder drehen?</p> <p>Dokumentation der Bibliothek Gloop, → Nutzung der Fähigkeiten der Objekte.</p> <p>Erweiterung der Fähigkeiten: Beispielmethode: dreheReifen()</p> <p>Erweiterung der Attribute aktuellePosition; Farbe?; Geschwindigkeit;</p> <p>Methoden mit Wertzuweisung: erhoeheGeschwindigkeit; verringere Geschwindigkeit;</p> | | |

| | | | | | | | | |
|--|--|---|--|--|---|---|--|----------------------------------|
| | | | | | | Berechne Abstand zu Objekt. | | |
| | <p>Unterrichtsvorhaben 4:</p> <p>Analyse, Entwurf und Implementierung einfacher Algorithmen/</p> <p>Syntax und Semantik der Programmiersprache Java</p> | <p>Algorithmen</p> <ul style="list-style-type: none"> analysieren einfache Algorithmen und stellen diese umgangssprachlich und grafisch dar, implementieren einfache Algorithmen unter Nutzung bedingter Anweisungen implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, bedingten Anweisungen, Schleifen und Methodenaufrufen, erweitern Klassen um Anfragen erweitern Klassen um Methoden mit Parametern <p>Min. 6 Wochen</p> | <p>Modellieren, Implementieren, Argumentieren</p> | <p>Algorithmen</p> | <p>Präsentationen der Ergebnisse, Protokolle, Referate, Tests, Klausur</p> | <p>Projekt Bildschirmschoner:</p> <p>auf einen zu Rollende Kugeln (Schleifen)</p> <p>Rollende Kugeln, die automatisch verschwinden</p> <p>Projekt Flugsimulation</p> <p>SetzePosition/ SetzeGeschwindigkeit..</p> | | |
| | <p>Unterrichtsvorhaben 5:</p> <p>Geschichte der automatischen Datenverarbeitung, Internet, Einzel-</p> | <p>Vorbereitung und Präsentation von Referaten zu:</p> <ul style="list-style-type: none"> Geschichte der Digitalen Datenverarbeitung Von-Neumann-Architektur Darstellung von Zeichen und Zeichen in Binärcodes | <p>Kommunizieren u. Kooperieren</p> | <p>Informatiksysteme, Informatik, Mensch und Gesellschaft</p> | <p>Referat, Thesenpapier, evtl. kurze schriftl. Überprüfung</p> | | | <p>Physik, Mathematik</p> |

| | | | | | | | | |
|-------|--|--|--|---|--|--|--|--|
| | rechner, Digitalisierung | Min. 2 Wochen | | | | | | |
| EF. 2 | Unterrichtsvorhaben 6: Objekte und Klassen, Lineare Datenstruktur Array | Exemplarische Planung und Durchführung eines Softwareprojekts mit <ul style="list-style-type: none"> • Assoziation • Vererbung • Sequenzdiagramme zur Visualisierung der Kommunikation zwischen Objekten • Array Min. 6 Wochen | Modellieren und Implementieren; Kommunizieren und Kooperieren | Daten und ihre Strukturierung | | Hindernisse müssen Flugzeug kennen, damit sie ausweichen können / ein Crash simuliert werden kann. | Differenzierungsprojekt für das 2.HJ mithilfe Stationenlernen: Bildverarbeitungsprogramm | Je nach Projekt: Physik, Mathematik (Diff-Projekt: Kunst) |
| | Unterrichtsvorhaben 7: Algorithmen zum Suchen und Sortieren | Suchen und Sortieren von z.B. Zahlen oder Buchstaben Datenstrukturen und Algorithmen zum Suchen und Sortieren Min. 1 Woche | Modellieren, Kommunizieren und Kooperieren | Daten und ihre Strukturierung; Algorithmen | | | | |
| | Unterrichtsvorhaben 8: Wirkungen der Automatisierung | Auswirkungen des Einsatzes von Informatiksystemen am Beispiel eines aktuellen Themas Min. 1 Woche | Argumentieren | Informatik, Mensch und Gesellschaft | | z.B. „Verdatet und Verkauft – wem gehören unsere Daten“ (Filmbeitrag Quarks u. Co WDR 20.05.2014) | | Sozialwissenschaften |

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 Schüler*innen 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. Dabei 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.

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</p> <p>(b) Analyse der Modellierung (Implementationsdiagramm)</p> <p>(c) Erweiterung der Modellierung im Implementationsdiagramm (Vererbung, abstrakte Klasse)</p> <p>(d) Kommunikation zwischen mindestens zwei Objekten (grafische Darstellung)</p> <p>(e) Dokumentation von Klassen</p> <p>(f) Implementierung der Anwendung oder von Teilen der Anwendung</p> | <p>Die Schüler*innen</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 Implementierung und zur Analyse von Programmen (I), • wenden eine didaktisch orientierte Entwicklungsumgebung zur | <p><i>Beispiel: Pong</i></p> |

| | | |
|--|--|--|
| | <p>Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an (I),</p> <ul style="list-style-type: none">• 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 Last-In-First-Out-Prinzip verwaltet werden, werden der Aufbau von Stapeln am Beispiel dargestellt und die Operationen der Klasse `Stack` erläutert. Anschließend werden für die Anwendung notwendige Klassen modelliert und implementiert. Anschließend wird die Anwendung modifiziert, um den Umgang mit der Datenstruktur zu üben. Anhand einer Anwendung, in der Daten nach dem First-In-First-Out-Prinzip verwaltet werden, werden Unterschiede zwischen den Datenstrukturen Schlange und Stapel erarbeitet und die Operationen der Klasse `Queue` betrachtet. 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.

Sequenzierung des Unterrichtsvorhabens:

| Unterrichtssequenzen | Zu entwickelnde Kompetenzen | Beispiele, Medien, Materialien |
|---|--|---|
| <p>1. Die Datenstruktur Stapel im Anwendungskontext unter Nutzung der Klasse <code>Stack</code></p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <p>(b) Erarbeitung der Funktionalität der Klasse <code>Stack</code></p> | <p>Die Schüler*innen</p> <ul style="list-style-type: none"> • erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A), • analysieren und erläutern Algorithmen und Programme (A), | <p><i>Beispiel:</i> Palindrom o. Klammerausdruck Palindrome OTTO und später Klammerausdrücke (<code>{[]}</code>) als verwandtes Problem auf Korrektheit testen</p> <p><i>Mögliche Alternativbeispiele:</i></p> <p><i>Beispiel:</i> Heftstapel</p> |

| | | |
|---|---|--|
| <p>(c) Modellierung und Implementierung der Anwendung unter Verwendung eines oder mehrerer Objekte der Klasse <code>Stack</code></p> | <ul style="list-style-type: none"> • beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M), • 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>In einem Heftstapel soll das Heft einer Schülerin gefunden werden.</p> <p>oder</p> <p><i>Beispiel:</i> Kisten stapeln</p> <p>In einem Stapel nummerierter Kisten soll eine bestimmte Kiste gefunden und an einen Kunden geliefert werden. Dazu müssen Kisten auf verschiedene Stapel gestapelt und wieder zurückgestellt werden.</p> <p><i>Materialien:</i> Zentralabiturklassenbeschreibung und vorgegebene Zentralabiturklasse <code>Stack</code></p> |
| <p>2. Die Datenstruktur Schlange im Anwendungskontext unter Nutzung der Klasse <code>Queue</code></p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <p>(b) Erarbeitung der Funktionalität der Klasse <code>Queue</code> Modellierung und Implementierung der Anwendung unter Verwendung eines oder mehrerer Objekte der Klasse <code>Queue</code></p> | | <p><i>Beispiel:</i> Patientenwarteschlange (jeder kennt seinen Nachfolger bzw. alternativ: seinen Vorgänger)</p> <p>Sobald ein Patient in einer Arztpraxis eintrifft, werden sein Name und seine Krankenkasse erfasst. Die Verwaltung der Patientenwarteschlange geschieht über eine Klasse, die hier als Wartezimmer bezeichnet wird. Wesentliche Operationen sind das „Hinzufügen“ eines Patienten und das „Entfernen“ eines Patienten, wenn er zur Behandlung gerufen wird.</p> |

| | | |
|--|--|--|
| | | <p>Die Simulationsanwendung stellt eine GUI zur Verfügung, legt ein Wartezimmer an und steuert die Abläufe. Wesentlicher Aspekt des Projektes ist die Modellierung des Wartezimmers mit Hilfe der Klasse Queue. Hier kann fakultativ das Model-View-Controller-Entwurfsmuster behandelt werden.</p> <p>Anschließend kann der Funktionsumfang der Anwendung erweitert werden: Patienten können sich zusätzlich in die Warteschlange zum Blutdruckmessen einreihen. Objekte werden von zwei Schlangen verwaltet.</p> <p><i>Materialien:</i> Zentralabiturklassenbeschreibung und vorgegebene Zentralabiturklasse Queue</p> |
| <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><i>Beispiel:</i> Erweiterung des Wartezimmerprojekts</p> <p>Neben der normalen Warteschlange soll es nun eine weitere geben, die nicht mehr nach dem FIFO-Prinzip organisiert ist, sondern nach Dringlichkeit. Betritt ein neuer Notfallpatient die Praxis muss seine Karte entsprechend der Schwere seines Notfalls einsortiert werden. Erst wenn die Notfallliste leer ist, ist der nächste Patient aus der normalen Liste dran.</p> |

| | | |
|---|--|---|
| | | <p><i>Materialien:</i> Zentralabiturklassenbeschreibung und vorgegebene Zentralabiturklasse List</p> |
| <p>Fakultativ:</p> <p>4. Vertiefung - Anwendungen von Listen, Stapeln oder Schlangen in einem weiteren Kontext</p> | | <p><i>Beispiel:</i> Skispringen</p> <p>Ein Skispringen hat folgenden Ablauf: Nach dem Sprung erhält der Springer eine Punktzahl und wird nach dieser Punktzahl in eine Rangliste eingeordnet. Die besten 30 Springer qualifizieren sich für den zweiten Durchgang. Sie starten in umgekehrter Reihenfolge gegenüber der Platzierung auf der Rangliste. Nach dem Sprung erhält der Springer wiederum eine Punktzahl und wird nach der Gesamtpunktzahl aus beiden Durchgängen in die endgültige Rangliste eingeordnet.</p> <p><i>Beispiel:</i> Terme in Postfix-Notation</p> <p>Die sog. UPN (<i>Umgekehrt-Polnische-Notation</i>) bzw. <i>Postfix-Notation</i> eines Terms setzt den Operator hinter die Operanden. Um einen Term aus der gewohnten Infixschreibweise in einen Term in UPN umzuwandeln oder um den Wert des Terms zu berechnen, kann ein Stack verwendet werden.</p> <p><i>Beispiel:</i> Rangierbahnhof</p> <p>Auf einem Güterbahnhof gibt es drei Gleise, die nur zu einer Seite offen sind. Wagons können also von</p> |

einer Seite auf das Gleis fahren und nur rückwärts wieder hinausfahren. Die Wagons tragen Nummern, wobei die Nummer jedoch erst eingesehen werden kann, wenn der Wagon der vorderste an der offenen Gleisseite ist. (Zwischen den Wagons herumzuturnen, um die anderen Wagonnummern zu lesen, wäre zu gefährlich.) Zunächst stehen alle Wagons unsortiert auf einem Gleis. Ziel ist es, alle Wagons in ein anderes Gleis zu fahren, so dass dort die Nummern der Wagons vom Gleisende aus aufsteigend in richtiger Reihenfolge sind. Zusätzlich zu diesen beiden Gleisen gibt es ein Abstellgleis, das zum Rangieren benutzt werden kann.

Beispiel: Autos an einer Ampel zur Zufahrtsregelung

Es soll eine Ampel zur Zufahrtsregelung in Java simuliert werden. An einem geradlinigen, senkrecht von unten nach oben verlaufenden Straßenstück, das von Autos nur einspurig in eine Richtung befahren werden kann, ist ein Haltepunkt markiert, an dem die Ampel steht. Bei einem Klick auf eine Schaltfläche mit der Aufschrift „Heranfahren“ soll ein neues Auto an den Haltepunkt heranfahren bzw. bis an das letzte Auto, das vor dem Haltepunkt wartet. Grünphasen der Ampel werden durch einen Klick auf eine Schaltfläche mit der Aufschrift „Weiterfahren“

| | | |
|--|--|--|
| | | <p>simuliert. In jeder Grünphase darf jeweils nur ein Auto weiterfahren. Die anderen Autos rücken nach.</p> <p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1-II.3 – Anwendungen für lineare Datenstrukturen</p> |
|--|--|--|

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 aber nicht selbst implementiert. 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.

Sequenzierung des Unterrichtsvorhabens:

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

| | | |
|--|--|--|
| <p>1. Suchen von Daten in Listen und Arrays</p> <p>(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)</p> | <p>Die Schüler*innen</p> <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 unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (A), • 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), • 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><i>Beispiel:</i> Vokabeltrainer</p> <p>Das Programm verwaltet Paare aus deutschen und englischen Vokabeln. Diese werden sortiert (Sortieren durch Einfügen) abgespeichert. Auf diesen Daten kann die passende Vokabel mit linearer und alternativ binärer Suche gesucht werden.</p> <p>Eine weitere Sortierung ergibt sich dadurch, dass eine Lernliste verwaltet wird, in der die Vokabel nach der Häufigkeit der Fehler sortiert sind. Diese Sortierung ändert sich anders als die eigentliche Vokabelliste mit jedem Übungsdurchgang. Hier können alternative Sortieralgorithmen genutzt werden.</p> <p><i>Materialien:</i></p> <p>Zentralabiturklassenbeschreibung und vorgegebene Zentralabiturklasse List Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1.3 - Suchen und Sortieren</p> |
| <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><i>Beispiel:</i> Vokabeltrainer (s.o.)</p> <p>(s.o.)</p> <p>oder</p> <p><i>Beispiel:</i> Bundesligatabelle</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>Vgl. Kapitel „Speichern mit Struktur – Arrays“</p> <p><i>Materialien:</i></p> <p>(s.o.)</p> |
| <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> | | <p><i>Beispiel:</i> Laufzeitversuche an vorgegebenem Programm, welches Zahlen mittels der angegebenen Verfahren sortiert</p> <p>Anschließend: Theoretische Abschätzung der Vergleiche</p> |

Unterrichtsvorhaben Q1-IV:

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 Schüler*innen für sie relevante Fragestellungen, die mit dem vorhandenen Datenbestand beantwortet werden sollen. Zur Beantwortung dieser Fragestellungen wird die vorgegebene Datenbank von den Schüler*innen 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 Schüler*innen 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.

Sequenzierung des Unterrichtsvorhabens

| Unterrichtssequenzen | Zu entwickelnde Kompetenzen | Beispiele, Medien, Materialien |
|---|--|--|
| <p>1. Modellierung von relationalen Datenbanken Teil I</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>Die Schüler*innen</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), ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen (D), | <p><i>Möglicher Einstiege:</i></p> <ul style="list-style-type: none"> <i>Fragestellung:</i> Was stellt ihr euch unter einer Datenbank vor? Wo werden Daten von euch erhoben? <i>Materialien:</i> Eingeführtes Lehrwerk „Informatik 2. Schöningh Verlag“ Kap. 7 Bundesdatenschutzgesetz (BDSG) <p><i>Materialien ER-Diagramme:</i> Eingeführtes Lehrwerk „Informatik 2. Schöningh Verlag“ Kap. 7</p> <p>a</p> |
| <p>2. 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 | <p>(siehe oben)</p> | <p><i>Materialien SQL-Abfragen:</i> https://imoodle.de/sqltutorial/home.html ist ein hervorragend für die Schule geeignetes Tutuorial mit Erklärungen, Beispielen und Online-Aufgaben dabei Möglichkeit, einfach online SQL-Abfragen an verschiedene Beispieldatenbanken anzusetzen</p> |

| | | |
|---|---|--|
| <p>(SELECT (DISTINCT) ...FROM, WHERE, AND, OR, NOT) auf einer Tabelle</p> <ul style="list-style-type: none"> 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> | <ul style="list-style-type: none"> 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. Modellierung von relationalen Datenbanken</p> <p>Teil II</p> <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 | | <p><i>Materialien ER-Diagramme:</i> Eingeführtes Lehrwerk „Informatik 2. Schöningh Verlag“ Kap. 7</p> <p><i>Materialien Datenbankentwurf und Normalisierung:</i> Eingeführtes Lehrwerk „Informatik 2. Schöningh Verlag“ Kap. 7</p> |

| | | |
|--|--|--|
| <ul style="list-style-type: none">• Überprüfung von Datenbankschemata hinsichtlich der 1. bis 3. Normalform und Normalisierung (um Redundanzen zu vermeiden und Konsistenz zu gewährleisten) | | |
|--|--|--|

Unterrichtsvorhaben Q1-V:

Thema: Sicherheit und Datenschutz in Netzstrukturen

Leitfragen: *Wie werden Daten in Netzwerken übermittelt? Was sollte man in Bezug auf die Sicherheit beachten?*

Vorhabenbezogene Konkretisierung:

Anschließend an das vorhergehende Unterrichtsvorhaben zum Thema Datenbanken werden der Datenbankzugriff aus dem Netz, Topologien von Netzwerken, eine Client-Server-Struktur, das TCP/IP-Schichtenmodell sowie Sicherheitsaspekte beim Zugriff auf Datenbanken und verschiedene symmetrische und asymmetrische kryptografische Verfahren analysiert und erläutert. Exemplarisch wird ein einfaches Java-Programm an eine SQL-Datenbank angebunden. Fallbeispiele zur Datenschutzproblematik und zum Urheberrecht runden das Unterrichtsvorhaben ab.

Sequenzierung des Unterrichtsvorhabens:

| Unterrichtssequenzen | Zu entwickelnde Kompetenzen | Beispiele, Medien, Materialien |
|---|--|--|
| <p>1. Daten in Netzwerken und Sicherheitsaspekte in Netzen sowie beim Zugriff auf Datenbanken</p> <p>(a) Beschreibung eines Datenbankzugriffs im Netz anhand eines Anwendungskontextes und einer Client-Server-Struktur zur Klärung der Funktionsweise eines Datenbankzugriffs</p> <p>(b) Netztopologien als Grundlage von Client-Server-Strukturen und TCP/IP-Schichtenmodell als Beispiel für eine Paketübermittlung in einem Netz</p> | <p>Die Schüler*innen</p> <ul style="list-style-type: none"> • beschreiben und erläutern Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken (A), • analysieren und erläutern Eigenschaften und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren (A), • untersuchen und bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen, die Sicherheit von | <p><i>Materialien Netzwerke:</i> Eingeführtes Lehrwerk „Informatik 2. Schöningh Verlag“ Kap. 6</p> |

| | | |
|--|---|---|
| <p>(c) Implementation einer SQL-Anfrage aus einem Java-Programm per Netzwerkzugriff</p> <p>(d) Vertraulichkeit, Integrität, Authentizität in Netzwerken sowie symmetrische und asymmetrische kryptografische Verfahren (Cäsar-, Vigenère-, RSA-Verfahren, Diffie-Hellman) als Methoden Daten im Netz verschlüsselt zu übertragen</p> | <p>Informatiksystemen sowie die Einhaltung der Datenschutzbestimmungen und des Urheberrechts (A),</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), • nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zum Erschließen, zur Aufbereitung und Präsentation fachlicher Inhalte (D). | <p><i>Materialien: SQL-Anfrage per Netzwerkzugriff</i> SQLite (-> Moodle Fachschaftsseiten) Zentralabiturklassenbeschreibung und vorgegebene Zentralabiturklassen DatabaseConnector und QueryResult</p> <p><i>Materialien Kryptographie:</i> Eingeführtes Lehrwerk „Informatik 2. Schöningh Verlag“ S.280ff Informationsblätter Parthe zu Cäsar, Vigenère und Diffie-Hellman (-> Moodle Fachschaftsseiten)</p> |
| <p>2. Fallbeispiele zur Datenschutzproblematik und zum Urheberrecht</p> | | <p><i>Materialien:</i> Eingeführtes Lehrwerk „Informatik 2. Schöningh Verlag“ S. 82ff Bundesdatenschutzgesetz (BDSG)</p> |

Unterrichtsvorhaben Q1-VI

siehe StuBo-Maßnahmenbeschreibung im Anhang

Unterrichtsvorhaben Q1-VII = Q2-I

Unterrichtsvorhaben Q2-I:

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 Baumhalte 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.

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 Schüler*innen</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 | <p><i>Beispiel:</i> Codierungsbäume für Codierungen, deren Alphabet aus genau zwei Zeichen besteht. Morse hat Buchstaben als Folge von Punkten und Strichen codiert. Diese Codierungen können in einem Binärbaum dargestellt werden, so dass ein Übergang zum linken Teilbaum einem Punkt und ein Übergang zum rechten Teilbaum einem Strich entspricht. Wenn man im Gesamtbaum startet und durch Übergänge zu linken oder rechten Teilbäumen einen Pfad zum gewünschten Buchstaben sucht, erhält man die Morsecodierung des Buchstabens.</p> <p>oder</p> <p><i>Beispiel:</i> Termbaum Der Aufbau von Termen wird mit Hilfe von binären Baumstrukturen verdeutlicht.</p> <p>oder</p> <p><i>Beispiel:</i> Ahnenbaum Die binäre Baumstruktur ergibt sich daraus, dass jede Person genau einen Vater und eine Mutter hat.</p> <p><i>Weitere Beispiele für Anwendungskontexte für binäre Bäume:</i></p> |

| | | |
|---|--|--|
| | <p>Konstruktionsstrategien „Modularisierung“ und „Teilen und Herrschen“ (M),</p> <ul style="list-style-type: none"> • implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), • modifizieren Algorithmen und Programme (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), • stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D). | <p><i>Beispiel:</i> Suchbäume (zur sortierten Speicherung von Daten) Alle Inhalte, die nach einer Ordnung vor dem Inhalt im aktuellen Teilbaum stehen, sind in dessen linkem Teilbaum, alle die nach dem Inhalt im aktuellen Teilbaum stehen, sind in dessen rechtem Teilbaum. (Dies gilt für alle Teilbäume.)</p> <p><i>oder</i> <i>Beispiel:</i> Entscheidungsbäume Um eine Entscheidung zu treffen, werden mehrere Fragen mit ja oder nein beantwortet. Die Fragen, die möglich sind, wenn die Antwort auf eine Frage mit „ja“ beantwortet wird, befinden sich im linken Teilbaum, die Fragen, die möglich sind, wenn die Antwort „nein“ lautet, stehen im rechten Teilbaum.</p> <p><i>Materialien:</i> Eingeführtes Lehrwerk „Informatik 2. Schöningh Verlag“ Kap. 5</p> |
| <p>2. Die Datenstruktur Binärbaum im Anwendungskontext unter Nutzung der Klasse <code>BinaryTree</code></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><i>Beispiel:</i> MorseCodierer (s.o.) implementieren</p> <p><i>Materialien:</i> Zentralabiturklassenbeschreibung und vorgegebene Zentralabiturklasse <code>BinaryTree</code></p> <p>Eingeführtes Lehrwerk „Informatik 2. Schöningh Verlag“ Kap. 5</p> |

| | | |
|--|--|---|
| <p>(c) Erarbeitung der Klasse <code>BinaryTree</code> und beispielhafte Anwendung der Operationen</p> <p>(d) Implementierung der Anwendung oder von Teilen der Anwendung</p> <p>(e) Traversierung eines Binärbaums im Inorderdurchlauf (Übung Pre- und Postorderdurchlauf)</p> | | |
| <p>3. Die Datenstruktur binärer Suchbaum im Anwendungskontext unter Verwendung der Klasse <code>BinarySearchTree</code></p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften 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 <code>BinarySearchTree</code> und Einführung des Interface <code>Item</code> zur Realisierung einer geeigneten Ordnungsrelation</p> <p>(d) Implementierung der Anwendung oder von Teilen der Anwendung inklusive einer sortierten Ausgabe des Baums</p> | | <p><i>Beispiel:</i> Informatikerbaum als <i>Suchbaum</i> In einem binären <i>Suchbaum</i> werden die Namen und die Geburtsdaten von Informatikern lexikographisch geordnet abgespeichert. Alle Namen, die nach dieser Ordnung vor dem Namen im aktuellen Teilbaum stehen, sind in dessen linkem Teilbaum, alle die nach dem Namen im aktuellen Teilbaum stehen, sind in dessen rechtem Teilbaum. (Dies gilt für alle Teilbäume.) Folgende Funktionalitäten werden benötigt:</p> <ul style="list-style-type: none"> • Einfügen der Informatiker-Daten in den Baum • Suchen nach einem Informatiker über den Schlüssel Name • Ausgabe des kompletten Datenbestands in nach Namen sortierter Reihenfolge <p><i>Materialien:</i> Zentralabiturklassenbeschreibung und vorgegebene Zentralabiturklasse <code>BinarySearchTree</code> und Interface <code>ComparableContent</code> Eingeführtes Lehrwerk „Informatik 2. Schöningh Verlag“ Kap. 5</p> |



| | | |
|--|--|--|
| | | |
|--|--|--|

| | | |
|--|--|--|
| <p>4. Übung und Vertiefungen der Verwendung von Binärbäumen oder binären Suchbäumen anhand weiterer Problemstellungen</p> | | <p><i>Beispiel:</i> Codierungsbäume (s.o.) oder Huffman-Codierung</p> <p><i>oder</i></p> <p><i>Beispiel:</i> Buchindex Es soll eine Anwendung entwickelt werden, die anhand von Stichworten und zugehörigen Seitenzahlen ein Stichwortregister erstellt. Da die Stichwörter bei der Analyse des Buches häufig gesucht werden müssen, werden sie in der Klasse <code>Buchindex</code> als Suchbaum (Objekt der Klasse <code>BinarySearchTree</code>) verwaltet. Alle Inhalte, die nach einer Ordnung vor dem Inhalt im aktuellen Teilbaum stehen, sind in dessen linkem Teilbaum, alle die nach dem Inhalt im aktuellen Teilbaum stehen, sind in dessen rechtem Teilbaum. (Dies gilt für alle Teilbäume.)</p> <p><i>oder</i></p> <p><i>Beispiel:</i> Entscheidungsbäume (s.o.)</p> <p><i>oder</i></p> <p><i>Beispiel:</i> Termbaum (s.o.)</p> <p><i>oder</i></p> <p><i>Beispiel:</i> Ahnenbaum (s.o.)</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.

Sequenzierung des Unterrichtsvorhabens:

| Unterrichtssequenzen | Zu entwickelnde Kompetenzen | Beispiele, Medien oder Materialien |
|--|---|--|
| <p>1. Endliche Automaten</p> <p>(a) Vom Automaten in den Schüler*innen bekannten Kontexten zur formalen Beschreibung eines endlichen Automaten</p> <p>(b) Untersuchung, Darstellung und Entwicklung endlicher Automaten</p> | <p>Die Schüler*innen</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), • 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), • 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 die jeweils andere Darstellungsform (D), | <p><i>Beispiele:</i> Cola-Automat, Geldspielautomat, Roboter, Zustandsänderung eines Objekts „Auto“, Akzeptor für bestimmte Zahlen, Akzeptor für Teilwörter in längeren Zeichenketten, Akzeptor für Terme</p> <p><i>Materialien:</i> Eingeführtes Lehrwerk „Informatik 2. Schöningh Verlag“ Kap. 4</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> | <p>(siehe oben)</p> | <p><i>Beispiele:</i> reguläre Grammatik für Wörter mit ungerader Parität, Grammatik für Wörter, die bestimmte Zahlen repräsentieren, Satzgliederungsgrammatik</p> <p><i>Materialien:</i> Eingeführtes Lehrwerk „Informatik 2. Schöningh Verlag“ Kap. 4</p> |
| <p>3. Grenzen endlicher Automaten</p> | | <p><i>Beispiele:</i></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). | Klammerausdrücke, $a^n b^n$ im Vergleich zu $(ab)^n$ <i>Materialien:</i> Eingeführtes Lehrwerk „Informatik 2. Schöningh Verlag“ Kap. 4 |
|--|---|--|

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.

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>c) Analyse und Erläuterung der Funktionsweise eines einfachen maschinennahen Programms</p> | <p>Die Schüler*innen</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 eingegeben Zahl mit einem Rechnermodell</p> <p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.3 –Von-Neumann-Architektur und maschinennahe Programmierung (Download Q2-III.1)</p> |

| | | |
|--|--|---|
| <p>2. Grenzen der Automatisierbarkeit</p> <ul style="list-style-type: none">a) Vorstellung des Halteproblemsb) Unlösbarkeit des Halteproblemsc) Beurteilung des Einsatzes von Informatiksystemen hinsichtlich prinzipieller Möglichkeiten und prinzipieller Grenzen | | <p><i>Beispiel:</i> Halteproblem</p> <p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.3 - Halteproblem (Download Q2-III.2)</p> |
|--|--|---|

Unterrichtsvorhaben Q2-IV:

Wiederholung und Vertiefung ausgewählter Kompetenzen und Inhalte der Qualifikationsphase

Hier sollten insbesondere auch Abituraufgaben aus den Vorjahren exemplarisch behandelt werden. Teile der darin enthaltenen Szenarien können auch implementiert werden.

2.3 Grundsätze der fachmethodischen und –didaktischen Arbeit

Auszug aus dem Kernlehrplan für die Sek. II aus dem Jahr 2014:

„Ausgangspunkt im Informatikunterricht ist häufig ein Problem mit lebensweltlichem Bezug. Schüler*innen erwerben und erweitern in der aktiven Auseinandersetzung mit komplexen Problemstellungen Kompetenzen, die sie zum selbstständigen informatischen Problemlösen befähigen. Mit der Aneignung von Strategien und Techniken zur strukturierten Zerlegung im Problemlöseprozess, zur Algorithmisierung von Abläufen sowie zur formalsprachlichen, grafischen oder symbolischen Beschreibung von Sachverhalten und Zusammenhängen erwerben die Lernenden Kompetenzen zur Bewältigung von Komplexität. Die Konstruktion eines abstrakten Modells zu einer anwendungsbezogenen Problemstellung fördert das Abstraktionsvermögen sowie kreatives und strukturelles Denken. Die Umsetzung eines informatischen Modells in ein lauffähiges Informatiksystem hat für Schüler*innen nicht nur einen hohen Motivationswert, sondern ermöglicht ihnen auch die eigenständige Überprüfung der Angemessenheit und Wirkung des Modells im Rückbezug auf die Problemstellung. Im Unterricht lassen sich umfangreiche Informatiksysteme nur in arbeitsteiliger projektorientierter Zusammenarbeit im Team erstellen. Solche Projekte können nur gelingen, wenn die gemeinsame Arbeit strukturiert geplant und organisiert wird. Insgesamt leistet das Fach Informatik in der gymnasialen Oberstufe damit einen wichtigen Beitrag zu einer erweiterten Allgemeinbildung und allgemeinen Studierfähigkeit der Schüler*innen.

Die inhaltliche und methodische Gestaltung des Unterrichts ist entscheidend dafür, dass Schüler*innen die ausgewiesenen Kompetenzen erwerben können. Informatikunterricht erfordert in hohem Maße die Anwendung schüleraktivierender Methoden, die selbstständiges Lernen ermöglichen und individuelle Förderung begünstigen. Unterschiedliche, auch geschlechtsspezifische Herangehensweisen, Interessen, Vorerfahrungen und fachspezifische Kenntnisse sind angemessen zu berücksichtigen. In der Einführungsphase werden ausgehend von einfachen Fragestellungen und unter Anleitung der Lehrperson zunächst einzelne Stufen eines Problemlösungsprozesses durchlaufen. Am Ende der Qualifikationsphase sollen die Schüler*innen dann in der Lage sein, Lösungsansätze in Form von lauffähigen Informatiksystemen weitgehend selbstständig zu entwickeln.“

In den vergangenen Jahren hat sich gezeigt, dass in der Einführungsphase genügend Zeit für große Softwareentwicklungsprojekte zur Verfügung steht. Nach einer Anfangsphase, in der in die Programmierung eingeführt wird, sollten die Schüler*innen auf jeden Fall Zeit für ein oder zwei größere Softwareentwicklungsprojekte bekommen. Dort erfolgt die Entwicklung zwar nicht immer gradlinig und systematisch, aber die Schüler*innen haben so Zeit, eigenständig algorithmische Ideen auszuprobieren, im Austausch zu erkennen, wieso Lösungsansätze funktionieren oder nicht zielführend sind.

In der Qualifikationsphase werden die zeitlichen Spielräume für größere Projekte nach und nach kleiner und es wird immer stärker auf die Aufgabenformate des Zentralabiturs hingearbeitet. Hierbei ist darauf zu achten, dass es ausreichend Übungsphasen gibt, in denen die Schüler*innen möglichst selbstständig Beispielaufgaben lösen, deren Formulierung und Komplexität sich im Laufe Zeit an die Anforderungen des Zentralabiturs annähert. In der Wiederholungsphase sollen dann verstärkt Originalabituraufgaben aus den Vorjahren eigenständig bearbeitet werden.

2.4 Grundsätze der Leistungsbewertung und Leistungsrückmeldung

Es gelten die Vorgaben des „Kernlehrplan für die Sekundarstufe II Gymnasium/Gesamtschule in NRW“ aus dem Jahr 2014 (siehe dort die Seiten 40 – 47).

Darauf aufbauend hat sich die Fachschaft folgende Konkretisierungen vereinbart:

Im Bereich der SONSTIGEN MITARBEIT fließen insbesondere die folgenden Beobachtungen ein:

- die **regelmäßige Beteiligung am Unterrichtsgespräch**. Dabei gehen insbesondere die Wiedergabe von Kenntnissen, Darstellung von Problemsituationen, Beiträge zur Entwicklung von Problemlösungen, Vorstellung und Diskussion von Arbeitsergebnissen aus Gruppen- oder Partnerarbeit und die Klarheit der Darstellungen in die Bewertung mit ein. SchülerInnen mit Vorkenntnissen sind aufgefordert, sich so zu äußern, dass ihnen alle folgen können. Zunehmend wird eine angemessene Verwendung der Fachsprache erwartet. Keinesfalls kann es darum gehen, immer sofort fehlerfreie Lösungen vorzustellen. Auch ein fehlerhafter Ansatz kann von der Tiefe der Auseinandersetzung zeugen und bringt den Unterricht oft besser voran als eine perfekte Lösung.
- die **Präsentation und Erläuterung der erstellten Programme/Programmteile**.
- **selbstständiges Arbeiten zu zweit oder in Gruppen** (mit und ohne Rechner). Dabei wird auch beurteilt, wie diese Arbeit organisiert wird. (Absprachen, Arbeitseinteilung, gegenseitige Hilfen)
- die **Beratung anderer Teams** (insbesondere in einem großen Kurs ist die gegenseitige Unterstützung für den gemeinsamen Fortschritt im Kurs von sehr großer Bedeutung)
- die **selbstständige und intensive Auseinandersetzung mit schriftlichen Informationen und Aufgabenstellungen**.
- die **Anfertigung und Präsentation von Hausaufgaben**. (Sek-II)
- die **selbstständige Weiterarbeit und Vervollständigung von Programmierlösungen**
- das **Erstellen von Protokollen**
- **Referate**
- nach ausreichender Eingewöhnungszeit: **Umgang mit den technischen Systemen** (insbesondere der Umgang mit der Programmierumgebung, angemessene Reaktion auf Fehlermeldungen usw.)
- einzelne Sachverhalte können in kurzen schriftlichen Überprüfungen abgefragt werden

Für Klausuren wurden ergänzend zu den allgemeinen Vorgaben folgende Vereinbarungen getroffen:

- Es werden nur ganzzahlige Punktevergeben.
- Die Zuordnung der Notenstufen zu den erreichten Punkten erfolgt gemäß der schriftlichen Abiturprüfung, d.h.:

| Notenpunkte | Note | ab |
|-------------|------|-----|
| 15 | 1+ | 95% |
| 14 | 1 | 90% |
| 13 | 1- | 85% |
| 12 | 2+ | 80% |
| 11 | 2 | 75% |
| 10 | 2- | 70% |
| 9 | 3+ | 65% |
| 8 | 3 | 60% |
| 7 | 3- | 55% |
| 6 | 4+ | 50% |
| 5 | 4 | 45% |
| 4 | 4- | 40% |
| 3 | 5+ | 33% |
| 2 | 5 | 27% |
| 1 | 5- | 20% |
| 0 | 6 | 0% |

Für die Leistungsbewertung während des „Distanzlernens“ gelten die allgemeinen Grundsätze, die im CBG-eigenem „Konzept zum Distanzlernen“ niedergeschrieben sind.

Für das Fach Informatik werden die Grundsätze wie folgt präzisiert:

Im Informatikunterricht wird häufig auf technische Werkzeuge (PC und Tablet mit entsprechender Software) zurückgegriffen, die den Schüler*innen zuhause nur zum Teil oder gar nicht zur Verfügung stehen. Dies muss berücksichtigt werden. Die Aufgaben/Arbeitsaufträge müssen so gestellt werden, dass sie auch ohne diese technischen Werkzeuge „mit Papier und Stift“ zu bearbeiten sind. Abweichend davon dürfen in den Jahrgängen, die „1-zu-1“ mit einem von der Schule verwalteten Tablet ausgestattet sind, Aufgaben/Arbeitsaufträge gestellt werden, die mit dem Tablet und den darauf durch die Schule zur Verfügung gestellten Apps zu bearbeiten sind. Es darf dabei ein Zugriff auf das Internet vorausgesetzt werden. Sollten die notwendigen technischen Voraussetzungen bei einer Schülerin bzw. einem Schüler nicht gegeben sein, muss sich die Schülerin bzw. der Schüler an die Fachlehrkraft wenden. Sollten die technischen Voraussetzungen nicht hergestellt werden, ist die Aufgabe bzw. der Arbeitsauftrag so anzupassen, dass der Schülerin bzw. dem Schüler keine Nachteile entstehen.

2.5 Lehr- und Lernmittel

In der Jahrgangsstufe EF ist das Lehrwerk „Informatik - Lehrwerk für die gymnasiale Oberstufe - Ausgabe 2014 - Schülerband 1 Einführungsphase“ (ISBN 978-3-14-037126-1) des Schöningh-Verlags eingeführt.

In den Jahrgangsstufe Q1 und Q2 ist das Lehrwerk „Informatik - Lehrwerk für die gymnasiale Oberstufe - Ausgabe 2014 - Schülerband 2 Qualifikationsphase“ (ISBN 978-3-14-037127-8) des Schöningh-Verlags eingeführt.

3. Entscheidungen zu fach- oder unterrichtsübergreifenden Fragen

Die Fachschaft Informatik steht im ständigen Austausch mit den Fachschaften Mathematik und Physik. Alle Informatiklehrkräfte unterrichten ebenfalls das Fach Mathematik und arbeiten dort aktiv an der Fortentwicklung des Curriculums mit.

Es werden in der Q1 Projektkurse mit den Fächern Mathematik und Physik angeboten.

Im AG-Bereich ergänzen sich die Fächer Informatik und Physik bei den AGs „Lego-Roboter“ (Klassen 5 und 6) und „Arduino“ (Klassen 7-9) hervorragend. Die AGs werden im unregelmäßigen Wechsel von Informatik- und Physiklehrkräften unterrichtet.

Alle zwei Jahre findet am CBG in der Woche vor den Sommerferien eine Projektwoche statt. Dort hat es in der Vergangenheit gemeinsam von Informatik- und Physiklehrkräften geleitete Projektangebote gegeben. Dies ist auch für die kommenden Projektwochen geplant.

Und schließlich gibt es am CBG einen MINT-Koordinator, der den Austausch zwischen den Fachgruppen der MINT-Fachschaften begleitet.

4. Qualitätssicherung und Evaluation

Innerhalb der kleinen Fachschaft Informatik findet ein kontinuierlicher und intensiver Austausch über die Erfahrungen mit Unterrichtsinhalten, Überprüfungsformen und Werkzeugen statt. Klausuren werden regelmäßig ausgetauscht. Sämtliche Fachschaftsmitglieder prüfen seit Jahren gemeinsam im mündlichen Abitur. In diesem Zusammenhang findet der Austausch besonders intensiv statt.

Die Ergebnisse der Abiturprüfungen werden in der Fachkonferenz Informatik analysiert und bewertet.

An Fortbildungen nehmen in der Regel mindestens zwei Mitglieder der Fachschaft teil. Über die Fortbildung findet jeweils kurzfristig ein Austausch statt.

Das vorliegende Curriculum ist das Resultat eines regelmäßigen Anpassungsprozesses in der Änderungen in den Vorgaben eingearbeitet und auf Grundlage der Diskussion in der Fachschaft Optimierungen vorgenommen wurden.

5. Anhang

Maßnahmenkarte zur Berufs- und Studienorientierung

Fach: Informatik

Stand: 14.04.2020

| | |
|---|---|
| Bezeichnung der Maßnahme | Vorstellung von Ausbildungsberufen und Studiengängen im Bereich IT |
| Dauer | 2-4 UStd. |
| Jahrgangsstufe | Q1 (bei Beteiligung externer ggf. auch jahrgangsstufenübergreifend inkl. der Q2) |
| Phase(n) im BO-Prozess | <input type="checkbox"/> Potenziale erkennen <input checked="" type="checkbox"/> Berufsfelder kennen lernen <input type="checkbox"/> Praxis erproben <input type="checkbox"/> Hochschulen und Studienwege erkunden <input type="checkbox"/> Entscheidung <input type="checkbox"/> Übergang gestalten <input type="checkbox"/> Sonstiges |
| zu fördernde BSO-Kompetenzen | BES1, BEM1 |
| weitere, zu fördernde Kompetenzen | BEM2 |
| verantwortlich | in der Schule: Fachlehrer ggf. beim Kooperationspartner (inkl. Kontaktdaten): keine feste Kooperation, Fachlehrer hat in der Vergangenheit den Kontakt zu den betreffenden ehemaligen S* gehalten |
| Beschreibung der Durchführung | Die S* erarbeiten Präsentationen zu Studienfächern und Ausbildungsberufen im Bereich IT. Sofern möglich stellen auch ehemalige S* ihren Ausbildungsberuf bzw. ihr Studienfach vor. Letzteres ist besonders wünschenswert und hat auch schon mehrfach funktioniert. |
| Vernetzung mit vorherigen und nachfolgenden Maßnahmen | Unterrichtliche Vor- und Nachbereitung: Beratungsaktivitäten (z. B. durch Berufsberatung, sozialpädagogische Fachkräfte): Praxiskontakte (z. B. Betriebspraktikum, Erkundung einer Berufsmesse): |
| Evaluation/ Erfolgskriterien | Rückmeldung bei den betroffenen S* einholen |