

Informatik: Schulinterner Lehrplan zum Kernlehrplan für die gymnasiale Oberstufe am Städtischen Gymnasium Bergkamen

Das vorliegende Dokument orientiert sich an dem Beispiel für schulinterne Lehrpläne im Fach Informatik¹ und wird hier für das Städtische Gymnasium Bergkamen ausgestaltet.

Inhaltsverzeichnis

1 Fachgruppe Informatik am Städtischen Gymnasium Bergkamen	2
2 Entscheidungen zum Unterricht	2
2.1 Unterrichtsvorhaben	3
2.1.1 Übersichtsraster Unterrichtsvorhaben	3
2.1.2 Konkretisierte Unterrichtsvorhaben	12
2.2 Grundsätze der fachmethodischen und fachdidaktischen Arbeit	40
2.3 Grundsätze der Leistungsbewertung und Leistungsrückmeldung	41
2.3.1 Beurteilungsbereich Klausuren	42
2.3.2 Beurteilungsbereich »Sonstige Mitarbeit«	42
3 Qualitätssicherung und Evaluation	46
Anhang	47
A Hinweise auf konkrete Materialien, Werkzeuge, . . .	47
B Mitarbeitende	51

¹(SILP-IF-2014) – <http://www.standardsicherung.schulministerium.nrw.de/lehrplaene/lehrplannavigator-s-ii/gymnasiale-oberstufe/informatik/hinweise-und-beispiele/hinweise-und-beispiele.html>

1 Fachgruppe Informatik am Städtischen Gymnasium Bergkamen

Beim Städtischen Gymnasium Bergkamen handelt es sich um ein voll ausgebautes Gymnasium mit gymnasialer Oberstufe, die ungefähr 400 Schülerinnen und Schüler umfasst.

„Der Umgang mit den neuen Medien ist fester Bestandteil des Unterrichts an unserer Schule“, und ausdrücklich im Schulprogramm verankert. „Die Schülerinnen und Schüler sollen befähigt werden, kreativ und sozial verantwortlich mit ihnen umzugehen, auf elektronischem Wege mit anderen Menschen zu kommunizieren und in Projekten zusammenzuarbeiten.“ Der Informatikunterricht im Allgemeinen und in der gymnasialen Oberstufe im Speziellen soll und kann einen wesentlichen Teil zu diesen schulischen Grundsätzen beitragen.

Durch projektorientiertes Vorgehen, offene und lebensnahe Problem- und Aufgabenstellungen sowie die Vermittlung von Vorgehensweisen und Hinweise zur Verfeinerung und Optimierung bestehender Problemlösungen entspricht der Informatikunterricht der Oberstufe in besonderem Maße dem Erziehungsziel Leistungsbereitschaft zu fördern ohne zu überfordern, und bietet durch nahezu unbegrenzte Skalierbarkeit der Aufgabenstellungen vielfältige Möglichkeiten zur Differenzierung.

Der regelmäßige Austausch von Materialien und Unterrichtsvorhaben zwischen den Kollegen des Fachbereichs Informatik, sowie die stete, gemeinsame Evaluation der Lehr- und Lernprozessen stellen einen wichtigen Beitrag zur Qualitätssicherung und -entwicklung des Unterrichts dar.

Derzeit unterrichten am Städtischen Gymnasium Bergkamen zwei Lehrer das Fach Informatik, Herr Kelter und Herr Brinkmann. Den Fachvorsitz führt derzeit Herr Brinkmann. Für die Arbeit am Computer stehen dem Informatikunterricht drei Computerräume mit 12, 13 und 15 Schülerarbeitsplätzen zur Verfügung, wobei der Informatikunterricht bei der Zuordnung der Computerräume gegenüber Unterrichtsfächern, die nicht so stark an den Einsatz von Computern gekoppelt sind, üblicherweise Vorrang hat. Die Größe der Lerngruppen im Fach Informatik beträgt in der Oberstufe üblicherweise 15 bis 20 Schülerinnen und Schüler, wovon einige das Fach schriftlich und teils auch im schriftlichen oder mündlichen Abitur belegen.

2 Entscheidungen zum Unterricht

Die Entscheidungen zum Unterricht sind für Grund- und Leistungskurse. Alle Angaben für die Leistungskurse sind dabei **fett** und *kursiv* gesetzt, um die Unterschiede zu verdeutlichen.

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.

Im »Übersichtsraster Unterrichtsvorhaben« (Abschnitt 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. Um Freiraum für Vertiefungen, besondere Schülerinteressen, aktuelle Themen bzw. die Erfordernisse anderer besonderer Ereignisse (z. B. Praktika, Kursfahrten o. ä.) zu erhalten, wurden im Rahmen dieses schulinternen Lehrplans ca. 75 Prozent der Bruttounterrichtszeit verplant.

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« (Abschnitt 2.1.2) Beispiele und Materialien, die empfehlenden Charakter haben. Referendarinnen und Referendaren sowie neuen Kolleginnen und Kollegen dienen diese vor allem zur standardbezogenen Orientierung in der neuen Schule, aber auch zur Verdeutlichung von unterrichtsbezogenen fachgruppeninternen Absprachen zu didaktisch-methodischen Zugängen, fächerübergreifenden Kooperationen, Lernmitteln und -orten sowie vorgesehenen Leistungsüberprüfungen, die im Einzelnen auch den Abschnitten 2.2 bis 2.3 zu entnehmen sind.

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

Bei den Übersichten werden die Bezeichnungen für die Inhaltsfelder in der Überschrift komplett ausgewiesen, auch wenn in den Unterrichtsvorhaben nur Teilaspekte behandelt werden.

1) Einführungsphase

Einführungsphase	
<p>Unterrichtsvorhaben E-1</p> <p>Thema: Informatik – Fachgebiete, typische Problemstellungen, Geschichte</p> <p>Kompetenzbereiche (Prozesse):</p> <ul style="list-style-type: none">• Argumentieren• Modellieren• Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none">• Daten und ihre Strukturierung• Algorithmen• Informatiksysteme• Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none">• Betriebssysteme: Dateisystem• Informatiksysteme• Automatisierung durch Formalisierung• Geschichte der Informatik• Digitalisierung• Struktureller Aufbau und Arbeitsweise singulärer Rechner am Beispiel der Von-Neumann-Architektur <p>Zeitbedarf: 10 Unterrichtsstunden</p>	<p>Unterrichtsvorhaben E-2</p> <p>Thema: Grundlagen der objektorientierten Analyse und Modellierung anhand von Beispielkontexten</p> <p>Kompetenzbereiche (Prozesse):</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• Formale Sprachen und Automaten <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none">• Objekte und ihre Darstellung• Syntax und Semantik einer Programmiersprache <p>Zeitbedarf: 20 Unterrichtsstunden</p>

Einführungsphase

Unterrichtsvorhaben E-3

Thema: Datenschutz aus informati-
scher Perspektive

Kompetenzbereiche (Prozesse):

- Argumentieren
- Darstellen und Interpretieren
- Kommunizieren und Kooperieren

Inhaltsfelder:

- Daten und ihre Strukturierung
- Algorithmen
- Informatiksysteme
- Informatik, Mensch und Gesellschaft

Inhaltliche Schwerpunkte:

- Wirkung der Automatisierung

Zeitbedarf: 9 Unterrichtsstunden

Unterrichtsvorhaben E-4

Thema: Strukturierung und Orga-
nisation von Daten

Kompetenzbereiche (Prozesse):

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

Inhaltsfelder:

- Algorithmen
- Daten und ihre Strukturierung

Inhaltliche Schwerpunkte:

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

Zeitbedarf: 10 Unterrichtsstunden

Einführungsphase

Unterrichtsvorhaben E-5

Thema: Klassen und ihre Implementierung durch Umsetzung einer objektorientierten Modellierung

Kompetenzbereiche (Prozesse):

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

Inhaltsfelder:

- Daten und ihre Strukturierung
- Formale Sprachen und Automaten

Inhaltliche Schwerpunkte:

- Objekte und Klassen
- Analyse, Entwurf und Implementierung einfacher Algorithmen
- Syntax und Semantik einer Programmiersprache
- Dateisystem
- Einsatz von Informatiksystemen

Zeitbedarf: 20 Unterrichtsstunden

Unterrichtsvorhaben E-6

Thema: Modellierung von Abläufen – Kontrollstrukturen

Kompetenzbereiche (Prozesse):

- Argumentieren
- Modellieren
- Implementieren

Inhaltsfelder:

- Algorithmen
- Formale Sprachen und Automaten

Inhaltliche Schwerpunkte:

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

Zeitbedarf: 6 Unterrichtsstunden

Summe Einführungsphase: 75 Unterrichtsstunden

II) Qualifikationsphase (Q1 und Q2)

Qualifikationsphase	
<p>Unterrichtsvorhaben Q1-1</p> <p>Thema: Verwaltung von Daten in linearen Datenstrukturen</p> <p>Kompetenzbereiche (Prozesse):</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• Algorithmen in ausgewählten informatischen Kontexten• Syntax und Semantik einer Programmiersprache• Nutzung von Informatiksystemen <p>Zeitbedarf: 15 Unterrichtsstunden</p>	<p>Unterrichtsvorhaben Q1-2</p> <p>Thema: Suchen und Sortieren auf linearen Datenstrukturen</p> <p>Kompetenzbereiche (Prozesse):</p> <ul style="list-style-type: none">• Argumentieren• Modellieren• Implementieren• Darstellen und Interpretieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none">• Algorithmen• Formale Sprachen und Automaten• Informatiksysteme <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• Nutzung von Informatiksystemen <p>Zeitbedarf: 15 Unterrichtsstunden</p>

Qualifikationsphase

Unterrichtsvorhaben Q1-3

Thema: Bäume

Kompetenzbereiche (Prozesse):

- 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
- Algorithmen in ausgewählten informatischen Kontexten
- Syntax und Semantik einer Programmiersprache
- Nutzung von Informatiksystemen

Zeitbedarf: 15 Unterrichtsstunden

Unterrichtsvorhaben Q1-4

Thema: Graphen

Kompetenzbereiche (Prozesse):

- 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
- Algorithmen in ausgewählten informatischen Kontexten
- Syntax und Semantik einer Programmiersprache
- Nutzung von Informatiksystemen

Zeitbedarf: 5 Unterrichtsstunden

Qualifikationsphase

Unterrichtsvorhaben Q1-5

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

Kompetenzbereiche (Prozesse):

- 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: 15 Unterrichtsstunden

Unterrichtsvorhaben Q1-6

Thema: Datenbanken

Kompetenzbereiche (Prozesse):

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

Inhaltsfelder:

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

Inhaltliche Schwerpunkte:

- Datenbanken
- Analyse, Entwurf und Implementierung von Algorithmen
- Algorithmen in ausgewählten informatischen Kontexten
- Syntax und Semantik einer Programmiersprache
- Nutzung von Informatiksystemen
- Sicherheit
- Wirkung der Automatisierung

Zeitbedarf: 15 Unterrichtsstunden

Summe Qualifikationsphase 1: 80 Unterrichtsstunden

Qualifikationsphase

Unterrichtsvorhaben Q2-1

Thema: Endliche Automaten und formale Sprachen

Kompetenzbereiche (Prozesse):

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

Inhaltsfelder:

- Formale Sprachen und Automaten

Inhaltliche Schwerpunkte:

- Endliche Automaten *und Kellerautomaten*
- Grammatiken regulärer Sprachen *und kontextfreier Sprachen*
- *Scanner, Parser und Interpreter für eine reguläre Sprache*
- Möglichkeiten und Grenzen von Automaten und formalen Sprachen

Zeitbedarf: 20 Unterrichtsstunden

Unterrichtsvorhaben Q2-2

Thema: Kommunikation in und Aufbau von Netzwerken

Kompetenzbereiche (Prozesse):

- 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: 20 Unterrichtsstunden

Qualifikationsphase

Unterrichtsvorhaben Q2-3

Thema: Sicherheit in der Informatik: Verschlüsselung und ihre Folgen

Kompetenzbereiche (Prozesse):

- 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
- Sicherheit
- Wirkung der Automatisierung
- Grenzen der Automatisierung

Zeitbedarf: 20 Unterrichtsstunden

Unterrichtsvorhaben Q2-4

Summe Qualifikationsphase 2: 60 Unterrichtsstunden

2.1.2 Konkretisierte Unterrichtsvorhaben

Für die Arbeit in den Informatikkursen der gymnasialen Oberstufe werden Richtlinien zur Dokumentation von Ergebnissen entwickelt. Diese werden in der eigenständigen Materialsammlung gepflegt.

In allen Konkretisierungen werden die Kompetenzen zum Bereich Kommunizieren und Kooperieren vertieft. Um die Lesbarkeit des Dokumentes zu erhöhen, werden sie bei den Unterrichtsvorhaben nicht gesondert aufgeführt.

Einführungsphase

Unterrichtsvorhaben EF – 1

Thema: Informatik – Fachgebiete, typische Problemstellungen, Geschichte

Im Unterschied zu anderen Schulfächern ist Informatik nicht in der Stundentafel der Sekundarstufe I verankert und damit vielen Schülerinnen und Schülern nicht als Fach bekannt. Die dokumentierten Vorstellungen von Informatik (nicht nur) bei Schülerinnen und Schülern sind nicht nur vielfältig, sondern erweisen sich häufig als fehlerhaft. Eine Folge sind Fehlwahlen bei Informatikstudierenden, die bis zum Studienabbruch führen können.

Daher ist eine fachliche Klärung unabdingbar. Sie betrifft zum Einen die Darstellung der sechs Fachgebiete der Informatik und zum Anderen die Arbeit an ausgewählten typischen Problemen der Informatik mit dem Ziel, die Breite der Wissenschaft Informatik für die Schülerinnen und Schülern zugänglich zu machen.

In diesem Unterrichtsvorhaben soll die Benutzung von Informatiksystemen weitgehend vermieden werden (auch nicht zur Recherche). Vielmehr bietet sich an, die ausgewählten prototypischen Problemstellungen als Freihandversuche (mit Papier und Bleistift) durchzuführen und zu dokumentieren.

Objekte werden implizit benutzt, um die Schülerinnen und Schüler auf das folgende Vorhaben vorzubereiten und eine objektorientierte Denkweise zu fördern. Der Objektbegriff sollte jedoch zur Trennung der Vorhaben noch nicht eingeführt werden.

Viele Fachkonzepte, die in der Schulinformatik thematisiert werden, können als fundamentale Ideen der Informatik auf eine lange Geschichte in der Fachwissenschaft zurückblicken. Die geschichtliche Dimension der Entwicklung der Konzepte und Strategien, mit denen die Informatik in alle Wissens- und Lebensbereiche erfolgreich vorstoßen konnte, verdeutlichen einerseits die prinzipiellen Möglichkeiten geeigneter Abstraktionsmechanismen. Diese Abstraktionsmechanismen wurden und werden von der Fachwissenschaft Informatik entwickelt und in Form von Informatiksystemen bereitgestellt. Nahezu alle Bereiche des menschlichen Lebens können durch geeignete Daten und Algorithmen repräsentiert werden.

Sequenzierung des Unterrichtsvorhabens

Kompetenzen

Die Schülerinnen und Schüler

- ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (IF1, M).
- entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (IF2, M).
- nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung und gemeinsamen Verwendung von Daten unter Berücksichtigung der Rechteverwaltung (IF4, K).
- beschreiben und erläutern den strukturellen Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der Von-Neumann-Architektur (IF4, A).
- bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (IF5, A).
- erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (IF5, A).
- stellen ganze Zahlen und Zeichen in Binärcodes dar und interpretieren Binärcodes als Zahlen und Zeichen (IF4, D).

Unterrichtssequenzen

1. Klärung des Begriffs Informatik im Hinblick auf Automatisierung, Miniaturisierung und allgegenwärtige Computer
2. Einordnung von informatischen Meilensteinen in den geschichtlichen Kontext
3. Erarbeitung der Fachgebiete der Informatik anhand von exemplarischen Problemen
4. Herleitung allgemeiner Dateisystemstrukturen aus gegebenen Beispielen
5. Modellierung einer Dateisystemstruktur und Beurteilung der Struktur in Hinblick auf den Vergleich mit einem bestehenden Informatiksystem
6. Erarbeitung und Veranschaulichung einer Von-Neumann-Architektur durch den Einsatz eines Simulators und dem Entwickeln von kurzen Programmen zur Berechnung einfachster mathematischer Probleme (Addition, Multiplikation)
7. Verwendung des Binärsystems zur Darstellung von (ganzen) Zahlen / Zeichen und Erarbeitung der schriftlichen Addition im Binärsystem

Unterrichtsvorhaben EF – 2

Thema: Grundlagen der objektorientierten Analyse und Modellierung anhand von Beispielkontexten

»Ausgangspunkt im Informatikunterricht ist häufig ein Problem mit lebenswelt-

lichem Bezug. Schülerinnen und Schüler 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 Schülerinnen und Schüler nicht nur einen hohen Motivationswert, sondern ermöglicht ihnen auch die eigenständige Überprüfung der Angemessenheit und Wirkung des Modells im Rückbezug auf die Problemstellung« (KMNW_KLP_IF_2013).

»In der Einführungsphase werden ausgehend von einfachen Fragestellungen und unter Anleitung der Lehrperson zunächst einzelne Stufen eines Problemlösungsprozesses durchlaufen« (KMNW_KLP_IF_2013).

Um diesen im Kernlehrplan für die Sekundarstufe II formulierten Anforderungen gerecht zu werden, wurden Probleme »mit lebensweltlichem Bezug« so vorbereitet, dass sie im Informatikunterricht der Einführungsphase eingesetzt werden können. Die Problemstellungen stehen unter einer CC-Lizenz und können direkt im Informatikunterricht eingesetzt werden (Pieper2014).

Ein zentraler Bestandteil des Informatikunterrichts der Einführungsphase ist die Objektorientierte Modellierung. Dieses Unterrichtsvorhaben führt in die Grundlagen der Analyse, Modellierung und Implementierung ein. Die Komplexität der Anforderungen macht eine Sequenzierung erforderlich, die berücksichtigt, dass grafische Beschreibungselemente benutzt werden, die nicht selbsterklärend sind.

Grafische Beschreibungselemente:

- Railroad-Diagramme (syntaktische Strukturen)
- Objektkarten
- Objektdiagramme
- Sequenzdiagramme
- Struktogramme (hier nur Sequenz)

Konkrete Gegenstandsbereiche aus der Lebenswelt der Schülerinnen und Schüler werden analysiert und im Sinne des Objektorientierten Paradigmas strukturiert. Grundlegende Begriffe der Objektorientierung und Modellierungswerkzeuge wie Objektkarten, Objektdiagramme, Sequenzdiagramme werden eingeführt. Bei der Umsetzung der ersten Problemlösungen wird auf die Notwendigkeit der Verwendung von Kontrollstrukturen verzichtet. Damit besteht der Quellcode – als Umsetzung des Sequenzdiagramms in die Punktnotation – aus linearen Sequenzen. So ist auf diese Weise eine Fokussierung auf die Grundlagen der Objektorientierung möglich, ohne dass Kontrollstrukturen benötigt werden.

Kompetenzen

Die Schülerinnen und Schüler

- ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (IF1, M),
- nutzen Railroad-Diagramme, um syntaktisch korrekte Strukturen zu entwickeln und zu prüfen (IF1, I),
- erstellen syntaktisch korrekte Bezeichner für Objekte, Attribute und Methoden (IF3, I),
- stellen den Zustand eines Objekts dar – Objektkarte (IF1, D),
- modellieren Objekte mit ihren Attributen, Attributwerten, Methoden und Beziehungen (IF1, M),
- stellen die Ergebnisse der Modellierungsüberlegungen der objektorientierten Analyse grafisch dar – Objektkarten und Objektdiagramme (IF1, D),
- modellieren die Kommunikation zwischen Objekten (IF1, M),
- stellen die Ergebnisse der Modellierungsüberlegungen zum Ablauf der Kommunikation der Objekte grafisch dar – Sequenzdiagramme (IF1, D),
- setzen Sequenzdiagramme in die Punktnotation um (IF1, I),
- analysieren und erläutern eine objektorientierte Modellierung (IF1, A).

Unterrichtssequenzen

Identifikation von Objekten

1. An lebensweltbezogenen Beispielen werden Kandidaten für Objekte, Attribute (mit ihren Attributwerten) und Methoden im Sinne der objektorientierten Modellierung identifiziert. Dieses kann z. B. mit der Methode von Abbott geschehen.
2. Die korrekte Syntax von Bezeichnern wird mit Railroad-Diagrammen eingeführt.
3. Objekte, Attribute und Methoden werden mit syntaktisch korrekten Bezeichnern benannt.
4. Objekte werden mit Objektkarten dargestellt, mit Attributen, ihren Attributwerten und ihren »Fähigkeiten«, d. h. Methoden versehen.
5. Vertiefung: Modellierung weiterer Beispiele ähnlichen Musters.

Interaktionen zwischen Objekten

1. Der Ablauf des Szenarios wird innerhalb eines Objektspiels durchgearbeitet. Die Aktionen zwischen den Objekten in einem Protokoll festgehalten.
2. Die Interaktionen aus dem Protokoll werden in ein Sequenzdiagramm überführt.
3. Die bei den Methoden nötigen Parameter werden identifiziert.
4. Die Objektkarten werden angepasst, indem die für die Interaktionen nötigen Methoden verbessert oder ergänzt werden.

Beziehungen zwischen Objekten

1. Das Objektspiel wird wiederholt, wobei die nötigen Beziehungen zwischen den Objekten in den Vordergrund gestellt werden.
2. Die Beziehungen erhalten Bezeichner und die Objektkarten werden durch die Beziehungen zu einem Objektdiagramm ergänzt.

Abläufe von Methoden

1. Ein Ausschnitt des Sequenzdiagramms wird herausgenommen, um die Aktionen innerhalb einer Methode zu untersuchen.
2. Die Aufrufe innerhalb der Methode werden in die Punktnotation überführt und durch ein Struktogramm dokumentiert.

Unterrichtsvorhaben EF – 3

Thema: Datenschutz aus informatischer Perspektive

Vorbemerkungen – Begriffe – rechtlicher Rahmen

Der Begriff **Datenschutz** ist eigentlich eine falsche Bezeichnung, denn nicht die Daten, sondern die Person(en) müssen geschützt werden.

Die **Charta der Grundrechte der Europäischen Union** enthält einen eigenen Artikel zum Datenschutz:

1. Jede Person hat das Recht auf Schutz der sie betreffenden personenbezogenen Daten.
2. Diese Daten dürfen nur nach Treu und Glauben für festgelegte Zwecke und mit Einwilligung der betroffenen Person oder auf einer sonstigen gesetzlich geregelten legitimen Grundlage verarbeitet werden. Jede Person hat das Recht, Auskunft über die sie betreffenden erhobenen Daten zu erhalten und die Berichtigung der Daten zu erwirken.
3. Die Einhaltung dieser Vorschriften wird von einer unabhängigen Stelle überwacht.

.....
Im **Grundgesetz der Bundesrepublik Deutschland** wird in Artikel 1 Absatz 1 festgestellt: »Die Würde des Menschen ist unantastbar. Sie zu achten und zu schützen ist Verpflichtung aller staatlichen Gewalt«.

Artikel 2 Absatz 1 lautet: »Jeder hat das Recht auf die freie Entfaltung seiner Persönlichkeit, soweit er nicht die Rechte anderer verletzt und nicht gegen die verfassungsmäßige Ordnung oder das Sittengesetz verstößt«. Dieser Absatz wird als **Allgemeines Persönlichkeitsrecht** bezeichnet.

.....
Im Zusammenhang mit dem sogenannten **Volkszählungsurteil** von 1983 hat das **Bundesverfassungsgericht** dem **Recht auf informationelle Selbstbestimmung** als Ausprägung des allgemeinen Persönlichkeitsrechts den Rang eines **Grundrechts** zugesprochen.
.....

Einige Landesverfassungen (wie die **Verfassung des Landes Nordrhein-Westfalen** im Jahr 1978) wurden so erweitert, dass Datenschutz Grundrechtcharakter hat. Im Artikel 4 Absatz 2 der **Landesverfassung Nordrhein-Westfalen** heißt es: »Jeder hat Anspruch auf Schutz seiner personenbezogenen Daten. Eingriffe sind nur in überwiegendem Interesse der Allgemeinheit auf Grund eines Gesetzes zulässig.«

Kompetenzen

Die Schülerinnen und Schüler

- führen Handlungsanweisungen im Kontext von Rollenbeschreibungen aus (IF2, D),
- werten aggregierte Datensammlungen interessensgeleitet und zielgerichtet aus (IF1, K),
- bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (IF5, A),
- nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und **verantwortungsbewusst** (IF4, D).

Unterrichtssequenzen

Erweiterung einer bereits vorgenommenen objektorientierten Modellierungsfragestellung. Lebensweltlich verankerte Modellierungsaufgaben oder -szenarien können immer um Aufgabenteile ergänzt werden, die eine datenschutzrelevante Öffnung ermöglichen. So wird problembezogen an Fragen des Datenschutzes gearbeitet und für konkrete Anwendungsfälle wird die Sensibilisierung für datenschutzrelevante Fragestellungen erreicht.

Eine weitere – von der vorgängigen Arbeit unabhängige – Möglichkeit besteht darin, ein Planspiel oder ein Rollenspiel durchzuführen, das Problemsituationen provoziert, die datenschutzrelevante Elemente enthält.

In diesem Vorhaben ist die informatische Perspektive einzunehmen. Dies bedeutet insbesondere, dass aktuelle Problemsituationen, die regelmäßig bekannt werden (z. B. zur Überwachung) einen Anlass darstellen, sich mit Fragen des Persönlichkeitsschutzes auseinander zu setzen. Es ist unbedingt zu empfehlen, hier die informatischen Kompetenzen der Schülerinnen und Schüler so zu stärken, dass sie auf einer fachlichen Grundlage handlungs- und gestaltungsfähig werden.

Beide Varianten verweisen zudem auf die weitere Arbeit in dem konkreten Informatikkurs:

- Die Frage der informatischen Modellierung der jeweiligen Situation – insbesondere der wissensbasierten z. B. mit Hilfe der ER-Modelle für Datenbanken
- Fragen nach Datensicherheit – Kryptologie
- Informatiksysteme als vernetzte Strukturen – Protokolle und ihre Konsequenzen für die technisch gestützte Kommunikation

Unterrichtsvorhaben EF – 4

Thema: Strukturierung und Organisation von Daten

Zum derzeitigen Lernstand der Schülerinnen und Schüler soll die Strukturierung und Organisation von Daten lediglich auf der Entwurfsebene behandelt werden. Die Implementierung der Algorithmen kann zu diesem Zeitpunkt noch nicht erarbeitet werden. Diese Ebene wird in der Qualifikationsphase angegangen. Dazu gehört auch die Entscheidung für die zu verwendende Datenstruktur Feld oder verkettete Liste.

Dennoch können bereits in der EF Vor- und Nachteile bei der Verwendung von Feldern und verketteten Listen als grundlegende Möglichkeiten zur Verwaltung von linearen Datenstrukturen erarbeitet werden (direkte Adressierung vs. Suchen und statische vs. dynamische Datenstruktur).

Sequenzierung des Unterrichtsvorhabens

Kompetenzen

Die Schülerinnen und Schüler

- analysieren Such- und Sortieralgorithmen und wenden sie auf Beispiele an (IF2, D)
- entwerfen einen weiteren Algorithmus zum Sortieren (IF2, M)
- beurteilen die Effizienz von Algorithmen am Beispiel von Sortierverfahren hinsichtlich Zeitaufwand und Speicherplatzbedarf (IF2, A)
- testen Programme schrittweise anhand von Beispielen (IF2, I)
- ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (IF1, M)

Unterrichtssequenzen

Zum derzeitigen Lernstand der Schülerinnen und Schüler kann die Beschreibung der Algorithmen zunächst nur auf textueller Ebene stattfinden, wobei ein besonderes Augenmerk auf die sprachlich korrekte und nachvollziehbare Beschreibung typischer Ablaufstrukturen, wie zum Beispiel Verzweigungen und Schleifen, gelegt werden sollte.

Strukturierung von Daten

1. Anhand lebensweltlicher Beispiele wird die Notwendigkeit, mindestens aber der Nutzen linearer Datenstrukturen verdeutlicht.
2. Aufbauend auf der grundlegenden Idee zur linearen Datenstruktur werden die Gemeinsamkeiten und Unterschiede zwischen Liste, Schlange und Keller sowie die jeweiligen Standardmethoden an passenden Beispielen erarbeitet.

Zur Erarbeitung der linearen Datenstrukturen bieten sich praktisch unzählige Beispiele aus der Lebenswelt der Schülerinnen und Schüler an: eine Senderliste eines Fernsehers/Radios (verkettete Liste), eine Warteschlange an der Supermarktkasse (Schlange), ein Ablegestapel beim Kartenspiel (Stapel), eine Wiedergabeliste eines Musikabspielgeräts (Mischung aus verketteter Liste und Schlange) usw. Bei allen Beispielen sollte darauf geachtet werden, dass die Ordnungsrelation leicht zu identifizieren ist.

Organisation von Daten

1. Erarbeitung eines Suchalgorithmus ggf. inklusive Thematisierung von Feldern und verketteten Listen.
2. Erarbeitung einiger einfacher Sortieralgorithmen, wie Sortieren durch Einfügen, Sortieren durch Auswahl usw.
3. Die Schülerinnen und Schüler erarbeiten einen weiteren, eigenen Algorithmus und diskutieren diesen.

Die Algorithmen können an den bereits gewählten Beispielen erarbeitet werden. Für ein besseres Verständnis und zur Visualisierung werden die Algorithmen von den Schülern im einfachsten Fall mit echten Objekten wie zum Beispiel Münzen, Spielkarten usw. durchgeführt, darüber hinaus aber auch in Objektspielen an umfangreicheren Beispielen. Die Objektspiele bieten darüber hinaus auch an, um die Strukturierung der Daten in Feldern und verketteten Listen zu thematisieren. Bei der Strukturierung der Daten in Feldern übernimmt ein Schüler die Organisation der Daten, bei der verketteten Liste wird Funktionalität an die einzelnen Listenelemente abgegeben. Hinweis: Falls das Unterrichtsvorhaben EF-5 (Klassen und ihre Implementierung durch Umsetzung einer objektorientierten Modellierung) vor diesem Unterrichtsvorhaben unterrichtet wird, dann sollten Anpassungen an EF-4 vorgenommen werden, um die Vorkenntnisse der Schülerinnen und Schüler im Bereich Klassenmodellierung zu berücksichtigen.

Bei allen Algorithmen wird eine einfache Aufwandsbetrachtung durchgeführt. Hier können auch die ersten Landau-Symbole eingeführt werden (O-Notation mit: \mathcal{O}). Auf eine nähere mathematische Betrachtung wie zum Beispiel die Rechenregeln im Zusammenhang mit Landau-Symbolen sollte jedoch aufgrund der Komplexität verzichtet werden.

Thema: Klassen und ihre Implementierung durch Umsetzung einer objektorientierten Modellierung

Der Übergang von der Sicht, die sich auf die in der Problemstellung erkannten und analysierten notwendigen Objekte zur Lösung des Problems ergeben, auf die nächste Abstraktionsstufe stellt eine nicht zu unterschätzende Herausforderung dar. Daher muss dieser Übergang sorgfältig vorbereitet werden.

Im Zusammenhang mit bereits erfolgreich bearbeiteten Problemstellungen – beginnend mit der Methode von Abbott über Objektkarten und Objektdiagramme bis zum Sequenzdiagramm, wurde eine Reihe von Objekten identifiziert und zur Lösung des Problems herangezogen. Viele der Problemstellungen aus (**Pieper2014**) enthalten Fragestellungen, bei denen jeweils immer mindestens zwei Objekte sich zu einer Klasse zusammenfassen lassen.

An diesen Beispielen wird die Klassifizierung als Werkzeug zur Abstraktion eingeführt. Der Übergang von den Attributwerten zu Typen von Attributwerten, Überlegungen, welche der Objekte der jeweiligen Problemstellung sinnvoll zu einer Klasse abstrahiert werden können/sollten und der Übergang von Objektbeziehungen zu Beziehungen auf Klassenebene stellen weitere wichtige Abstraktionsschritte dar, die hier mit einer deutlichen Pointierung zunächst den Abstraktionsprozess auf der Ebene des Übergangs von Objekt- zu Klassendiagrammen betreffen.

Die Abstraktion von Objekten und ihren Beziehungen zu Klassen stellt eine Metastruktur dar, die nicht trivial ist.

Kompetenzen

Die Schülerinnen und Schüler

- ermitteln gemeinsame Attribute und Methoden verschiedener Objekte (IF1, M),
- klassifizieren Objekte nach Attributen und Methoden (IF1, M),
- identifizieren die Art von Attributwerten verschiedener Objekte und geben die Art fachgerecht an als Zeichenkette, Zahl, Wahrheitswert, Objekt, Sammlung von Objekten (IF1, M),
- modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (IF1, M),
- modellieren Klassen unter Verwendung von Vererbung (IF1, M),
- stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (IF1, D),
- dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (IF1, D),
- implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (IF1, I).

Unterrichtssequenzen

Klasse als Abstraktion

1. Die Objekte eines Szenarios werden gruppiert und ihre Methoden und Attribute verglichen.
2. Für die Klassen werden Klassenkarten erstellt, die die Methoden und Attribute aus der Vereinigung der gruppierten Objekte erhalten.
3. In einem Klassendiagramm werden die Klassenkarten eines Szenarios zusammengeführt und durch Beziehungen miteinander verknüpft.

Überführung von Klassenkarten in eine Programmiersprache

1. Aus der Darstellungsform Klassenkarte wird der Quellcode in einer Programmiersprache erstellt.
2. Die Klassendeklaration in einer Programmiersprache wird in die Darstellungsform Klassenkarte überführt.
3. Formale Korrektheit des Quellcodes zur Klassendeklaration in einer Programmiersprache wird mittels der zugrundeliegenden Syntax geprüft/entwickelt/diskutiert.

Implementierung einer Methode

1. Die einfachen Get- und Set-Methoden werden erstellt.
2. Die bereits in der Punktnotation mit einem Struktogramm aufgeschriebenen Methoden werden in die Programmiersprache übersetzt.
3. Mit Hilfe von Textausgaben lässt sich der Ablauf des Programms nachvollziehen.

Nach ersten Erfolgen mit der Ein- und Ausgabe von Daten über die Konsole kann man als Alternative/Erweiterung das MVC-Konzept einführen (model-view-controller). Auf diese Weise können die Schülerinnen und Schüler ohne die Verwendung von unverstandenen Werkzeugen in Entwicklungsumgebungen eine einfache grafische Bedienoberfläche nach dem EVA Prinzip verwenden, um Daten in ihrer Modellierung ein- und auszugeben. Der Quelltext sollte zu Beginn vorgegeben und möglichst knapp und einfach sein (insbesondere für den View).

Unterrichtsvorhaben EF – 6

Thema: Modellierung von Abläufen – Kontrollstrukturen

Um die bestehenden Modelle um vielfältige Abläufe erweitern zu können, müssen die Schülerinnen und Schüler in der Lage sein, Abläufe mit Hilfe von Kontrollstrukturen präzise zu beschreiben, zu modellieren und schließlich zu implementieren. Dabei sollte nicht unterschätzt werden, wie viel Zeit Schülerinnen und Schüler

benötigen, um ein Programm erfolgreich zu implementieren und zum Ablauf zu bringen. Da das Erfolgserlebnis eine hohe Motivation für die weitere Arbeit im Schulfach Informatik begünstigt, ist es notwendig, für diese Phase eine zeitliche Streckung zu ermöglichen – nur wenn alle Schülerinnen und Schüler diese Phase erfolgreich bewältigen, kann sie als abgeschlossen betrachtet werden.

Es ist anzustreben, dass jede Schülerin und jeder Schüler jede Zeile des Quelltextes einer entwickelten Lösung erläutern kann.

Sequenzierung des Unterrichtsvorhabens

Kompetenzen

Die Schülerinnen und Schüler

- analysieren und erläutern einfache Algorithmen und Programme (IF2, A)
- modifizieren einfache Algorithmen und Programme (IF2, I)
- entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (IF2, M)
- implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (IF2, I)
- testen Programme schrittweise anhand von Beispielen (IF2, I)
- implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (IF3, I)
- interpretieren Fehlermeldungen und korrigieren den Quellcode (IF3, I)

Unterrichtssequenzen

Die in der EF erworbenen Kenntnisse über Ablauf- und Kontrollstrukturen bilden die Grundlage für einige der Unterrichtsvorhaben in der Qualifikationsphase. Unabhängig von den gewählten Unterrichtsbeispielen erarbeiten die Schülerinnen und Schüler die nachfolgend aufgeführten Ablauf- und Kontrollstrukturen:

- einfache Anweisung und Sequenzen aus einfachen Anweisungen
- Verzweigung (inkl. Mehrfachverzweigung)
- Zyklus und/oder Rekursion

Die Schülerinnen und Schüler sollen die aufgeführten Ablauf- und Kontrollstrukturen auf verschiedenen Ebenen erfassen und beschreiben können:

1. Verbale (umgangssprachliche) Formulierung des Ablaufs
2. Halbverbale Formulierung (ggf. vereinbarter Pseudocode)
3. Struktogrammdarstellung

4. Programm Quelltext in der gewählten Programmiersprache

Da die Schülerinnen und Schüler Vorwissen zu Kontrollstrukturen mitbringen und zumindest ohne Fachsprache Abläufe umgangssprachlich verbal beschreiben können, bietet es sich an, die obige Bearbeitungsreihenfolge einzuhalten, also von der Umgangssprache zu Struktogrammen und Programm Quelltexten.

Vorschläge für Unterrichtsbeispiele befinden sich in der Materialsammlung (**Pieper2014**).

Unterrichtsvorhaben Q1 – 1

Thema: Verwaltung von Daten in linearen Datenstrukturen

Listen kommen in Realweltsituationen in allgemeiner und in spezieller Form vor. Sei es eine Warteschlange an einer Kasse, eine Polizeikontrolle oder das Stapeln von Getränkeboxen. In allgemeiner Form bei der Zusammenstellung von Zügen oder bei der Aufstellung einer Mannschaft beim Staffellauf. Eine Modellierung derartiger Abläufe stellt eine geeignete Möglichkeit dar, Listenstrukturen zu thematisieren.

Kompetenzen

Da dieses Unterrichtsvorhaben die Basis für die Arbeit mit (linearen) Datenstrukturen bildet, wird eine Vielzahl der im Kernlehrplan definierten Kompetenzen bedient. Aufgrund der sehr großen Zahl der Kompetenzen werden hier nur diejenigen aufgeführt, die bei der Verwaltung von Daten in linearen Datenstrukturen am stärksten gefördert werden, insbesondere im Vergleich zu den Unterrichtsvorhaben zur Modellierung und Implementierung in der Einführungsphase.

Die Schülerinnen und Schüler

- stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (IF1, D)
- modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (IF1, M)
- verwenden bei der Modellierung geeigneter Problemstellungen Möglichkeiten der Polymorphie (IF1, M)
- dokumentieren Klassen (IF1, D)
- implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (IF1, I)
- implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (IF2, I)
- testen Programme systematisch anhand von Beispielen (IF2, I)
- erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (IF2, A)

Unterrichtssequenzen

Es gibt zwei Möglichkeiten, lineare Datenstrukturen zu thematisieren: Entweder man beginnt mit einer der speziellen Formen der Liste, der Schlange (Queue) oder dem Stapel (Stack), oder man beginnt mit der allgemeinen Form der Liste (Einfügen und Entnehmen an einer beliebigen Stelle). Bei den erstgenannten ist

das Einfügen und Entnehmen nicht an einer beliebigen Stelle möglich, so dass hier von speziellen Formen der Liste gesprochen werden kann, die dem FIFO- bzw. LIFO-Prinzip unterliegen. Strukturen, die der Funktionsweise von Schlangen und Stapeln entsprechen, können im Alltag leicht identifiziert werden, so dass über solche lebensweltlichen Beispiele unproblematisch ein Einstieg gefunden werden kann. Außerdem ist durch das FIFO/LIFO-Prinzip der Umfang der Methoden im Vergleich zur allgemeinen Liste stark reduziert, was den Einstieg in lineare Datenstrukturen ebenfalls erleichtert.

Übersicht zum Vorgehen im Unterricht

1. Entwicklung der Standardmethoden von Schlangen/Stapeln anhand eines realweltlichen Beispiels und darauf aufbauender Modellierung des Beispiels
2. Einführung der für das Zentralabitur vorgegebenen Klassen
3. Implementation des Beispiels unter Verwendung der vorgegebenen Klassen

Nach erfolgter Modellierung und Implementierung des Beispiels einer Schlange bzw. Stapels kann dieses in Richtung allgemeiner Liste erweitert werden. Zum Beispiel könnte das Modell einer Warteschlange vor einer Kasse um Methoden erweitert werden, die das Vordrängeln einer Person (insert) oder das vorzeitige Verlassen der Schlange (remove) ermöglichen, was dem eigentlichen FIFO-Prinzip der Schlange widerspricht. Das Vorgehen zur Erarbeitung der Liste kann analog zum Vorgehen für die Schlange/den Stapel erfolgen.

Unterrichtsvorhaben Q1 – 2

Thema: Suchen und Sortieren auf linearen Datenstrukturen

Die Schülerinnen und Schüler haben bereits im Unterrichtsvorhaben Q1-1 die Modellierung und Implementierung linearer Datenstrukturen kennengelernt und angewendet. Sie kennen daher den strukturellen Aufbau verketteter Listen und ihre Standardoperationen und haben sie in einfachen Beispielen verwendet. Außerdem haben die Schülerinnen und Schüler in der EF einfache Such- und Sortieralgorithmen informell (also ohne Implementierung) an realweltlichen Objekten kennengelernt. Darauf baut dieses Unterrichtsvorhaben auf.

Sequenzierung des Unterrichtsvorhabens

Kompetenzen

Die Schülerinnen und Schüler

- analysieren und erläutern Algorithmen und Programme (IF2, A)
- modifizieren Algorithmen und Programme (IF2, I)

- stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (IF2, D)
- entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien »Modularisierung« und »Teilen und Herrschen« (IF2, M)
- implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (IF2, I)
- testen Programme systematisch anhand von Beispielen (IF2, I)
- erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (IF2, A)
- implementieren und erläutern iterative und rekursive Such- und Sortierverfahren (IF2, I)
- beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (IF2, A)
- nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (IF3, I)
- beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (IF3, A)
- interpretieren Fehlermeldungen und korrigieren den Quellcode (IF3, I)
- wenden eine didaktisch orientierte Entwicklungsumgebung zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an (IF4, I)

Unterrichtssequenzen

Zunächst werden Suchalgorithmen erarbeitet. Es bietet sich an, auf im vorangegangenen Unterrichtsvorhaben modellierte und implementierte Beispiele für lineare Datenstrukturen zurückzugreifen und diese ggf. noch mit unsortierten Daten zu füllen. Die Schülerinnen und Schüler entwickeln eigene Algorithmen zum Suchen von Inhalten in linearen Datenstrukturen und vergleichen diese mit Standard-Suchalgorithmen, wie die lineare Suche und die binäre Suche (hierfür ist die direkte Adressierung praktisch Voraussetzung, daher Formalisierung und Implementierung nur im Array).

Die folgende Übersicht stellt eine mögliche Strukturierung für die Erarbeitung von Suchalgorithmen dar:

Übersicht zum Vorgehen im Unterricht

1. Entwicklung und Durchführung von Suchalgorithmen anhand realweltlicher Objekte (z. B. Doppelkopfkarten, Supertrumpf-Karten, Münzen etc.)
2. strukturierte Darstellung der Algorithmen
3. Implementierung der Algorithmen (am besten in bestehende Projekte)
4. Testen der Implementationen

Eine einfache Aufwandsbetrachtung der Suchalgorithmen führt schnell zu der Erkenntnis, dass der Suchaufwand je nach gewähltem Algorithmus bei vorsortierten Daten geringer sein kann. Dies führt zur Notwendigkeit von Sortieralgorithmen. Auch hier entwickeln die Schülerinnen und Schüler eigene Algorithmen und vergleichen diese mit Standard-Suchalgorithmen, wie Bubble-Sort, Selection-Sort, Insertion-Sort, Quick-Sort. Die Erarbeitung der Sortieralgorithmen kann analog zur Erarbeitung der Suchalgorithmen erfolgen (siehe obige Übersicht).

Unterrichtsvorhaben Q1 – 3

Thema: Bäume

Man sagt oft, dass nichtlineares Denken bessere Leistungen hervorbringen kann. Bäume bilden eine hierarchische Struktur ab, die man im realen Leben z. B. aus Stammbäumen kennt. Strukturen von Unternehmen sind häufig hierarchisch organisiert (Aufbauorganisation). Weitere Beispiele sind das Inhaltsverzeichnis eines Buches und – aus technischer Sicht – die Struktur des Dateisystems eines Datenträgers. Hierbei handelt es sich um allgemeine Bäume. Ein Sonderfall des allgemeinen Baumes ist der Binärbaum, bei dem jeder Elternknoten höchstens zwei Kindknoten haben kann. Diese Bäume tauchen im realen Leben z. B. dort auf, wo Entscheidungen getroffen werden, die exklusiv mit »ja« oder »nein« beantwortet werden (Entscheidungsbäume). Effizientes Suchen erfolgt im allgemeinen Fall auf Grundlage einer Baumstruktur, so dass in jedem Schritt der Suche die Hälfte der Elemente ausgeschlossen werden kann. Somit ist die Anzahl der Schritte beschränkt mit: $\mathcal{O}(\log n)$. Dies trifft auch auf effiziente Sortierverfahren zu, die sich explizit oder implizit den Baum zu nutze machen und in $\mathcal{O}(n \log n)$ arbeiten.

Kompetenzen

Die Schülerinnen und Schüler

- stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (IF1, D),
- ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (IF1, M),
- entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien »Modularisierung« und »Teile und Herrsche« (IF2, M),
- implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (IF2, I),
- erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (IF2, A),

- implementieren und erläutern iterative und rekursive Such- und Sortierverfahren (IF2, I),
- beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (IF2, A),

Unterrichtssequenzen

Beginnend bei einem Stammbaum, einer Unternehmensstruktur oder des Inhaltsverzeichnisses eines Buches wird die Struktur eines allgemeinen Baumes hergeleitet. Nach einer allgemeinen Definition kann anhand eines Inhaltsverzeichnisses herausgearbeitet werden, dass ein allgemeiner Baum auch eine Ordnungsrelation aufweisen kann. Bei einem allgemeinen Baum wird in Folge über Traversierungen gesprochen. Beginnend mit der Tiefe und Höhe des Baumes. Anhand der Fragestellung, wie alle Knoten eines Baumes »besucht« werden können, wird die Preorder-Traversierung erarbeitet. Hierbei wird deutlich, dass ein rekursives Vorgehen notwendig ist. Ausgehend von der Fragestellung, wie die Speicherplatzbelegung eines Datenträgers ermittelt werden kann, kann die Postorder Traversierung eingeführt werden. Binärbäume haben in der Informatik einen besonderen Stellenwert. Die Schüler kennen aus dem Mathematikunterricht bereits Termbäume. Ausgehend von einem vollständig geklammerten Ausdruck soll ein Termbaum modelliert werden. In diesem Zusammenhang wird der Baum durch Anwendung einer binären Postordermethode ausgewertet und mittels der neu einzuführenden Inorder-Traversierung wieder als der Term ausgegeben, der als Eingabe für den Baum verwendet wurde. Wird der Binärbaum nun um eine Ordnungsrelation ergänzt, kommt man zur Anwendung eines binären Suchbaumes. Es soll schnell ein bestimmtes Objekt gefunden werden. Steht man vor einem sortierten Bücherregal so wird man ein bestimmtes Buch jedoch nicht linear suchen, sondern intuitiv in der Mitte beginnen und bei jeder Entscheidung im Laufe der Suche die Hälfte der Bücher ausschließen. Dieses Verfahren kann nun auch auf den modellierten Baum übertragen werden. Es sollte jedoch berücksichtigt werden, dass die Effizienz eines binären Suchbaums davon abhängt, wie dieser aufgebaut worden ist. Es sollte verdeutlicht werden, dass dieser zu einer linearen Liste degradieren kann, wenn die Objekte geordnet in den Baum eingefügt werden. *Im Leistungskurs wird darum auch der AVL-Baum als Antwort auf dieses Problem thematisiert. Nachdem der Suchbaum entwickelt wurde und bekannt ist, wann dieser effizient funktioniert, werden Sortierverfahren besprochen, die sich die Baumstruktur zu nutze machen. Hier sollten Heapsort, Mergesort und Quicksort besprochen werden. Zum Abschluss wird die Frage thematisiert, ob es Sortierverfahren gibt, die eine bessere Laufzeit als die zuvor besprochenen Sortierverfahren aufweisen. In diesem Zusammenhang sollte das Prinzip von Bucket-Sort erarbeitet werden.*

Thema: Graphen

Dieses Vorhaben ist für den Leistungskurs verpflichtend, es wird jedoch empfohlen, die Algorithmen von Dijkstra und Kruskal im Grundkurs ohne Implementationen zu thematisieren.

Graphen sind Bestandteil vieler Anwendungen aus der realen Welt. Zu aller erst denkt man an Navigationssysteme, bei denen das Wegenetz mittels Graphen in Informatiksystemen abgebildet wird. Aber auch Beziehungsgeflechte zwischen Personen und Optimierungsprobleme in der Betriebswirtschaftslehre lassen sich mit Graphen abbilden.

Kompetenzen

Die Schülerinnen und Schüler

- stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (IF1, D),
- stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (IF2, D),
- entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien »Modularisierung«, »Teile und Herrsche« und »Backtracking« (IF2, M),
- implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (IF2, I),
- erläutern Operationen dynamischer (linearer und nicht-linearer) Datenstrukturen (IF2, A),
- implementieren Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (IF2, I),
- beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (IF2, A).

Unterrichtssequenzen

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 betrachtet, 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 Spannbaumes und der Algorithmus von Kruskal zu thematisieren sein.

Unterrichtsvorhaben Q1 – 5

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

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.

Das Unterrichtsvorhaben kann als eigenständiges Vorhaben durchgeführt oder aber auch in anderen Vorhaben integriert werden. Im Grundkurs kann das MVC-Prinzip im Anschluss an das Unterrichtsvorhaben zu linearen Datenstrukturen durchgeführt werden, um so zu zeigen, dass Programme so entwickelt werden sollten, dass sie erweiterbar bzw. leicht veränderbar sind. ***Im Leistungskurs bietet sich hier beispielsweise das Unterrichtsvorhaben zum Thema Netzwerk an. Da hier Daten von außen empfangen werden, müssen diese dem Benutzer sichtbar gemacht werden.*** 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.

Spezielle Anmerkungen, die weder durch die Kompetenzen noch durch die Sequenz deutlich werden – z. B. was hier nicht thematisiert werden sollte etal. ggf Voraussetzungen, die explizit angegeben werden müssen, Hinweise auf Alternativen.

Kompetenzen

Die Schülerinnen und Schüler

- analysieren und erläutern objektorientierte Modellierungen (IF1, A)
- implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (IF1, M)
- analysieren und erläutern Algorithmen und Programme (IF2, A)
- modifizieren Algorithmen und Programme (IF2, I)

Unterrichtssequenzen

Dass es für unterschiedliche Situationen sinnvoll ist, verschiedene Arten von Ausgaben (View) anzeigen zu können, ist den Schülerinnen und Schülern aus der Nutzung von Informatiksystemen bekannt. Bei den im Zusammenhang mit den bisher erstellten Programmen wurde das Hauptaugenmerk auf die fachliche Modellierung (Model) gelegt. Der interaktiven Steuerung (Control) wurde wenig Aufmerksamkeit geschenkt. Daher bietet sich ein Einstieg über die Analyse bereits erstellter Programme an, die nicht konsequent nach dem MVC-Prinzip erstellt wurden. Eine leitende Fragestellung ist »Wie kann eine anders gestaltete Oberfläche für dieses Programm realisiert werden?«. Als Beispiel für eine alternative Ausgabe kann die Konsole verwendet werden. Hierbei wird die Notwendigkeit der Einteilung der Klassen in Gruppen ersichtlich, da die Controller-Klasse die Ausgabe organisiert und die Model-Klassen sowohl bei einem Konsolen-Programm, als auch bei einem Programm mit GUI gleich bleiben. Der umgekehrte Weg ist natürlich auch möglich. Ein Programm, das auf einer Konsolenausgabe basiert, so zu modifizieren, dass ohne großen Aufwand eine GUI hinzugefügt wird.

Vor allem methodisch bieten sich hier viele Möglichkeiten. Bei einer konsequenten Entwicklung gemäß des MVC-Prinzips kann in den Gruppen je nach Leistungsniveau arbeitsteilig gearbeitet werden. Die Erstellung der Model-Klassen dürfte hierbei eher einfach sein. Bei obigem Ansatz, bei dem ein fertiges Programm modifiziert wird, können verschiedene Oberflächen parallel entwickelt werden. Die Oberflächen sollten dann einfach austauschbar sein. So lernen die Schülerinnen und Schüler, wie wichtig es ist, im Vorfeld Schnittstellen klar zu definieren.

Unterrichtsvorhaben Q1 – 6

Thema: Datenbanken

Datenbanken sind Grundlage vieler Informatiksysteme, mit denen Schülerinnen und Schüler alltäglich in Kontakt kommen. Meist merken sie nichts davon. Soziale Netzwerke, Fahrplanauskunftssysteme, Online-Shops, die Verarbeitung der Patientendaten beim Arzt usw. basieren auf Datenbanksystemen. Aus diesem Grund erscheint es sinnvoll, anhand eines Realweltproblems in die Thematik der

Datenbanken einzusteigen und dieses System z. B. einen Online-Shop oder eine Supermarktkasse zu analysieren, um so den Aufbau eines Datenbanksystems kennen zu lernen. In Folge wird dieses Datenbanksystem modelliert und implementiert.

Kompetenzen

Die Schülerinnen und Schüler

- ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten (IF1, M),
- stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten mit Kardinalitäten in einem Entity-Relationship-Diagramm grafisch dar (IF1, D),
- modifizieren eine Datenbankmodellierung (IF1, M),
- modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema (IF1, M),
- bestimmen Primär- und Sekundär- und Fremdschlüssel (IF1, M),
- analysieren und erläutern eine Datenbankmodellierung (IF1, A),
- stellen grafisch den Ablauf einer Anfrage an ein Datenbanksystem dar (Client-Server-Modell) (IF4, M),
- erläutern die Eigenschaften normalisierter Datenbankschemata (IF1, A),
- überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (IF1, D),
- überführen Datenbankschemata in die erste bis dritte Normalform (IF1, M).

Unterrichtssequenzen

Gehen wir davon aus, dass die Schülerinnen und Schüler bereits objektorientiert modellieren können, kann eine unterrichtliche Sequenzierung umgesetzt werden, bei der das objektorientierte Modell über das MVC-Konzept zur ER-Modellierung verwendet werden.

Folgender Ablauf erweist sich für den Datenbankentwurf als sinnvoll:

- Datenanalyse für ein bestehendes Realweltproblem.
- Erstellung eines ER-Modells.
- Überführung des ER-Modells in ein relationales Modell.
- Einführung der Normalformen und Überprüfung des relationalen Modells auf die ersten drei Normalformen.

Bei der Datenanalyse für ein bestehendes Realweltproblem, wie etwa ein Online-Shop oder eine Kasse im Supermarkt sollen Vorteile von Datenbanksystemen

gegenüber dateibasierter Datenspeicherung von den Schülerinnen und Schülern geäußert werden. Auch sollte die Client-Server-Struktur während des Anfrageprozesses erläutert werden.

Während der Modellierungsphase erlernen die Schülerinnen und Schüler das Anwenden der Fachbegriffe wie etwa Entität, Entitätstyp, Attribute, Beziehung, Kardinalität, Primärschlüssel, Sekundärschlüssel, Fremdschlüssel.

Bei der Überführung des ER-Modells in ein relationales Modell argumentieren die Schülerinnen und Schüler, welche beteiligten Attribute bei der Beziehungsrelation Primärschlüssel sein können und formulieren für die Beziehungstypen jeweils Abbildungsregeln. Sinnvoll ist das exemplarische Einfügen von Beispieldatensätzen in eine Relationentabelle und ebenso die schematische Darstellung einer Relation. Durch ein vorgegebenes Beispiel können Schwächen (Datenredundanz, Dateninkonsistenz) in einer Datenbankmodellierung den Schülerinnen und Schülern verdeutlicht werden. Durch die Analyse der Modellierungsschwächen können grundsätzliche Regeln erarbeitet werden, die später bei den drei Normalformen eine wesentliche Rolle spielen:

- Attribute können nicht sinnvoll weiter unterteilt werden (Vorname und Name sind z. B. bereits unterteilt)
- Attribute sind nicht nur von Teilen des Primärschlüssels (funktional) abhängig.
- Ein Attribut ist nicht von einem anderen Nichtprimärschlüssel-Attribut funktional abhängig.

Neben der Konstruktion der Datenbank durch die wissensbasierte Modellierung ist es wichtig, Anfragen an die Datenbank stellen zu können. Bei relationalen Datenbanken wird nach den Vorgaben für das Zentralabitur in NRW SQL für Anfragen verwendet. Dabei sind folgende Aspekte zu berücksichtigen:

- Datendefinition (Datenbank und Tabellen Anlegen sowie Einfügen der Relationen)
- Datenmanipulation (Insert, Update, Delete)
- Anfragen an die Wissensbasis
 - Abfragen auf einer Tabelle
 - Abfragen über mehrere Tabellen
 - Unterabfragen/Verschachtelungen, Sortieren, Gruppieren, Aggregationsfunktionen
 - Relationalalgebraische Betrachtung der durchgeführten Abfragen

Unterrichtsvorhaben Q2 – 1

Thema: Endliche Automaten und formale Sprachen

Endlichen Automaten und formalen Sprachen gehört zum Teilbereich der theoretischen Informatik. Sie bildet einen Grundbaustein für die Berechenbarkeits- und Komplexitätstheorie. Dabei wird untersucht, welche Probleme mit Automaten gelöst werden können.

Die Transduktoren, als Teilgruppe der endlichen Automaten, haben noch einen Bezug zur Lebenswelt den Schülerinnen und Schülern, da mit ihnen zum Beispiel Ticket- oder Getränkeautomaten modelliert werden können. Eine besondere Rolle kommt aber den Akzeptoren zu. Die von ihnen akzeptierten Wörter bilden eine reguläre Sprache, die sich auch über Grammatiken oder reguläre Ausdrücke darstellen lässt. Anwendung finden diese Bereiche unter anderem beim Parsern, die zum Untersuchen von Texten eingesetzt werden.

Kompetenzen

Die Schülerinnen und Schüler

- analysieren und erläutern die Eigenschaften endlicher Automaten einschließlich ihres Verhaltens bei bestimmten Eingaben (IF3, A),
- ermitteln die Sprache, die ein endlicher Automat *oder Kellerautomat* akzeptiert (IF3, D),
- entwickeln und modifizieren zu einer Problemstellung endliche Automaten *oder Kellerautomaten* (IF3, M),
- stellen endliche Automaten in Tabellen oder Graphen dar und überführen sie in die jeweils andere Darstellungsform (IF3, D),
- entwickeln zur Grammatik einer regulären Sprache *oder kontextfreien Sprache* einen zugehörigen endlichen Automaten *oder Kellerautomaten* (IF3, M),
- analysieren und erläutern Grammatiken regulärer *oder kontextfreier* Sprachen (IF3, A),
- modifizieren Grammatiken regulärer *oder kontextfreier* Sprachen (IF3, M),
- ermitteln die formale Sprache, die durch eine Grammatik erzeugt wird (IF3, A),
- entwickeln zu einer regulären *oder kontextfreien Sprache* eine Grammatik, die die Sprache erzeugt (IF3, M),
- entwickeln zur akzeptierten Sprache eines Automaten eine zugehörige Grammatik (IF3, M),

- beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken (IF3, D),
- *modellieren und implementieren Scanner, Parser und Interpreter zu einer gegebenen regulären Sprache (IF3, I),*
- zeigen die Grenzen endlicher Automaten und regulärer Grammatiken im Anwendungszusammenhang auf (IF3, A).

Unterrichtssequenzen

Der erste Teil des Unterrichtsvorhaben ist die Behandlung der endlichen Automaten. Dabei werden zuerst die Transduktoren behandelt um an die Erfahrungswelt der Schülerinnen und Schüler anzuknüpfen. Bei den anschließenden Akzeptoren sollten nicht nur die Deterministischen Automaten (DEA) behandelt werden, sondern auch die nichtdeterministischen (NEA) und ihre Umwandlungen. Eine Vielzahl von Problemen lässt sich einfacher mit einem NEA von Hand modellieren, der anschließend in einen DEA umgewandelt werden kann.

Zu den endlichen Automaten gehört auch die Modellierung und Implementierung eines entsprechenden Parsers mit dazugehörigem Scanner. Mit diesen lässt sich ein endlicher Automat mithilfe eines Programms realisieren. Ein entsprechender Interpreter setzt eine passende Eingabe an das Programm in verschiedene Aktionen um.

Auf den Automaten aufbauend werden Grammatiken und die Bedingungen für reguläre Grammatiken behandelt, so dass auf dieser Grundlage der Begriff der Sprache ausgearbeitet werden kann. Darauf folgen die reguläre Ausdrücke, dessen Einsatz zum Beispiel im Bereich der Suche vertieft wird.

Durch seinen Stapelspeicher ist der Kellerautomat eine Erweiterung des endlichen Automaten. Er akzeptiert deshalb auch kontextfreie Sprachen. Deshalb müssen hier auch die Unterschiede zu den regulären Sprachen thematisiert werden.

Unterrichtsvorhaben Q2 – 2

Thema: Kommunikation in und Aufbau von Netzwerken

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 technischen Grundlagen von Netzwerken betrachtet werden. Die folgende Auflistung führt Aspekte auf, die in der Unterrichtsreihe betrachtet werden könnten. Im Grundkurs müssen aufgrund der begrenzten Lernzeit natürlich einige Aspekte wegfallen oder deutlich gekürzt werden.

- Definition eines Netzwerkes
- Netzwerktypen nach Ausdehnung (LAN, MAN, WAN, GAN)

- Peer-to-Peer vs. Client-Server-Netzwerk
- OSI-Schichtenmodell und die Funktion der Schichten
- Datenpakete und -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 z. B. in Form von Schülerreferaten erfolgen. Die folgenden Themen könnten behandelt werden, wobei auch hier im Grundkurs aufgrund der begrenzten Lernzeit einige Aspekte wegfallen oder deutlich gekürzt werden müssen. Insbesondere die aufgeführten Aspekte sind nur als Beispiele zu verstehen und müssen natürlich nur exemplarisch betrachtet werden.

- 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/IMAP4)
- Weiterentwicklung IPv6

Nach der eher theoretischen Betrachtung werden im Leistungskurs einige Netzwerkanwendungen unter Verwendung der Landesklassen implementiert. Dabei kann beispielhaft die folgende Reihenfolge von Anwendungen umgesetzt werden:

- Portscanner (natürlich nur an eigenen Systemen)
- Echo-Client / Echo-Server
- Chat-Client / Chat-Server

Es erscheint in diesem Zusammenhang sinnvoll den Echo-Client und Echo-Server gemeinsam mit den Schülern zu entwickeln und die Schüler dann in Folge die Erweiterung des Echo-Client und Echo-Server zum Chat-Client bzw. Chat-Server durchführen zu lassen. Je nach verfügbarer Zeit kann besonders leistungsfähigen Schülern die Aufgabe gestellt werden zusätzlich noch einen vereinfachten E-Mail

Client zu schreiben. Dieser kann dann auf einen Mail Server im Internet zugreifen und dort Mails abrufen. Sowohl für den Grund- als auch für den Leistungskurs erscheint es zu Abschluss sinnvoll, die gesellschaftlichen Auswirkungen des Einsatzes von Netzwerken zu betrachten. Hier sollten Beispiele aus der Lebenswelt der Schüler (z. B. Soziale Netzwerke) als auch Auswirkungen auf die Arbeitswelt thematisiert werden.

Kompetenzen

Die Schülerinnen und Schüler

- beschreiben und erläutern Netzwerk-Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken (IF4,A).
- untersuchen und bewerten anhand von Fallbeispielen Auswirkungen des Einsatzes von Informatiksystemen sowie Aspekte der Sicherheit von Informatiksystemen, des Datenschutzes und des Urheberrechts (IF5, A),
- untersuchen und bewerten Problemlagen, die sich aus dem Einsatz von Informatiksystemen ergeben, hinsichtlich rechtlicher Vorgaben, ethischer Aspekte und gesellschaftlicher Werte unter Berücksichtigung unterschiedlicher Interessenlagen (IF5, A)
- *analysieren und erläutern Algorithmen und Methoden zur Client-Server-Kommunikation (IF2 LK, A),*
- *entwickeln und implementieren Algorithmen und Methoden zur Client-Server-Kommunikation (IF2 LK, I),*
- *analysieren und erläutern Protokolle zur Kommunikation in einem Client-Server-Netzwerk (IF4 LK, A),*
- entwickeln und erweitern Protokolle zur Kommunikation in einem Client-Server-Netzwerk (IF4, M).

Unterrichtssequenzen

Bezugnehmend auf die oben ausgewiesenen Bereiche sollten grundlegende Definitionen zu Beginn gegeben werden. Danach sollte eine grundsätzliche Einführung von Referenzmodellen erfolgen. Hierbei ist nach Möglichkeit auf die zwei gebräuchlichsten (TCP/IP, OSI) zu verweisen. Anhand von Analogien aus der Alltagswelt (Telefonat zweier Regierungschefs inklusive Dolmetscher) könnte man die verschiedenen Schichten im OSI-Modell verdeutlichen.

Danach wäre eine exemplarische Bearbeitung von mindestens zwei Protokollen sinnvoll, bevor die Schülerinnen und Schüler ihr eigenes Protokoll gemäß Vorgaben (vgl. Zentrale Klausuren Informatik ab 2008, NRW) entwickeln sollen. Ggf. könnten weitere Protokolle in Referate ausgelagert werden.

Ein Großteil der oben aufgeführten Themen (Routing, Topologien, DNS) lässt sich zeitbedingt nicht im Detail besprechen und sollten ebenfalls als Referate vergeben werden. Auch im Grundkurs sollte die Implementation von mindestens einem Beispiel der Client-Server-Kommunikation (z. B. ECHO-Client) erfolgen.

Unterrichtsvorhaben Q2 – 3

Thema: Sicherheit in der Informatik: Verschlüsselung und ihre Folgen

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.

Kompetenzen

Die Schülerinnen und Schüler

- analysieren und erläutern Eigenschaften, *Funktionsweisen* und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren (IF4, A).

Unterrichtssequenzen

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. ***Im LK sollte das Verfahren auf jeden Fall auch implementiert werden.***

Ergänzend wird 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 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.

2.2 Grundsätze der fachmethodischen und fachdidaktischen Arbeit

In Absprache mit der Lehrerkonferenz sowie unter Berücksichtigung des Schulprogramms hat die Fachkonferenz Informatik des Städtischen Gymnasiums Bergkamen fachmethodische und fachdidaktische Grundsätze beschlossen. Die Grundsätze 1 bis 13 beziehen sich auf fächerübergreifende Aspekte, die auch Gegenstand der Qualitätsanalyse sind, die Grundsätze 14 bis 20 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 Schülerinnen und Schüler.
3. Medien und Arbeitsmittel sind schülernah gewählt.
4. Die Schülerinnen und Schüler erreichen einen Lernzuwachs.
5. Der Unterricht fördert eine aktive Teilnahme der Schülerinnen und Schüler.
6. Der Unterricht fördert die Zusammenarbeit zwischen den Schülerinnen und Schülern und bietet ihnen Möglichkeiten zu eigenen Lösungen.
7. Der Unterricht berücksichtigt die individuellen Lernwege der einzelnen Schülerinnen und Schüler.
8. Die Schülerinnen und Schüler erhalten Gelegenheit zu selbstständiger Arbeit und werden dabei unterstützt.
9. Der Unterricht fördert strukturierte und funktionale Partner- und Gruppenarbeit.
10. Der Unterricht fördert strukturierte und funktionale Arbeit im Plenum.
11. Die Lernumgebung ist vorbereitet; der Ordnungsrahmen wird eingehalten.
12. Die Lehr- und Lernzeit wird intensiv für Unterrichtszwecke genutzt.
13. Es herrscht ein positives pädagogisches Klima im Unterricht.

Fachliche Grundsätze:

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

2.3 Grundsätze der Leistungsbewertung und Leistungsrückmeldung

Auf der Grundlage von §13 – §16 der APO-GOST sowie Kapitel 3 des Kernlehrplans Informatik für die gymnasiale Oberstufe hat die Fachkonferenz des Städtischen Gymnasiums Bergkamen die folgenden Grundsätze zur Leistungsbewertung und Leistungsrückmeldung beschlossen. Diese Absprachen stellen die Minimalanforderungen an das lerngruppenübergreifende gemeinsame Handeln aller Fachgruppenmitglieder dar. Bezogen auf die einzelne Lerngruppe kommen ergänzend weitere der in den Folgeabschnitten genannten Instrumente der Leistungsüberprüfung zum Einsatz.

Gewichtung der Leistungen

Die Beurteilungsbereiche Klausuren und sonstige Mitarbeit gehen etwa zu gleichen Teilen in die Bewertung der Schülerinnen und Schüler ein. In Zweifelsfällen überwiegt die sonstige Mitarbeit, wobei begründeter pädagogischer Spielraum bei der Leistungsbewertung grundsätzlich gegeben ist.

2.3.1 Beurteilungsbereich Klausuren

Bei der Formulierung von Aufgaben werden die Operatoren, die auch für die Abiturprüfungen im Fach Informatik gelten, eingeführt und erklärt. In den Aufgabenstellung für die Klausuren ist auf sie zurückzugreifen.

Instrumente

In den verschiedenen Jahrgangsstufen werden folgende Klausuren geschrieben.

Einführungsphase:	eine Klausur pro Halbjahr über zwei Unterrichtsstunden
Qualifikationsphase 1:	zwei Klausuren pro Halbjahr über zwei Unterrichtsstunden Im zweiten Quartal der Q1 kann die erste Klausur durch die Anfertigung einer Facharbeit ersetzt werden.
Qualifikationsphase 2.1:	zwei Klausuren über drei Unterrichtsstunden
Qualifikationsphase 2.2:	eine Klausur unter Abiturbedingungen

2.3.2 Beurteilungsbereich »Sonstige Mitarbeit«

Den Schülerinnen und Schülern werden die Kriterien zum Beurteilungsbereich »sonstige Mitarbeit« zu Beginn des Schuljahres genannt.

Verbindliche Absprachen der Fachkonferenz

- Alle Schülerinnen und Schüler führen in der Einführungsphase in Partnerarbeit oder Kleingruppen ein Softwareprojekt durch. Die so erbrachte Leistung wird in die Note für die sonstige Mitarbeit einbezogen.
- Die Themen des Softwareprojekts werden in enger Absprache mit dem Fachlehrer festgelegt. Die Anforderungsniveaus der verschiedenen Projekte sollten im Sinne der Binnendifferenzierung bei der Festlegung der Themen den individuellen Fähigkeiten der Schülerinnen und Schüler angepasst werden. So können leistungsstarke Schülerinnen und Schüler z. B. eine graphische Benutzeroberfläche (GUI) in ihre Projekte integrieren.
- Das Softwareprojekt wird über einen Zeitraum von sechs Wochen selbstständig und **außerhalb** des Unterrichts von den Schülerinnen und Schülern bearbeitet.
- Das Softwareprojekt soll ausführlich dokumentiert werden. Einige mögliche Werkzeuge zur Dokumentation sind z. B. Klassendiagramme, Objektdiagramme, Sequenzdiagramme, Struktogramme usw. Darüber hinaus werden ausführliche und nachvollziehbare Kommentare zur besseren Lesbarkeit des Quelltexts gefordert.

Leistungsaspekte

Mündliche Leistungen

- Beteiligung am Unterrichtsgespräch
- Zusammenfassungen zur Vor- und Nachbereitung des Unterrichts
- Präsentation von Arbeitsergebnissen
- Referate
- Mitarbeit in Partner-/Gruppen- und Projektarbeitsphasen

Praktische Leistungen durch zielführende Arbeit mit Informatiksystemen

- Implementierung, Test und Anwendung von Informatiksystemen

Sonstige schriftliche Leistungen

- Arbeitsmappe und Arbeitstagebuch zu einem durchgeführten Unterrichtsvorhaben
- Bearbeitung von schriftlichen Aufgaben im Unterricht
- Lernerfolgsüberprüfung durch kurze schriftliche Übungen (LZKs)

In Kursen, in denen nur wenige Kursmitglieder Klausuren schreiben, können mehrfach schriftliche Übungen durchgeführt werden. Die Art der Aufgabenstellungen ist an den Klausuren angelehnt.

Schriftliche Übungen dauern ca. 20 bis 30 Minuten und umfassen den Stoff der letzten ca. 4 bis 9 Stunden.

Kriterien

Die folgenden allgemeinen Kriterien gelten sowohl für die mündlichen als auch für die schriftlichen Formen der sonstigen Mitarbeit.

Die Bewertungskriterien stützen sich auf

- die Qualität der Beiträge,
- die Quantität der Beiträge und
- die Kontinuität der Beiträge.

Besonderes Augenmerk ist dabei auf

- die sachliche Richtigkeit,
- die angemessene Verwendung der Fachsprache,
- die Darstellungskompetenz,

- die Komplexität und den Grad der Abstraktion,
- die Selbstständigkeit im Arbeitsprozess,
- die Präzision und
- die Differenziertheit der Reflexion zu legen.

Bei Gruppenarbeiten auch auf

- das Einbringen in die Arbeit der Gruppe,
- die Durchführung fachlicher Arbeitsanteile und
- die Qualität des entwickelten Produktes.

Bei Projektarbeit darüber hinaus auf

- die Dokumentation des Arbeitsprozesses,
- den Grad der Selbstständigkeit,
- die Reflexion des eigenen Handelns und
- die Aufnahme von Beratung durch die Lehrkraft.

Grundsätze der Leistungsrückmeldung und Beratung

Die Grundsätze der Leistungsbewertung werden zu Beginn eines jeden Halbjahres den Schülerinnen und Schülern transparent gemacht.

Leistungsrückmeldungen können erfolgen

- nach einer mündlichen Überprüfung,
- bei Rückgabe von schriftlichen Leistungsüberprüfungen,
- nach Abschluss eines Projektes,
- nach einem Vortrag oder einer Präsentation,
- bei auffälligen Leistungsveränderungen,
- auf Anfrage,
- als Quartalsfeedback und
- zu Eltern- oder Schülersprechtagen.

Die Leistungsrückmeldung kann

- durch ein Gespräch mit der Schülerin oder dem Schüler,
- durch einen Feedbackbogen,

- durch die schriftliche Begründung einer Note oder
- durch eine individuelle Lern-/Förderempfehlung

erfolgen.

Leistungsrückmeldungen erfolgen auch in der Einführungsphase im Rahmen der kollektiven und individuellen Beratung zur Wahl des Faches Informatik als fortgesetztes Grund- oder Leistungskursfach in der Qualifikationsphase.

3 Qualitätssicherung und Evaluation

Das schulinterne Curriculum (vgl. Abschnitt 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.

A Hinweise auf konkrete Materialien, Werkzeuge, ...

Hier werden Details für einzelne Vorhaben als Sequenzen dargestellt, die Hinweise auf konkrete Materialien, Werkzeuge, programmiersprachliche Elemente usw. enthalten.

Material zum Vorhaben Informatik – Fachgebiete, typische Problemstellungen, Geschichte

Begrifflichkeit Arbeitsblätter zur Klärung der Begrifflichkeiten finden sich in der Materialsammlung <http://ddi.uni-wuppertal.de/material/materialsammlung/oberstufe/allgemein/definition.html>.

Geschichte der Informatik Siehe <http://ddi.uni-wuppertal.de/material/materialsammlung/oberstufe/>

Exemplarische Probleme der Fachgebiete Zur Erarbeitung der verschiedenen Fachgebiete können zum Beispiel Aufgaben aus dem Fundus von Computer Science Unplugged <http://csunplugged.org/> genommen werden.

Dateistrukturen Eine Einführung in den Umgang mit Dateien und Verzeichnissen lässt sich mithilfe von Objektkarten und Heftstreifen realisieren, siehe Materialproduktion <http://ddi.uni-wuppertal.de/material/materialsammlung/oberstufe/>

Material zum Vorhaben Grundlagen der objektorientierten Analyse und Modellierung anhand von Beispielkontexten

Beispielszenarien mit ausgearbeiteten Materialien dazu lassen sich auf der Internetseite <http://ddi.uni-wuppertal.de/material/materialsammlung/oberstufe/oom.html> finden. Material zu Syntaxdiagrammen ist auf <http://ddi.uni-wuppertal.de/material/materialsammlung/oberstufe/allgemein/syntax.html> erhältlich.

Eine Anleitung für das Objektspiel ist auf <http://ddi.uni-wuppertal.de/material/materialsammlung/oberstufe/allgemein/objektspiel.html> erhältlich. Material für das Sequenzdiagramm ist auf <http://ddi.uni-wuppertal.de/material/materialsammlung/oberstufe/allgemein/sequenz.html> und für die Parameter auf <http://ddi.uni-wuppertal.de/material/materialsammlung/oberstufe/allgemein/parameter.html> erhältlich.

Material zum Vorhaben Datenschutz aus informatischer Perspektive

Wir schlagen zwei Möglichkeiten vor, das Thema zu diesem Zeitpunkt mit den Schülerinnen und Schülern zu bearbeiten.

Planspiel Datenschutz das wohl bekannteste – von Diplom-Informatikern ent-

wickelte Planspiel (**Hammer1987; PlanspielDatenschutz1991**) ist der Klassiker und erweist sich auch heute noch als geeignet, um die zentralen Fragestellungen des Datenschutzes herauszuarbeiten und Übergangsmöglichkeiten zu weiteren Fragestellungen – sowohl fachlicher als auch rechtlicher Natur einzuleiten.

Dabei ist ein wesentlicher Punkt darin zu sehen, dass die Daten »von Hand« erfasst und ausgewertet werden. Vorschläge, den Vorgang der Datenerfassung, -speicherung und -verarbeitung Informatiksystemen zu überantworten, sind für den aufschließenden Charakter an dieser Stelle abzulehnen. Allerdings eignet sich das Planspiel durchaus für folgende Modellierungsfragestellung und kann den Übergang von der objektorientierten Sicht auf die wissensbasierte Sicht gut illustrieren.

Modellierungsbeispiel Musikcenter eine bereits modellierte Problemsituation eröffnet durch geeignete Erweiterung(en) den Zugang zu datenschutzrelevanten Fragen:

- Viele Personen möchten an einem gemeinsamen Event teilnehmen und geben dazu ihre Musikvorlieben in eine Sammlung.
- Die Auswertung der Sammlung ergibt das Musikprogramm für die gemeinsame Veranstaltung.
- Ein Teilnehmer möchte nach der Veranstaltung diejenigen kennenlernen, die die Musik gewählt haben, die er auch gut findet. Er schreibt an die Veranstalter und bittet darum, die E-Mail-Adressen derjenigen zu erhalten, die seinen Musikgeschmack teilen.

Geeignete Beispielszenarien finden sich in der Sammlung (**WeberClassCoyKurz2009**).

Cybermobbing Fragen nach dem informatischen Hintergrund für Mobbinghandlungen werden in den bekannten Präventionsempfehlungen nur selten gestellt und aufgearbeitet. Dabei stellt die Informatik die Mittel bereit: will man Cybermobbing begegnen, so sollte klar sein, dass man die informatische Qualität dieser Mittel einordnen kann: (**Hilbig2012; Hilbig2014**)

Material zum Vorhaben Klassen und ihre Implementierung durch Umsetzung einer objektorientierten Modellierung

Für die digitale Erstellung von Klassendiagrammen eignet sich das Programm Dia <http://dia-installer.de/index.html.de>.

Die Übersetzung von Klassenkarten in Programmcode lassen sich auch automatisch durch Programme wie z. B. Dia durchführen.

Material zum Vorhaben Verwaltung von Daten in linearen Datenstrukturen

Einstiegsbeispiele zu den linearen Datenstrukturen sind in unter <http://ddi.uni-wuppertal.de/material/materialsammlung/oberstufe/datenstrukturen.html> dokumentiert. Alternativ finden sich Aufgaben in den Informatikbüchern der Schulbuchverlage.

Material zum Vorhaben Bäume

Einstieg und Übungsmaterial zu Bäumen ist unter <http://ddi.uni-wuppertal.de/material/materialsammlung/oberstufe/datenstrukturen/baeume.html> zu finden. Dort gibt es auch ein Leitprogramm, mit dem die Lernenden Teile des Themas selbst erarbeiten können.

Material zum Vorhaben Graphen

Einstieg und Übungsmaterial zu Bäumen ist unter <http://ddi.uni-wuppertal.de/material/materialsammlung/oberstufe/datenstrukturen/baeume.html> zu finden. Dort gibt es auch ein Leitprogramm, mit dem sich die Lernenden das Thema selbst erarbeiten können.

Material zum Vorhaben Vielfalt beim Einsatz von Klassen durch Abstrakte Klassen, Polymorphie und MVC

Zur Unterstützung bei der Erstellung von Programmen gibt es Werkzeuge, die verwendet werden können. So gibt es den Ansatz, dass Benutzungsoberflächen mithilfe des Fenstergenerators von Stifte und Mäuse in BlueJ erzeugt werden. Auch der JavaEditor von Gerhard Röhner <http://www.javaeditor.org/> ermöglicht die Erstellung von Programmen mit grafischen Benutzungsoberflächen.

Material zum Vorhaben Datenbanken

Als eine Möglichkeit der Modellierung bietet sich die »Migration« eines objektorientierten Modells in ein ER-Modell an.

Diese Variante wurde prototypisch für das Beispiel Musikcenter <http://ddi.uni-wuppertal.de/material/materialsammlung/oberstufe/oom/musikcenter.html> umgesetzt und ist in der Staatsarbeit von Susanne Löffler (vgl. (Loeffler2010)) dokumentiert <http://www.ham.nw.schule.de/pub/bscw.cgi/4231726>

Auch das Leitprogramm Datenbanken <http://ddi.uni-wuppertal.de/material/materialsammlung/oberstufe/datenbanken.html> verfolgt diesen Ansatz.

Zur Umsetzung der Modellierungsergebnisse sollte auf Open-Source-Lösungen wie MYSQL zurückgegriffen werden.

Material zum Vorhaben Endliche Automaten und formale Sprachen

Das Leitprogramm »Automaten, Sprachen, Grammatiken und Reguläre Ausdrücke« <http://ddi.uni-wuppertal.de/material/materialsammlung/oberstufe/automaten.html> behandelt alle Bereiche aus dem Unterrichtsvorhaben in der angegebenen Reihenfolge. Dazu gehören auch die Kellerautomaten und die Turingmaschine, sowie die Umsetzung eines Parsers auf Grundlage eines endlichen Automaten.

Material zum Vorhaben Kommunikation in und Aufbau von Netzwerken

Zur Verdeutlichung von Netzwerkverkehr und den Zusammenspiel verschiedener Dienste kann die Lernsoftware FILIUS <http://www.lernsoftware-filius.de> genutzt werden. Arbeitsblätter zu verschiedenen Netzwerkprotokollen sind auch in der Materialsammlung <http://ddi.uni-wuppertal.de/material/materialsammlung/oberstufe/netzwerk.html> vorhanden.

Material zum Vorhaben Sicherheit in der Informatik: Verschlüsselung und ihre Folgen

SpionCamp (vgl. (MuellerSpionCamp2012), (AustMuellerGabriel2011)), (Puhlmann1998) (Grischott2008)

B Mitarbeitende

- Michael Albrecht, Gymnasium Essen-Werden, Essen
- Thomas Arbeiter, Europaschule Ostendorf-Gymnasium, Lippstadt
- David Beisel, Willy-Brandt-Gesamtschule, Bergkamen
- Thomas Brinkmann, Städtisches Gymnasium, Bergkamen
- Ralph Carrie, Geschwister-Scholl-Gymnasium, Münster
- Martin Emonts-Gast, Willy-Brandt-Gesamtschule, Bergkamen
- Andreas Grote, Einstein-Gymnasium, Rheda-Wiedenbrück
- André Hilbig, Carl-Humann-Gymnasium, Essen-Steele
- Prof. Dr. L. Humbert, Bergische Universität, Wuppertal
- Holger Kalinowski, Helene-Lange-Gymnasium, Dortmund
- Oliver Poth, Ernst-Barlach-Gymnasium, Unna
- Johannes Pieper, Joseph-König-Gymnasium, Haltern am See
- Alfred Schenk, Friedrich-Spee-Gymnasium, Geldern
- Tim Schmitz-Porten, Ernst-Barlach-Gymnasium, Unna
- Christian Seifert, Heinrich-Böll Gesamtschule, Bochum
- Daniel Spittank, Westfälische-Wilhelms-Universität, Münster
- André Wrede, Mariengymnasium, Arnsberg

Die notwendigen Anpassungen des gemeinsam entwickelten schulinternen Lehrplans für das Fach Informatik in der gymnasialen Oberstufe für den Einsatz am städtischen Gymnasium Bergkamen wurden in enger Absprache mit dem hausinternen Fachbereich Informatik von Thomas Brinkmann vorgenommen.