

Schulinterner Lehrplan für die gymnasiale Oberstufe

Informatik

Einführungsphase + Qualifikationsphase

1. Fachgruppe

Die Fachgruppe Informatik des Dietrich-Bonhoeffer-Gymnasiums Neunkirchen setzt sich zurzeit aus drei Lehrkräften zusammen, wobei zwei in der Sekundarstufe 1 und 2 unterrichten und eine in der Sekundarstufe 1.

Die Schule verfügt über zwei Computerräume mit Beamer mit jeweils ca. 15 Computerarbeitsplätzen. Zudem stehen Laptops und Tablets zur Verfügung.

Für den Differenzierungskurs Informatik stehen für den Anfangsunterricht einige alte Computer zur Erkundung zur Verfügung. Zudem besitzt die Schule mehrere Lego Mindstorm Roboter, die bisher jedoch ausschließlich von der Lego-Technik AG genutzt werden.

In der Sekundarstufe 1 bietet die Fachgruppe in der 8. und 9. Klasse einen Differenzierungskurs Informatik an, der zweistündig unterrichtet wird. Auf Grund des Doppelstundenprinzips an unserer Schule bedeutet dies, dass der Kurs einmal in der Woche eine Doppelstunde lang stattfindet. Diesen Kurs wählen in der Regel 40 bis 50 Schülerinnen und Schüler einer Stufe, so dass meist zwei Kurse pro Stufe eingerichtet werden. In diesem Kurs werden die theoretischen Grundlagen von Computern und Computernetzwerken vermittelt. Es erfolgt eine Einführung in die Office-Programme. Hierfür nutzt die Fachgruppe die LibreOffice-Programme, um allen Schülerinnen und Schülern zu ermöglichen auch zu Hause die Programme zu installieren und damit zu arbeiten. Zudem erlangen die Schülerinnen und Schüler erste Programmierkenntnisse.

In der Sekundarstufe 2 gibt es in der Einführungsphase in der Regel ein oder zwei Kurse, in der Qualifikationsphase gibt es meist einen Grundkurs. Die Kurse werden dreistündig unterrichtet. Auf Grund unseres Doppelstundenprinzips bedeutet dies, dass der Kurs in einer Woche 1 Doppelstunde und in der nachfolgenden Woche 2 Doppelstunden umfasst. Leistungskurse sind bisher noch nicht zu Stande gekommen.

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

Der Unterricht der Sekundarstufe II wird mit Hilfe der Programmiersprache Java durchgeführt. In der Einführungsphase kommt dabei zusätzlich die didaktische Umgebung Greenfoot zum Einsatz.

2. Entscheidungen zum Unterricht

2.1 Unterrichtsvorhaben

2.1.1 Übersichtsraster Unterrichtsvorhaben

EF-I: Theoretische und Praktische Grundlagen und Geschichte der Datenverarbeitung

EF-II: Einführung in die Objektorientierung und in die Modellierung mit Greenfoot

EF-III: Einführung in die Kontrollstrukturen

EF-IV: Methoden und Variablen

EF-V: Klassenentwurf

EF-VI: Projekt zur Wiederholung und Vertiefung der objektorientierten Grundlagen

EF-VII: Such- und Sortieralgorithmen anhand kontextbezogener Beispiele

EF-VIII: Informatik, Mensch und Gesellschaft

Q1-I: Wiederholung der objektorientierten Modellierung und Programmierung anhand einer kontextbezogenen Problemstellung

Q1-II: Modellierung und Implementierung von Anwendungen mit dynamischen, linearen Datenstrukturen

Q1-III: Suchen und Sortieren auf linearen Datenstrukturen

Q1-IV: Modellierung und Nutzung von relationalen Datenbanken in Anwendungskontexten

Q1-V: Sicherheit und Datenschutz in Netzstrukturen

Q2-I: Modellierung und Implementierung von Anwendungen mit dynamischen, nichtlinearen Datenstrukturen

Q2-II: Endliche Automaten und formale Sprachen

Q2-III: Prinzipielle Arbeitsweise eines Computers und Grenzen der Automatisierbarkeit

2.1.2 Konkretisierte Unterrichtsvorhaben

Unterrichtsvorhaben EF-I

Thema:

Theoretische und Praktische Grundlagen und Geschichte der Datenverarbeitung

Stichworte:

Binärsystem, Geschichte des Computers, Von-Neumann-Architektur, Einzelrechner, Schulnetzwerk, Dateisystem, Rechteverwaltung, Internet

Inhaltliche Kompetenzen:

Die Schülerinnen und Schüler

- stellen ganze Zahlen und Zeichen in Binärcodes dar (D).
- interpretieren Binärcodes als Zahlen und Zeichen (D).
- erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (A).
- beschreiben und erläutern den strukturellen Aufbau und die Arbeitsweise singularer Rechner am Beispiel der „Von-Neumann-Architektur“ (A).
- nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung und gemeinsamen Verwendung von Daten unter Berücksichtigung der Rechteverwaltung (K).
- nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D).
- nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K).

Inhaltsfelder:

- Informatiksysteme
- Informatik, Mensch und Gesellschaft

Inhaltliche Schwerpunkte:

- Digitalisierung
- Einzelrechner
- Dateisystem
- Internet
- Einsatz von Informatiksystemen
- Geschichte der automatischen Datenverarbeitung

Zeitbedarf:

6 Doppelstunden

Unterrichtsvorhaben EF-II

Thema:

Einführung in die Objektorientierung und in die Modellierung mit Greenfoot

Stichworte:

Objektorientierte Modellierung, Klasse, Objekt, Attribut, Operation/Methode, UML, Vererbung, Greenfoot-Roboter, Objektzustand, Bewegungen und Aktionen des Greenfoot-Roboters

Inhaltliche Kompetenzen:

Die Schülerinnen und Schüler

- ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (nur Vererbung) (M).
- modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (nur Vererbung) (M).
- modellieren Klassen unter Verwendung von Vererbung (M).
- stellen Klassen und Vererbungsbeziehungen in Diagrammen grafisch dar (D).
- stellen den Zustand eines Objektes (Greenfoot-Roboters) dar (D).
- analysieren und erläutern einfache Algorithmen und Programme (Bewegungen und Aktionen vom Greenfoot-Roboter ohne Kontrollstrukturen) (A).
- modifizieren einfache Algorithmen und Programme (des Greenfoot-Roboters ohne Kontrollstrukturen) (I).
- entwerfen einfache Algorithmen (ohne Kontrollstrukturen) des Greenfoot-Roboters und testen diese (M).

Inhaltsfelder:

- Daten und ihre Strukturierung
- Algorithmen

Inhaltliche Schwerpunkte:

- Objekte und Klassen
- Analyse, Entwurf und Implementierung einfacher Algorithmen

Zeitbedarf:

5 Doppelstunden

Unterrichtsvorhaben EF-III

Thema:

Einführung in die Kontrollstrukturen

Stichworte:

Bedingte Anweisungen, Pseudocode, Flussdiagramm, Schachtelung, Logische Verknüpfungen, vor- und nachprüfende Wiederholung, Zählschleifen, Algorithmen entwickeln und programmieren

Inhaltliche Kompetenzen:

Die Schülerinnen und Schüler

- analysieren und erläutern einfache Algorithmen und Programme (A).
- modifizieren einfache Algorithmen und Programme (I).
- entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch mit Flussdiagrammen dar (M).
- implementieren Algorithmen unter Verwendung von Kontrollstrukturen sowie Methodenaufrufen (I).
- benutzen in Bedingungen logische Verknüpfungen (I).
- testen Programme schrittweise anhand von Beispielen (I).
- implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I).
- interpretieren Fehlermeldungen und korrigieren den Quellcode (I).

Inhaltsfelder:

- Algorithmen
- Formale Sprachen und Automaten

Inhaltliche Schwerpunkte:

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

Zeitbedarf:

5 Doppelstunden

Unterrichtsvorhaben EF-IV

Thema:

Methoden und Variablen

Stichworte:

Methoden, Methodenkopf und –rumpf, Methodenaufruf, Variablen, Attribute, Datentypen (String, double, int, char, boolean), Deklaration und Initialisierung von Attributen, Rückgaben, Parameter, Konstruktor

Inhaltliche Kompetenzen:

Die Schülerinnen und Schüler

- ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M).
- dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D).
- implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I).

Inhaltsfelder:

- Daten und ihre Strukturierung

Inhaltliche Schwerpunkte:

- Objekte und Klassen

Zeitbedarf:

4 Doppelstunden

Unterrichtsvorhaben EF-V

Thema:

Klassentwurf

Stichworte:

Klassenbeziehungen, Assoziation / Kennt-Beziehung, Klassentwurf und – implementation, Sichtbarkeit (public, private, protectec), get-/set-Methoden

Inhaltliche Kompetenzen:

Die Schülerinnen und Schüler

- ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M).
- stellen die Kommunikation zwischen Objekten grafisch dar (M).
- stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D).
- analysieren und erläutern eine objektorientierte Modellierung (A).

Inhaltsfelder:

- Daten und ihre Strukturierung

Inhaltliche Schwerpunkte:

- Objekte und Klasse

Zeitbedarf:

5 Doppelstunden

Unterrichtsvorhaben EF-VI

Thema:

Projekt zur Wiederholung und Vertiefung der objektorientierten Grundlagen

Stichworte:

Entwurf, Analyse und Implementierung mehrere Klassen und ihrer Beziehung, Speicherung von Objekten in Feldern, SuM-Programmgenerator, SuM-Komponenten, Benutzungsoberfläche

Inhaltliche Kompetenzen:

Die Schülerinnen und Schüler

- ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M).
- modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M).
- modellieren Klassen unter Verwendung von Vererbung (M).
- ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen und lineare Datensammlungen in Form von Feldern zu (M).
- dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D).
- analysieren und erläutern eine objektorientierte Modellierung (A).
- implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierte Klassenbibliotheken (z.B.: SuM-Komponenten).
- erstellen und programmieren Benutzungsoberflächen mit dem SuM-Programmgenerator (I).
- implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I).
- testen Programme schrittweise anhand von Beispielen (I).
- implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I).
- interpretieren Fehlermeldungen und korrigieren den Quellcode (I).

Inhaltsfelder:

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

Inhaltliche Schwerpunkte:

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

Zeitbedarf:

10 Doppelstunden

Unterrichtsvorhaben EF-VII

Thema:

Such- und Sortieralgorithmen anhand kontextbezogener Beispiele

Stichworte:

Bubblesort, Insertionsort, Selectionsort, Quicksort, Effizienz hinsichtlich Zeitaufwand und Speicherplatz

Inhaltliche Kompetenzen:

Die Schülerinnen und Schüler

- analysieren Such- und Sortieralgorithmen und wenden sie auf Beispiele an (D).
- entwerfen einen weiteren Algorithmus zum Sortieren (M).
- beurteilen die Effizienz von Algorithmen am Beispiel von Sortierverfahren hinsichtlich Zeitaufwand und Speicherplatzbedarf (A).

Inhaltsfelder:

- Algorithmen

Inhaltliche Schwerpunkte:

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

Zeitbedarf:

4 Doppelstunden

Unterrichtsvorhaben EF-VIII

Thema:

Informatik, Mensch und Gesellschaft

Stichworte:

bei aktuellen Anlässen, gläserner Kunde, Künstliche Intelligenz, Roboter

Inhaltliche Kompetenzen:

Die Schülerinnen und Schüler

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

Inhaltsfelder:

- Informatik, Mensch und Gesellschaft

Inhaltliche Schwerpunkte:

- Wirkungen der Automatisierung

Zeitbedarf:

Parallel, bei aktuellen Anlässen

Unterrichtsvorhaben Q1-I

Thema:

Wiederholung der objektorientierten Modellierung und Programmierung anhand einer kontextbezogenen Problemstellung

Stichworte:

Vererbung, abstrakte Klassen, Polymorphie, Schnittstellen

Mögliche Projekte:

Inhaltliche Kompetenzen:

Die Schülerinnen und Schüler

- 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 Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an (I),
- interpretieren Fehlermeldungen und korrigieren den Quellcode (I),
- stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D),
- dokumentieren Klassen (D),
- stellen die Kommunikation zwischen Objekten grafisch dar (D).
- verwenden bei der Modellierung geeigneter Problemstellungen die Möglichkeiten der Polymorphie (M),

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:

4 Doppelstunden

Unterrichtsvorhaben Q1-II

Thema:

Modellierung und Implementierung von Anwendungen mit dynamischen, linearen Datenstrukturen

Stichworte:

Array (auch 2-dim.), Schlange, Klasse Queue, Stapel, Klasse Stack, Liste, Klasse List

Mögliche Projekte:

Für Schlange: Patientenschlange, Abituraufgabe 2007 – Statisten

Für Stapel: Stapelbauer, Biber-Teller-Aufgabe (schöningh 2 S. 73 Nr. 1), Palindrom-Aufgabe (schöningh 2 S. 73 Nr. 2)

Für Liste: Abfahrtslauf, Vokabeltrainer

Inhaltliche Kompetenzen:

Die Schülerinnen und Schüler

- erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A),
- analysieren und erläutern Algorithmen und Programme (A),
- beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),
- ordnen Attributen, Parametern und 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).

Inhaltsfelder:

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

Inhaltliche Schwerpunkte:

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

Zeitbedarf:

15 Doppelstunden

Unterrichtsvorhaben Q1-III

Thema:

Suchen und Sortieren auf linearen Datenstrukturen

Stichworte:

Bubble-Sort, Insertion-Sort, Selection-Sort, Quick-Sort

Mögliche Projekte:

Ländersuche, Bundesjugendspiele

Inhaltliche Kompetenzen:

Die Schülerinnen und Schüler

- 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).

Inhaltsfelder:

- Algorithmen
- Formale Sprachen und Automaten

Inhaltliche Schwerpunkte:

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

Zeitbedarf:

8 Doppelstunden

Unterrichtsvorhaben Q1-IV

Thema:

Modellierung und Nutzung von relationalen Datenbanken in Anwendungskontexten

Stichworte:

SQL, Entitäten, Attribute, Relationen, Kardinalitäten, Entity-Relationship-Modelle, Datenbankschemata, 1. bis 3. Normalform

Mögliche Projekte:

VideoCenter, Schulbuchausleihe, Reederei, Buchungssystem, Schulverwaltung

Inhaltliche Kompetenzen:

Die Schülerinnen und Schüler

- 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),
- 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),
- stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten in einem Entity-Relationship-Diagramm grafisch dar (D),
- überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D).

Inhaltsfelder:

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

Inhaltliche Schwerpunkte:

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

Zeitbedarf:

10 Doppelstunden

Unterrichtsvorhaben Q1-V

Thema:

Sicherheit und Datenschutz in Netzstrukturen

Stichworte:

Netzwerktopologien, TCP-IP Modell, Client-Server-Strukturen, Sicherheit und Verschlüsselung (symmetrisch und asymmetrisch), Datenschutz und Urheberrecht
optional: Programmierung einfacher Client-Server-Anwendungen

Mögliche Projekte:

Für Client-Server-Anwendungen: Daytime, Echo, Quote of the Day

Inhaltliche Kompetenzen:

Die Schülerinnen und Schüler

- 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 Informatiksystemen sowie die Einhaltung der Datenschutzbestimmungen und des Urheberrechts (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 (A),
- nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zum Erschließen, zur Aufbereitung und Präsentation fachlicher Inhalte (D).

Inhaltsfelder:

- Informatiksysteme
- Informatik, Mensch und Gesellschaft

Inhaltliche Schwerpunkte:

- Einzelrechner und Rechnernetzwerke
- Sicherheit
- Nutzung von Informatiksystemen, Wirkungen der Automatisierung

Zeitbedarf:

7 Doppelstunden

Unterrichtsvorhaben Q2-I

Thema:

Modellierung und Implementierung von Anwendungen mit dynamischen, nichtlinearen Datenstrukturen

Stichworte:

Rekursion, Bäume (Grad, Tiefe, Höhe, Blatt, Inhalt, Teilbaum, Ebene, Vollständigkeit), Binärbäume, Klasse BinaryTree, Traversierung (Pre-, Post- und Inorder), binäre Suchbäume, Klasse BinarySearchTree

Mögliche Projekte:

Für Rekursion: Fibonacci, Fakultät, Binomialkoeffizient

Für (Such-)Bäume: Stammbaum, Morsebaum, Informatikerbaum, Abituraufgabe zu

Rechenausdrücken: Ternbaum, Schönigh 2 S. 159-163

Inhaltliche Kompetenzen:

Die Schülerinnen und Schüler

- erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A),
- analysieren und erläutern Algorithmen und Programme (A),
- beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),
- ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M),
- ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M),
- modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M),
- verwenden bei der Modellierung geeigneter Problemstellungen die Möglichkeiten der Polymorphie (M),
- entwickeln iterative und rekursive Algorithmen unter Nutzung der Konstruktionsstrategien „Modularisierung“ und „Teilen und Herrschen“ (M),
- implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I),
- modifizieren Algorithmen und 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).

Inhaltsfelder:

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

Inhaltliche Schwerpunkte:

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

Zeitbedarf:

12 Doppelstunden

Unterrichtsvorhaben Q2-II

Thema:

Endliche Automaten und formale Sprachen

Stichworte:

Reguläre Grammatiken (Alphabet, Syntax, Semantik, Terminale, Nichtterminale, Startsymbol, Ableitungsregeln), linkslineare Grammatiken, Akzeptor
endliche deterministische Automaten (Zustände, Übergänge, Übergangsgraf, Übergangstabelle)

Umwandeln eines endlichen deterministischen Automaten in eine linkslineare Grammatik und umgekehrt

optional: kontextfreie Grammatiken, nichtdeterministische Automaten, Kellerautomat, Transduktor, Turingmaschine, Chomsky-Hierarchie

Mögliche Projekte:

Emailadressen, Zahlen (natürliche, ganze, Komma-, wissenschaftliche, Binär-Zahlen), Abituraufgabe 2008

Inhaltliche Kompetenzen:

Die Schülerinnen und Schüler

- 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 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),
- ermitteln die Sprache, die ein endlicher Automat akzeptiert (D),
- beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken (D).

Inhaltsfelder:

- Endliche Automaten und formale Sprachen

Inhaltliche Schwerpunkte:

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

Zeitbedarf:

10 Doppelstunden

Unterrichtsvorhaben Q2-III

Thema:

Prinzipielle Arbeitsweise eines Computers und Grenzen der Automatisierbarkeit

Stichworte:

Von-Neumann-Architektur (CPU, Rechenwerk, Steuerwerk, Register, Hauptspeicher), maschinennahe Befehle + Binärcode
Halteproblem

Mögliche Projekte:

Inhaltliche Kompetenzen:

Die Schülerinnen und Schüler

- 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).

Inhaltsfelder:

- Informatiksysteme
- Informatik, Mensch und Gesellschaft

Inhaltliche Schwerpunkte:

- Einzelrechner und Rechnernetzwerke
- Grenzen der Automatisierung

Zeitbedarf:

6 Doppelstunden

2.2 Grundsätze der fachmethodischen und fachdidaktischen Arbeit

Der Unterricht orientiert sich an den folgenden fachspezifischen Grundsätzen:

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

2.3 Grundsätze der Leistungsbewertung und Leistungsrückmeldung

Informatik als schriftliches Fach

Informatik als mündliches Fach

50 % - Sonstige Mitarbeit
50% - Klausur

100 % - Sonstige Mitarbeit

2.3.1 Beurteilungsbereich Klausuren

Pro Halbjahr werden in der E, Q1 und Q2.1 zwei Klausuren geschrieben. Die Klausuren werden in E und Q1 zweistündig geschrieben.

Klausuren:

ab 95%	15
ab 90%	14
ab 85%	13
ab 80%	12
ab 75%	11
ab 70%	10
ab 65%	9
ab 60%	8
ab 55%	7
ab 50%	6
ab 45%	5
ab 40%	4
ab 33%	3
ab 26%	2
ab 20%	1
unter 20%	0

2.3.2 Beurteilungsbereich Sonstige Mitarbeit

Die mündliche Beteiligung (auch Präsentationen, Mitarbeit in Gruppenarbeit, Hausaufgaben, ...) und die praktische Arbeit am Computer gehen in dem Verhältnis in die sonstige Mitarbeitsnote ein, in dem sie im Unterricht gefordert wurden. Werden schriftliche Lernerfolgskontrollen geschrieben, gehen diese pro Kontrolle bis zu 10 % in die sonstige Mitarbeitsnote ein, zusammen maximal jedoch zu 30 %.

3. Lehr- und Lernmittel

In der Einführungsphase arbeiten wir mit dem Buch Informatik 1 von schönigh, das den Schülerinnen und Schülern von der Schule zur Verfügung gestellt wird. In der Sekundarstufe 1 und in der Qualifikationsphase arbeiten wir ohne Buch.

4. Entscheidungen zu fach- und unterrichtsübergreifenden Fragen

Die Fachgruppe Informatik kooperiert mit der Firma buhl data. Bisher haben informatikinteressierte Schüler unserer Schule dort häufiger ihr Berufspraktikum gemacht. In Zukunft soll die Kooperation noch verstärkt werden. Zudem nutzt die Fachgruppe die Angebote des Haus der Berufsvorbereitung in Siegen, z.B. die Schülerkrypto- Veranstaltung.

5. Qualitätssicherung und Evaluation

Der Lehrplan sowie das Leistungsbewertungskonzept wird jährlich in den Fachschaftssitzungen anhand der im Unterricht gemachten Erfahrungen evaluiert und weiterentwickelt.