

Siegtal-Gymnasium Eitorf

Schulinterner Lehrplan

Informatik

Inhalt

1 Die Fachgruppe Informatik des Siegtal-Gymnasiums Eitorf	1
2 Entscheidungen zum Unterricht	2
2.1 Unterrichtsvorhaben	2
2.1.1 Übersichtsraster Unterrichtsvorhaben	2
2.1.2 Konkretisierte Unterrichtsvorhaben	7
2.2 Grundsätze der fachmethodischen und fachdidaktischen Arbeit	37
2.3 Grundsätze der Leistungsbewertung und Leistungsrückmeldung	38
2.3.1 Beurteilungsbereich Klausuren	38
2.3.2 Beurteilungsbereich Sonstige Mitarbeit	39
3 Entscheidungen zu fach- und unterrichtsübergreifenden Fragen	40
4 Qualitätssicherung und Evaluation.....	40

1 Die Fachgruppe Informatik des Siegtal-Gymnasiums Eitorf

Das Fach Informatik wird am Siegtal-Gymnasium ab der Jahrgangsstufe 8 im Wahlpflichtfach II zweistündig unterrichtet. Im Rahmen dieses Unterrichts machen die Lernenden erste Programmiererfahrungen in didaktischen Lernumgebungen wie FMSLogo und Scratch. Darüber hinaus erhalten die Schülerinnen und Schüler einen Einblick in Tabellenkalkulationen, Datenbanken und behandeln Grundlagen der technischen Informatik.

In den Kursen der Einführungsphase wird besonderer Wert darauf gelegt, dass kein Vorwissen aus der Sekundarstufe I zum erfolgreichen Bestehen des Kurses notwendig ist, damit auch Schülerinnen und Schüler, die in der Sekundarstufe I keinen Informatikunterricht besucht haben, diesen in der Oberstufe wählen können.

Bereits in der 5. Klasse wird für jeden Lernenden ein individueller Zugang zur Anmeldung an den Schulsystemen angelegt. Die Schülerinnen und Schüler kommen somit direkt zu Beginn ihrer Schulzeit am Siegtal-Gymnasium mit der IT in Kontakt und lernen den verantwortungsvollen Umgang mit sensiblen digitalen Zugängen.

Der Informatikunterricht wird am Siegtal-Gymnasium in der Sekundarstufe II als Wahlfach angeboten. In der Einführungsphase und in der Qualifikationsphase wird Informatik dabei als dreistündiger Kurs angeboten. Der Unterricht wird mit Hilfe der Programmiersprache Java durchgeführt.

Zurzeit besteht die Fachschaft Informatik am Siegtal-Gymnasium Eitorf aus vier Lehrkräften, von denen zwei in der Sekundarstufe II unterrichten. Ihnen stehen 2 Computerräume mit 16 und 30 Arbeitsplätzen zur Verfügung. Des Weiteren ist die Schule mit mehreren Sätzen mobiler Endgeräte ausgestattet. Alle Arbeitsplätze sind an das schulinterne Rechnernetz angeschlossen, so dass Schülerinnen und Schüler über einen individuellen Zugang Zugriff zum zentralen Server der Schule haben. Dieser Zugang kann an allen Arbeitsplätzen der Räume zum Zugriff auf ihre eigenen Daten, zur Recherche im Internet oder zur Bearbeitung schulischer Aufgaben verwendet werden.

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

Da derzeit am Siegtal-Gymnasium kein Informatik Leistungskurs angeboten wird, bezieht sich der folgende Entwurf ausschließlich auf die Inhalte des Grundkurses.

2 Entscheidungen zum Unterricht

2.1 Unterrichtsvorhaben

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

Die entsprechende Umsetzung erfolgt auf zwei Ebenen: der Übersichts- und der Konkretisierungsebene.

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

Im Folgenden werden die Unterrichtsvorhaben für den Grundkurs Informatik beschrieben. Zu jedem Unterrichtsvorhaben ist eine Anknüpfung an den Kernlehrplan durch die Angabe der entsprechenden, zu erwerbenden Kompetenzen gegeben. Der Prozess des Kompetenzerwerbs ist dabei als ein kontinuierlicher, kumulativer Prozess, der sich über längere Zeiträume erstreckt, anzusehen. Daher ist davon auszugehen, dass der Erwerb einer Kompetenz mit Abschluss einer Unterrichtssequenz nicht vollständig abgeschlossen ist, sondern erst im Rahmen eines Spiralcurriculums durch variierende Anwendungssituationen und Anforderungsniveaus abgeschlossen wird.

Übersichtsraster Unterrichtsvorhaben Einführungsphase	
Unterrichtsvorhaben EP-I	Unterrichtsvorhaben EP-II
<p>Thema: Grundlagen der objektorientierten Analyse und Programmierung anhand von Beispielkontexten</p> <p>Zentrale Kompetenzen: Modellieren Darstellen und Interpretieren Implementieren Kommunizieren und Kooperieren</p> <p>Inhaltsfelder: Daten und ihre Strukturierung Algorithmen Formale Sprachen und Automaten</p> <p>Inhaltliche Schwerpunkte: Objekte und Klassen Analyse, Entwurf und Implementierung einfacher Algorithmen Syntax und Semantik einer Programmiersprache</p> <p>Zeitbedarf: 20 Stunden</p>	<p>Thema: Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand von grafischen Spielen und Simulationen</p> <p>Zentrale Kompetenzen: Argumentieren Modellieren Implementieren Darstellen und Interpretieren Kommunizieren und Kooperieren</p> <p>Inhaltsfelder: Daten und ihre Strukturierung Algorithmen Formale Sprachen und Automaten</p> <p>Inhaltliche Schwerpunkte: Objekte und Klassen Analyse, Entwurf und Implementierung einfacher Algorithmen Syntax und Semantik einer Programmiersprache</p> <p>Zeitbedarf: 20 Stunden</p>
Unterrichtsvorhaben EP-III	Unterrichtsvorhaben EP-IV
<p>Thema: Grundlagen digitaler Datenverarbeitung</p> <p>Zentrale Kompetenzen: Argumentieren Darstellen und Interpretieren Kommunizieren und Kooperieren</p> <p>Inhaltsfelder: Informatiksysteme Informatik, Mensch und Gesellschaft</p> <p>Inhaltliche Schwerpunkte: Digitalisierung Einzelrechner Wirkung der Automatisierung Geschichte der automatischen Datenverarbeitung</p> <p>Zeitbedarf: 8 Stunden</p>	<p>Thema: Erstellung einer grafischen Benutzeroberfläche und Einführung in das Ereignis-Handling</p> <p>Zentrale Kompetenzen: Modellieren Implementieren Darstellen und Interpretieren Kommunizieren und Kooperieren</p> <p>Inhaltsfelder: Daten und ihre Strukturierung Formale Sprachen und Automaten Informatik, Mensch und Gesellschaft</p> <p>Inhaltliche Schwerpunkte: Objekte und Klassen Analyse, Entwurf und Implementierung einfacher Algorithmen Syntax und Semantik einer Programmiersprache Wirkung der Automatisierung</p> <p>Zeitbedarf: 12 Stunden</p>

Unterrichtsvorhaben EP-V	Unterrichtsvorhaben EP-VI
<p>Thema: Vertiefung von Algorithmen, insbesondere die Behandlung von Suchalgorithmen</p> <p>Zentrale Kompetenzen: Argumentieren Modellieren Implementieren Darstellen und Interpretieren Kommunizieren und Kooperieren</p> <p>Inhaltsfelder: Daten und ihre Strukturierung Algorithmen</p> <p>Inhaltliche Schwerpunkte: Objekte und Klassen Algorithmen zum Suchen Analyse, Entwurf und Implementierung einfacher Algorithmen</p> <p>Zeitbedarf: 10 Stunden</p>	<p>Thema: Algorithmen zum Sortieren von Daten</p> <p>Zentrale Kompetenzen: Argumentieren Modellieren Implementieren Darstellen und Interpretieren Kommunizieren und Kooperieren</p> <p>Inhaltsfelder: Daten und ihre Strukturierung Algorithmen</p> <p>Inhaltliche Schwerpunkte: Objekte und Klassen Algorithmen zum Sortieren Analyse, Entwurf und Implementierung einfacher Algorithmen</p> <p>Zeitbedarf: 6 Stunden</p>

Übersichtsraster Unterrichtsvorhaben Qualifikationsphase	
Unterrichtsvorhaben Q1-I	Unterrichtsvorhaben Q1-II
<p>Thema: Rekursive Algorithmen und Backtracking in Anwendungskontexten</p> <p>Zentrale Kompetenzen: Argumentieren Modellieren Implementieren Darstellen und Interpretieren Kommunizieren und Kooperieren</p> <p>Inhaltsfelder: Algorithmen Formale Sprachen und Automaten Informatiksysteme Informatik, Mensch und Gesellschaft</p> <p>Inhaltliche Schwerpunkte: Analyse, Entwurf und Implementierung von Algorithmen Algorithmen in ausgewählten informatischen Kontexten Syntax und Semantik einer Programmiersprache Nutzung von Informatiksystemen Grenzen der Automatisierung</p> <p>Zeitbedarf: 20 Stunden</p>	<p>Thema: Modellierung und Implementierung dynamischer Listenstrukturen und deren Anwendungen</p> <p>Zentrale Kompetenzen: Argumentieren Modellieren Implementieren Darstellen und Interpretieren Kommunizieren und Kooperieren</p> <p>Inhaltsfelder: Daten und ihre Strukturierung Algorithmen Formale Sprachen und Automaten Informatiksysteme</p> <p>Inhaltliche Schwerpunkte: Objekte und Klassen Analyse, Entwurf und Implementierung von Algorithmen Algorithmen in ausgewählten informatischen Kontexten Syntax und Semantik einer Programmiersprache Nutzung von Informatiksystemen</p> <p>Zeitbedarf: 25 Stunden</p>
Unterrichtsvorhaben Q1-III	Unterrichtsvorhaben Q1-IV
<p>Thema: Modellierung und Implementierung dynamischer nichtlinearer Datenstrukturen am Beispiel der Binärbäume</p> <p>Zentrale Kompetenzen: Argumentieren Modellieren Implementieren Darstellen und Interpretieren Kommunizieren und Kooperieren</p> <p>Inhaltsfelder: Daten und ihre Strukturierung Algorithmen Formale Sprachen und Automaten Informatiksysteme</p> <p>Inhaltliche Schwerpunkte: Objekte und Klassen</p>	<p>Thema: Sicherheit und Datenschutz in Netzstrukturen</p> <p>Zentrale Kompetenzen: Argumentieren Darstellen und Interpretieren Kommunizieren und Kooperieren</p> <p>Inhaltsfelder: Informatiksysteme Informatik, Mensch und Gesellschaft</p> <p>Inhaltliche Schwerpunkte: Einzelrechner und Rechnernetzwerke Sicherheit Nutzung von Informatiksystemen, Wirkungen der Automatisierung</p> <p>Zeitbedarf: 15 Stunden</p>

Analyse, Entwurf und Implementierung von Algorithmen Algorithmen in ausgewählten informatischen Kontexten Syntax und Semantik einer Programmiersprache Nutzung von Informatiksystemen Zeitbedarf: 20 Stunden	
--	--

Unterrichtsvorhaben Q2-I	Unterrichtsvorhaben Q2-II
Thema: Modellierung und Nutzung relationaler Datenbanken in Anwendungskontexten Zentrale Kompetenzen: Argumentieren, Modellieren, Implementieren Darstellen und Interpretieren Kommunizieren und Kooperieren Inhaltsfelder: Daten und ihre Strukturierung Algorithmen Formale Sprache und Automaten Informatiksysteme Informatik, Mensch und Gesellschaft Inhaltliche Schwerpunkte: Datenbanken Algorithmen in ausgewählten informatischen Kontexten Syntax und Semantik einer Programmiersprache Nutzung von Informatiksystemen Sicherheit Wirkung der Automatisierung Zeitbedarf: 25 Stunden	Thema: Endliche Automaten und Formale Sprachen Zentrale Kompetenzen: Argumentieren Modellieren Implementieren Darstellen und Interpretieren Kommunizieren und Kooperieren Inhaltsfelder: Formale Sprachen und Automaten Informatiksysteme Inhaltliche Schwerpunkte: Endliche Automaten Grammatiken regulärer Sprachen Möglichkeiten und Grenzen von Automaten und formalen Sprachen Nutzung von Informatiksystemen Zeitbedarf: 25 Stunden

Unterrichtsvorhaben Q2-III
Thema: Prinzipielle Arbeitsweise eines Computers und Grenzen der Automatisierbarkeit Zentrale Kompetenzen: Argumentieren, Kommunizieren und Kooperieren Inhaltsfelder: Informatiksysteme, Informatik, Mensch und Gesellschaft Inhaltliche Schwerpunkte: Einzelrechner und Rechnernetzwerke, Grenzen der Automatisierung

2.1.2 Konkretisierte Unterrichtsvorhaben

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

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

Die Schülerinnen und Schüler

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

Ebenso bieten fast alle Unterrichtsvorhaben, in denen Programme implementiert werden, die Gelegenheit, die folgenden Kompetenzen zu erwerben bzw. zu vertiefen:

Schülerinnen und Schüler

- dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D),
- implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I),
- analysieren und erläutern einfache Algorithmen und Programme (A),
- modifizieren einfache Algorithmen und Programme (I),
- implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I),
- testen Programme systematisch anhand von Beispielen (I).
- implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I),
- interpretieren Fehlermeldungen und korrigieren den Quellcode (I),
- nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, ziel führend und verantwortungsbewusst (D).

Unterrichtsvorhaben EP-I

Thema: Grundlagen der objektorientierten Analyse und Programmierung anhand von Beispielkontexten

Leitfragen: Was ist der Unterschied zwischen einer Klasse und einem Objekt? Wie implementiere ich einfache Algorithmen in der Programmiersprache Java?

Vorhabenbezogene Konkretisierung:

Ausgehend von einem vorgegebenen Projekt in Greenfoot wird der Unterschied zwischen einer Klasse und einem Objekt dargestellt. Innerhalb des Projekts werden eigene Klassen durch Kopieren und geringfügiges Modifizieren des Quelltextes erstellt. Der Schwerpunkt liegt dabei auf der Erarbeitung erster Kontrollstrukturen (if-Abfragen, einfache Schleifenkonstrukte). Die Schülerinnen und Schüler lernen grundlegende Datentypen kennen und verwenden sie zur Schleifensteuerung. Darüber hinaus nutzen sie das Schulnetzwerk zur Verwaltung und zum Austausch von Daten.

Sequenzierung des Unterrichtsvorhabens

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<ol style="list-style-type: none"> 1) Unterscheidung von Objekten und Klassen 2) Analyse von Objekten und Klassen in einem Greenfoot-Szenario 3) Grundlagen der Entwicklung in Greenfoot (Klassen: Actor, Animal, World; Methode act(); Objektinteraktion) 4) Modifikation von Klassen durch Implementierung einfacher Algorithmen mit Hilfe von Wiederholungen und bedingten Anweisungen sowie deren Verknüpfung durch UND, ODER und NICHT. 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M) • analysieren und erläutern eine objektorientierte Modellierung (A) • modellieren Klassen mit ihren Attributen und ihren Methoden (M) • analysieren und erläutern einfache Algorithmen und Programme (A) • modifizieren einfache Algorithmen und Programme (I) • entwerfen einfache Algorithmen und stellen sie • umgangssprachlich und grafisch dar (M) • testen Programme schrittweise anhand von Beispielen (I) • stellen die Kommunikation zwischen Objekten grafisch dar (M) • implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I) • interpretieren Fehlermeldungen und korrigieren den Quellcode (I) • 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) 	<p>Programmierungsumgebung: Greenfoot</p> <p>Kara (swisseduc.ch)</p>

Unterrichtsvorhaben EP-II

Thema:

Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand von BlueJ-Projekten

Leitfragen:

Wie kann eine Problemstellung objektorientiert modelliert und dieses Modell in Java implementiert werden?

Vorhabenbezogene Konkretisierung:

Anhand von einer Problemstellung werden erste eigene Klassen modelliert und implementiert. Die Schülerinnen und Schüler lernen die verschiedenen Arten von Methoden (anfragende Methoden und Aufträge, Konstruktor) kennen und verwenden sie. Sie nutzen Variablen zur Speicherung von Werten und zur Steuerung von Kontrollstrukturen. Sie unterscheiden dabei zwischen lokalen und globalen Variablen sowie formalen Parametern. Dazu analysieren die Lernenden vorgegebene BlueJ Projekte und wenden dieses Wissen auf freie Aufgabenstellungen an (z.B BMI-Waage).

Sequenzierung des Unterrichtsvorhabens

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<ol style="list-style-type: none"> 1) Einführung in die objektorientierte Programmierung mit BlueJ (Wiederholung der Begriffe Klasse und Objekt) 2) Modellierung von Klassen und Klassenbeziehungen in UML 3) Differenzierung der verschiedenen Methoden in einer Klasse (anfragende Methoden und Aufträge, Konstruktor) 4) Modellierung von Klassen und Klassenbeziehungen in Java mit Hilfe von BlueJ. 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • 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 und Objekttypen zu (M) • stellen die Kommunikation zwischen Objekten grafisch dar (M) • stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D), • implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I) • stellen den Zustand eines Objekts dar (D) 	<p>Programmierungsumgebung: BlueJ</p> <p>Projekte: „Figuren“, „Zeichnung“ und „Laborkurse“.</p> <p>Beispielhafte Aufgabenstellung: Klasse „BMI-Waage“, „Auto“, „Aufzug“, „Konto“</p>

Unterrichtsvorhaben EP-III

Thema:

Grundlagen digitaler Datenverarbeitung

Leitfragen:

Wie entstand der moderne Computer und wie ist er aufgebaut? Wie verarbeitet und speichert ein Computer Daten?

Vorhabenbezogene Konkretisierung:

Weitere Arbeitsfelder der Fachwissenschaft Informatik werden angerissen. Die Von-Neumann-Architektur und die Binärdarstellung von Zahlen werden als Grundlage moderner Computersysteme vertieft.

Sequenzierung des Unterrichtsvorhabens

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<ol style="list-style-type: none"> 1) Was ist Informatik? 2) Aufbau der Von-Neumann-Architektur als Beispiel des EVA-Prinzips 3) Binärdarstellung von Zahlen sowie Transformationen zwischen und Operationen auf verschiedenen Stellenwertsystemen 	<p>Die Schülerinnen und Schüle</p> <ul style="list-style-type: none"> • stellen ganze Zahlen und Zeichen in Binärcodes dar (D) • interpretieren Binärcodes als Zahlen und Zeichen (D) • beschreiben und erläutern den strukturellen Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der „Von-Neumann-Architektur“ (A) • nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K) • erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (A) 	<p>Geeigneter Modellrechner</p> <p>Internetrecherche + Präsentation</p>

Unterrichtsvorhaben EP-IV

Thema:

Erstellung einer grafischen Benutzeroberfläche und Einführung in das Ereignis-Handling

Leitfragen:

Wie werden benutzerfreundliche Oberflächen zu Programmen erstellt?

Vorhabenbezogene Konkretisierung:

Die Schülerinnen und Schüler erstellen grafische Benutzeroberflächen und verknüpfen diese mit einfachen Algorithmen. Dabei werden das Ereignis-Handling, das Geheimnis-Prinzip und das Model-View-Controller-Konzept verwendet.

Sequenzierung des Unterrichtsvorhabens

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<ol style="list-style-type: none"> 1) Erstellung einer einfachen GUI 2) Einführung einer einfachen Ereignisbehandlung 3) Erstellung einer GUI unter Berücksichtigung des MVC-Konzepts und des Geheimnis-Prinzips 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> - bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A) - ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M) - dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D) - implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I) 	<p>Programmierungsumgebung: BlueJ</p> <p>Projekt „Echo“ Projekt „Farbenwechsel“ Projekt „Getränkeautomat“ (GUI erstellen, Fenster, Button, Label, Textfeld,..)</p>

Unterrichtsvorhaben EP-V

Thema:

Vertiefung von Algorithmen, insbesondere die Behandlung von Suchalgorithmen

Leitfragen:

Wie können Daten effizient gehalten und gesucht werden?

Vorhabenbezogene Konkretisierung:

Die Schülerinnen und Schüler lernen die Datenstruktur Feld als eine Möglichkeit zur Datensammlung kennen. Sie implementieren eine Suche als Anwendung auf der Datenstruktur Feld, bei der der maximale und minimale Wert in einem Feld identifiziert werden. Ausgehend davon werden die Strategien binäre Suche und lineare Suche thematisiert.

Sequenzierung des Unterrichtsvorhabens

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<ol style="list-style-type: none"> 1) Die Datenstruktur Feld zur effizienten Speicherung von Datensammlungen 2) Die Suche von bestimmten Elemente (z.B. Min, Max) als Anwendung auf der Datenstruktur Feld. 3) Erkundung von Strategien für das Suchen auf unsortierten bzw. sortierten Daten. 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> - analysieren Such- und Sortieralgorithmen und wenden sie auf Beispiele an (D) - ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M) - implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I) - interpretieren Fehlermeldungen und korrigieren den Quellcode (I) 	<p>Programmierungsumgebung: BlueJ</p> <p>Einführung in Arrays durch ein Modell oder ein Rollenspiel</p> <p>Tauschen, Minimum, Maximum, Mittelwert als Anwendung auf einem Feld implementieren</p>

Unterrichtsvorhaben EP-VI

Thema:

Algorithmen zum Sortieren von Daten

Leitfragen:

Wie können Daten effizient sortiert werden?

Vorhabenbezogene Konkretisierung:

Ausgehend von unsortierten Datensätzen in einem Feld erarbeiten sich die Lernenden eigene Strategien zum Sortieren. Der Schwerpunkt des Sortierens liegt dabei auf den Algorithmen selbst und nicht auf deren Implementierung.

Sequenzierung des Unterrichtsvorhabens

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<ol style="list-style-type: none"> 1) Die Lernenden erarbeiten sich eigene Sortierverfahren 2) Sie formulieren ihre Verfahren als Pseudocode 3) Die erarbeiteten Verfahren werden hinsichtlich ihrer Effizienz durch die Lernenden untersucht. 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> - analysieren Such- und Sortieralgorithmen und wenden sie auf Beispiele an (D) - entwerfen einen weiteren Algorithmus zum Sortieren (M) - beurteilen die Effizienz von Algorithmen am Beispiel von Sortierverfahren hinsichtlich Zeitaufwand und Speicherplatzbedarf (A) - bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A) 	<p>Sortieren am Modell oder durch ein Rollenspiel.</p> <p>Visualisierung von Sortieralgorithmen durch Medien wie Matheprisma und YouTube.</p>

Unterrichtsvorhaben Q1-I

Thema:

Rekursive Algorithmen und Backtracking in Anwendungskontexten

Leitfragen:

Wie können komplexe, rekursiv definierte Probleme informatisch gelöst werden? Gibt es schnelle (rekursiv definierte) Sortier- und Suchverfahren?

Vorhabenbezogene Konkretisierung:

Ausgehend von unsortierten Datensätzen in einem Feld erarbeiten sich die Lernenden eigenen Strategien zum Sortieren. Der Schwerpunkt des Sortierens liegt dabei auf den Algorithmen selbst und nicht auf deren Implementierung.

Ausgehend vom einem Problem wie z. B. "Türme von Hanoi" wird Rekursion als fundamentale Idee der Informatik zunächst im mathematischen, danach aber auch im informatischen Zusammenhang angewendet. Dabei wird zwischen linearen und verzweigten Rekursionen unterschieden und das Laufzeitverhalten bei hoher Rekursionstiefe analysiert.

Verschiedene NP-vollständige Probleme (wie z. B. Rucksack, n-Damen, Springer, Irrgarten, etc.) werden algorithmisch rekursiv formuliert.

Bereits bekannte Such- und Sortierverfahren (z. B. Sortieren durch Einfügen, Sortieren durch Auswahl, Sequentielle Suche) werden rekursiv formuliert und durch leistungsfähigere Verfahren (z. B. Quicksort, Mergesort, Heapsort, Binäre Suche) ergänzt. Beispielhaft wird ein Such- oder Sortierverfahren sowohl iterativ wie auch rekursiv formuliert.

Sequenzierung des Unterrichtsvorhabens

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Entwicklung der Rekursion als fundamentale Idee der Informatik</p> <ul style="list-style-type: none"> • rekursive Formeln • rekursive Funktionen / Methoden • rekursive Programmierung 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • analysieren und erläutern Algorithmen und Programme (A), • modifizieren Algorithmen und Programme (I), 	<p>Türme von Hanoi mit Schwerpunkt auf</p> <ul style="list-style-type: none"> • Zahl der Versetzungsoperationen • Protokollierung der Versetzungen
<p>2. Rekursion in mathematischen und informatischen Kontexten</p> <ul style="list-style-type: none"> • Rekursion in mathematischen Kontexten • Analyse und Darstellung des rekursiven Ablaufs einer Methode • Analyse des Laufzeitverhaltens linearer und verzweigter Rekursion 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • analysieren und erläutern Algorithmen und Programme (A), • modifizieren Algorithmen und Programme (I), • stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D), • testen Programme systematisch anhand von Beispielen (I). • untersuchen und beurteilen Grenzen des Problemlösens mit Informatiksystemen (A). 	<ul style="list-style-type: none"> • Fakultätsfunktion (lineare Rekursion) • Fibonacci-Funktion (verzweigte Rekursion) • z.B. ggT (verzweigte Rekursion) • evtl. Fraktale (Kochkurve, Sierpinski dreieck, etc.)
<p>3. NP-vollständige Probleme lösen mit Backtracking</p> <ul style="list-style-type: none"> • Erarbeitung verschiedener NP-vollständiger Probleme • Algorithmische Beschreibung einer Lösungs-idee 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • analysieren und erläutern Algorithmen und Programme (A), • modifizieren Algorithmen und Programme (I), • stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D), • entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“ (M) • testen Programme systematisch anhand von Beispielen (I). • untersuchen und beurteilen Grenzen des Problemlösens mit Informatiksystemen (A). 	<ul style="list-style-type: none"> • z.B. Springerproblem, Rucksackproblem, Irrgartenproblem

<p>4. Effiziente Sortierverfahren / Suchverfahren</p> <ul style="list-style-type: none"> • Wiederholung bereits bekannter Sortier- und Suchverfahren als rekursiver Algorithmus • Erarbeitung eines Sortierverfahrens der Laufzeit $O(n \cdot \log(n))$ • Erarbeitung eines Suchverfahrens der Laufzeit $O(\log(n))$. 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • analysieren und erläutern Algorithmen und Programme (A), • modifizieren Algorithmen und Programme (I), • stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D), • entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“ (M), • testen Programme systematisch anhand von Beispielen (I). • implementieren und erläutern iterative und rekursive Such- und Sortierverfahren (I), • beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (A), • untersuchen und beurteilen Grenzen des Problemlösens mit Informatiksystemen (A). 	<ul style="list-style-type: none"> • Demonstrationsprogramm zur Visualisierung von Sortierverfahren • Quicksortvisualisierung zur Erarbeitung der Idee • Suchspiel zur Erarbeitung der Binären Suche
--	--	---

Unterrichtsvorhaben Q1-II

Thema:

Modellierung und Implementierung dynamischer Listenstrukturen und deren Anwendungen

Leitfragen:

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

Vorhabenbezogene Konkretisierung:

Nach Analyse einer Problemstellung in einem geeigneten Anwendungskontext, in dem Daten nach dem Last-In-First-Out-Prinzip verwaltet werden, werden der Aufbau von Stapeln am Beispiel dargestellt und die Operationen der Klasse Stack anhand der Abiturklasse erläutert. Anschließend werden für die Anwendung notwendige Klassen modelliert und implementiert.

Danach wird die Anwendung modifiziert, um den Umgang mit der Datenstruktur zu üben. Anhand einer Anwendung, in der Daten nach dem First-In-First-Out-Prinzip verwaltet werden, werden Unterschiede zwischen den Datenstrukturen Schlange und Stapel erarbeitet. Dabei werden die Operationen der Datenstruktur Schlange thematisiert und die entsprechende Abiturklasse Queue verwendet.

Um einfacher an Objekte zu gelangen, die zwischen anderen gespeichert sind, wird die Klasse List gemäß der Abiturklasse eingeführt und in einem Anwendungskontext verwendet.

(Je nach thematisierten Anwendungskontexten ist eine andere Reihenfolge bei der Behandlung der oben genannten Datenstrukturen möglich und zulässig.)

In mindestens einem weiteren Anwendungskontext wird die Verwaltung von Daten in Schlangen, Stapeln oder Listen vertieft. Modellierungen werden dabei in Entwurfs- und Implementationsdiagrammen dargestellt.

Sequenzierung des Unterrichtsvorhabens

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Die Datenstruktur Stapel im Anwendungskontext unter Nutzung der Klasse Stack</p> <ul style="list-style-type: none"> • Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen • Erarbeitung der Funktionalität der Klasse Stack • Modellierung und Implementierung der Anwendung unter Verwendung der Klasse Stack. 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), • stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D), • modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M), • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M), • ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M), 	<p>Visualisierungsprogramm zu dynamischen Datenstrukturen</p>
<p>2. Die Datenstruktur Schlange im Anwendungskontext unter Nutzung der Klasse Queue</p> <ul style="list-style-type: none"> • Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen • Erarbeitung der Funktionalität der Klasse Queue • Modellierung und Implementierung der Anwendung unter Verwendung der Klasse Queue. 	<ul style="list-style-type: none"> • stellen die Kommunikation zwischen Objekten grafisch dar (D), • stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D), • dokumentieren Klassen (D), • analysieren und erläutern objektorientierte Modellierungen (A), • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I). • analysieren und erläutern Algorithmen und Programme (A), • modifizieren Algorithmen und Programme (I), • stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D), • implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), • testen Programme systematisch anhand von Beispielen (I). • erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A) 	<p>Visualisierungsprogramm zu dynamischen Datenstrukturen</p>

<p>3. Die Datenstruktur lineare Liste im Anwendungskontext unter Nutzung der Klasse List</p> <ul style="list-style-type: none"> • Erarbeitung der Vorteile der Klasse List im Gegensatz zu den bereits bekannten linearen Strukturen • Modellierung und Implementierung einer kontextbezogenen Anwendung unter Verwendung der Klasse List. 	<p>(Siehe vorherige Seite)</p>	<p>Visualisierungsprogramm zu dynamischen Datenstrukturen</p>
<p>4. Vertiefung / Anwendung einer linearen Datenstruktur im Anwendungskontext.</p>	<p>zusätzlich: Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M), • verwenden bei der Modellierung geeigneter Problemstellungen Möglichkeiten der Polymorphie (M), 	<p>Umsetzung in einem größeren Projekt.</p>

Unterrichtsvorhaben Q1-III

Thema:

Modellierung und Implementierung von Anwendungen mit dynamischen, nichtlinearen Datenstrukturen am Beispiel der Binärbäume

Leitfragen:

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

Vorhabenbezogene Konkretisierung:

Anhand von Beispielen für Baumstrukturen werden grundlegende Begriffe eingeführt und der rekursive Aufbau binärer Bäume dargestellt.

Anschließend werden für eine Problemstellung in einem der Anwendungskontexte Klassen modelliert und implementiert. Dabei werden die Operationen der Datenstruktur Binärbaum thematisiert und die entsprechende Klasse BinaryTree der Vorgaben für das Zentralabitur NRW verwendet. Klassen und ihre Beziehungen werden in Entwurfs- und Implementationsdiagrammen dargestellt. Die Funktionsweise von Methoden wird anhand grafischer Darstellungen von Binärbäumen erläutert.

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

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

Zu einer Problemstellung in einem entsprechenden Anwendungskontext werden die Operationen der Datenstruktur Suchbaum thematisiert und unter der Verwendung der Klasse BinarySearchTree der Vorgaben für das Zentralabitur weitere Klassen oder Methoden in diesem Kontext modelliert und implementiert. Die Suchbäume werden wie zuvor auch grafisch dargestellt.

Die Verwendung von binären Bäumen und Suchbäumen wird anhand weiterer Problemstellungen oder anderer Kontexte weiter geübt.

Sequenzierung des Unterrichtsvorhabens

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Analyse von Baumstrukturen in verschiedenen Kontexten</p> <ul style="list-style-type: none"> • Grundlegende Begriffe (Grad, Tiefe, Höhe, Blatt, Inhalt, Teilbaum, Ebene, Vollständigkeit) • Aufbau und Darstellung von binären Bäumen anhand von Baumstrukturen in verschiedenen Kontexten 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), • stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D), • modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M), • modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M), • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare und nichtlineare Datensammlungen zu (M), • verwenden bei der Modellierung geeigneter Problemstellungen Möglichkeiten der Polymorphie (M), • ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M), • stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D), • dokumentieren Klassen (D), • analysieren und erläutern objektorientierte Modellierungen (A), • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I). • analysieren und erläutern Algorithmen und Programme (A), • modifizieren Algorithmen und Programme (I), • stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D), • entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“ (M), • implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), • testen Programme systematisch anhand von Beispielen (I). • erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A) 	<p>z.B. Tiere raten, Akinator, Termbaum, Morsebaum, Spielbäume.</p>

Unterrichtsvorhaben Q1-IV

Thema:

Sicherheit und Datenschutz in Netzstrukturen

Leitfragen:

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

Vorhabenbezogene Konkretisierung:

Ausgehend von einer Kommunikation zwischen zwei Kommunikationspartnern über eine einfache Leitung wird die Notwendigkeiten einer Datenübertragung erarbeitet. Die Schichten des TCP/IP-Schichtenmodells werden beispielgebunden erarbeitet (Basisbandübertragungsverfahren, Prüfverfahren, Vermittlungsschicht, Anwendungsprotokoll) und an einer Simulationssoftware getestet. Verschiedene Netzwerk-Topologien werden entwickelt und in Client-Server-Anwendungen simuliert.

Über die Sicherheit von Netzwerkanwendungen wird das Augenmerk auf verschiedene symmetrische und asymmetrische kryptografische Verfahren gelenkt, welche analysiert und erläutert werden. Fallbeispiele zur Datenschutzproblematik und zum Urheberrecht runden das Unterrichtsvorhaben ab.

Sequenzierung des Unterrichtsvorhabens

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Schichten des TCP/IP-Protokolls</p> <ul style="list-style-type: none"> • Erarbeitung der Notwendigkeiten einer Netzwerkkommunikation • Erarbeitung der Schichten des TCP/IP-Protokolls: Basisbandübertragung, Prüfverfahren, Routing/Vermittlungsschicht, Anwendungsprotokolle 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • beschreiben und erläutern Netzwerk-Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken (A), 	<ul style="list-style-type: none"> • Arbeitsblätter zur Einführung in Netzwerke
<p>2. Simulation von Netzwerken / Netzwerk-Topologien</p> <ul style="list-style-type: none"> • Erarbeitung der gängigen Topologien • Simulation von Client-Server-Anwendungen • Simulation von Protokollen der Anwendungsschicht (z.B. POP3, SMTP, etc.) 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • beschreiben und erläutern Netzwerk-Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken (A), 	<ul style="list-style-type: none"> • Simulationssoftware FILIUS • Arbeitsblätter und Skript zu FILIUS
<p>3. Analyse und Erläuterung kryptografischer Verfahren</p> <ul style="list-style-type: none"> • Erläuterung symmetrischer Verfahren: monoalphabetisch: Cäsar, polyalphabetisch: Vigenère • Erläuterung asymmetrischer Verfahren: RSA, Diffie-Hellman • Analyse der Sicherheit verschiedener Verfahren und Auswirkungen auf den Datenschutz/Urheberrecht 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • analysieren und erläutern Algorithmen und Programme (A), • stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D), • analysieren und erläutern Eigenschaften und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren (A). • untersuchen und bewerten anhand von Fallbeispielen Auswirkungen des Einsatzes von Informatiksystemen sowie Aspekte der Sicherheit von Informatiksystemen, des Datenschutzes 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). 	<ul style="list-style-type: none"> • Arbeitsblätter oder Referate zu kryptografischen Verfahren • z.B. CrypTool • z.B. Materialien von klicksafe (Zusatzmodule „Nicht alles was geht, ist auch erlaubt“, „Ich bin öffentlich ganz privat“, „Datenschutz im Internet“) • z.B. Matheprisma (RSA,...)

Unterrichtsvorhaben Q2-I

Thema:

Modellierung und Nutzung relationaler Datenbanken in Anwendungskontexten

Leitfragen:

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

Vorhabenbezogene Konkretisierung:

Ausgehend von einer konkreten Anwendungssituation entwickeln die Schülerinnen und Schüler Ideen zur Modellierung von Daten und erkennen die Vorzüge von Datenbanksystemen.

In weiteren Anwendungskontexten müssen Datenbanken entwickelt werden, um Daten zu speichern und Informationen für die Beantwortung von möglicherweise auftretenden Fragen zur Verfügung zu stellen. Dafür ermitteln Schülerinnen und Schüler in den Anwendungssituationen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten und stellen diese in Entity-Relationship-Modellen dar. Entity-Relationship-Modelle werden interpretiert und erläutert, modifiziert und in das Relationale Modell überführt.

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

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

Mit Hilfe von SQL-Anweisungen können anschließend im Kontext relevante Informationen aus der Datenbank extrahiert werden. Die Operationen der Relationalalgebra werden mit SQL-Abfragen simuliert.

Anhand von Fallbeispielen werden Probleme bei der Nutzung von Datenbanksystemen aufgezeigt und im Hinblick auf gesellschaftliche Auswirkungen diskutiert.

(Die exakte Abfolge der Teilvorhaben sowie die Verzahnung theoretischer und praktischer Anteile kann hierbei je nach Anwendungskontext flexibel angepasst werden.)

Sequenzierung des Unterrichtsvorhabens

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Modellierung von relationalen Datenbanken</p> <ul style="list-style-type: none"> • Entity-Relationship-Diagramm: Ermittlung von Entitäten, zugehörigen Attributen, Relationen und Kardinalitäten in Anwendungssituationen und Modellierung eines Datenbankentwurfs in Form eines Entity-Relationship-Diagramms Erläuterung und Modifizierung einer Datenbankmodellierung • Entwicklung einer Datenbank aus einem Datenbankentwurf: Modellierung eines relationalen Datenbankschemas zu einem Entity-Relationship-Diagramm inklusive der Bestimmung von Primär- und Sekundärschlüsseln • Redundanz, Konsistenz und Normalformen: Untersuchung einer Datenbank hinsichtlich Konsistenz und Redundanz in einer Anwendungssituation. Überprüfung von Datenbankschemata hinsichtlich der 1. bis 3. Normalform und Normalisierung (um Redundanzen zu vermeiden und Konsistenz zu gewährleisten) 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten (M), • stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten mit Kardinalitäten in einem Entity-Relationship-Diagramm grafisch dar (D), • modifizieren eine Datenbankmodellierung (M), • modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema (M), • bestimmen Primär- und Sekundärschlüssel (M), • analysieren und erläutern eine Datenbankmodellierung (A), • erläutern die Eigenschaften normalisierter Datenbankschemata (A), • überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D), • überführen Datenbankschemata in die 1. bis 3. Normalform (M). 	<ul style="list-style-type: none"> • Arbeitsblätter zur Einführung in Datenbanken oder Tutorial „Entwurf von Datenbanken“ (http://www.inf-schule.de/information/datenbanksysteme/ermodelle)

<p>2. Nutzung von relationalen Datenbanken</p> <ul style="list-style-type: none"> • Aufbau von Datenbanken und Grundbegriffe: Entwicklung von Fragestellungen zur vorhandenen Datenbank Analyse der Struktur der vorgegebenen Datenbank und Erarbeitung der Begriffe Tabelle, Attribut, Datensatz, Datentyp, Primärschlüssel, Fremdschlüssel, Datenbankschema • SQL-Abfragen: Analyse vorgegebener SQL-Abfragen und Erarbeitung der Sprachelemente von SQL (SELECT (DISTINCT) ...FROM, WHERE, AND, OR, NOT) auf einer Tabelle Analyse und Erarbeitung von SQL-Abfragen auf einer und mehrerer Tabelle zur Beantwortung der Fragestellungen (JOIN, UNION, AS, GROUP BY, ORDER BY, ASC, DESC, COUNT, MAX, MIN, SUM, Arithmetische Operatoren: +, -, *, /, (...), Vergleichsoperatoren: =, <>, >, <, >=, <=, LIKE, BETWEEN, IN, IS NULL) • Vertiefung am Beispiel der Relationen-Algebra 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • modifizieren eine Datenbankmodellierung (M), • bestimmen Primär- und Sekundärschlüssel (M), • analysieren und erläutern eine Datenbankmodellierung (A), • ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen (D), • analysieren und erläutern die Syntax und Semantik einer Datenbankabfrage (A), • verwenden die Syntax und Semantik einer Datenbankabfragesprache, um Informationen aus einem Datenbanksystem zu extrahieren (I). • erläutern Eigenschaften und Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nutzung (A), 	<ul style="list-style-type: none"> • z.B. Tutorial der Seite SQL-Zoo (http://sqlzoo.net/wiki/Main_Page), Videocenter (http://videocenter.schule.de/), SQL-Tutorial Hessen (http://www.imoodle.de/sqltutorial/home.html)
<p>3. Gesellschaftliche Auswirkungen der Nutzung von Datenbanksystemen</p>	<ul style="list-style-type: none"> • untersuchen und bewerten anhand von Fallbeispielen Auswirkungen des Einsatzes von Informatiksystemen sowie Aspekte der Sicherheit von Informatiksystemen, des Datenschutzes 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). 	<ul style="list-style-type: none"> • Fallbeispiele zur Nutzung von Datenbanksystemen (z.B. aus der aktuellen Presse) • gegebenenfalls Referate zu dem Thema

Unterrichtsvorhaben Q2-II

Thema:

Endliche Automaten und Formale Sprachen

Leitfragen:

Wie kann man endliche Automaten genau beschreiben? Wie können endliche Automaten modelliert werden? Wie können Sprachen durch Grammatiken beschrieben werden? Welche Zusammenhänge gibt es zwischen formalen Sprachen, Automaten und Grammatiken?

Konkretisierung:

Ausgehend von einem konkreten Anwendungsbeispiel entwickeln die Schülerinnen und Schüler das Modell der Grammatik einer formalen Sprache und das Modell des endlichen Automaten. Die Schülerinnen und Schüler überführen Automaten in verschiedene Darstellungsformen und ermitteln die akzeptierte Sprache eines Automaten (z. B. in Form von regulären Ausdrücken). An einem Beispiel wird ein nichtdeterministischer Akzeptor als Alternative gegenüber einem entsprechenden deterministischen Akzeptor eingeführt.

Der Zusammenhang zwischen endlichen Automaten und regulären Grammatiken wird durch die Entwicklung allgemeingültiger Verfahren zur Transformation zwischen Automat und Grammatik dargestellt. Die Unzulänglichkeit endlicher Automaten und regulärer Grammatiken wird an Beispielen verdeutlicht.

(Die exakte Abfolge der Teilvorhaben sowie die Verzahnung theoretischer und praktischer Anteile kann hierbei je nach Anwendungskontext flexibel angepasst werden.)

Sequenzierung des Unterrichtsvorhabens

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Einführung in Automaten/Grammatiken</p> <ul style="list-style-type: none"> • Grammatiken: Grammatik einer natürlichen Sprache Grammatik einer künstlichen Sprache Idee des Parsens • Automaten: erkennender Automat zu Symbolen einer Sprache Modell des endlichen Automaten Darstellungsformen Sprache eines Automaten als regulärer Ausdruck nichtdeterministische Automaten 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • analysieren und erläutern die Eigenschaften endlicher Automaten bestimmten Eingaben (A), • ermitteln die Sprache, die ein endlicher Automat akzeptiert (D), • entwickeln und modifizieren zu einer Problemstellung endliche Automaten (M), • stellen endliche Automaten in Tabellen oder Graphen dar und überführen sie in die jeweils andere Darstellungsform (D), • entwickeln zur Grammatik einer regulären Sprache einen zugehörigen endlichen Automaten (M). 	<ul style="list-style-type: none"> • Mögliches Vorgehen anhand des Buches „Informatik 5“ des Klett Verlages. • Automaten-Simulationsprogramm
<p>2. Zusammenhang zwischen endlichen Automaten und regulären Grammatiken</p> <ul style="list-style-type: none"> • reguläre Grammatik: Definition Anwendungen • Zusammenhang zu endlichen Automaten • Grenzen der endlichen Automaten/regulären Grammatiken 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • analysieren und erläutern Grammatiken regulärer Sprachen (A), • modifizieren Grammatiken regulärer Sprachen (M), • ermitteln die formale Sprache, die durch eine Grammatik erzeugt wird (A), • entwickeln zu einer regulären Sprache eine Grammatik, die die Sprache erzeugt (M), • entwickeln zur akzeptierten Sprache eines Automaten eine zugehörige Grammatik (M), • beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken (D), • zeigen die Grenzen endlicher Automaten und regulärer Grammatiken im Anwendungszusammenhang auf (A). 	<ul style="list-style-type: none"> • Mögliches Vorgehen anhand des Buches „Informatik 5“ des Klett Verlages. • Automaten-Simulationsprogramm

Unterrichtsvorhaben Q2-II

Thema:

Prinzipielle Arbeitsweise eines Computers und Grenzen der Automatisierbarkeit

Leitfragen:

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

Konkretisierung:

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

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

Sequenzierung des Unterrichtsvorhabens

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Von-Neumann-Architektur und die Ausführung maschinennaher Programme:</p> <ul style="list-style-type: none"> • prinzipieller Aufbau einer von Neumann-Architektur mit CPU, Rechenwerk, Steuerwerk, Register und Hauptspeicher • einige maschinennahe Befehle und ihre Repräsentation in einem Binär-Code, der in einem Register gespeichert werden kann • Analyse und Erläuterung der Funktionsweise eines einfachen maschinennahen Programms 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • erläutern die Ausführung eines einfachen maschinennahen Programms sowie die Datenspeicherung auf einer „Von-Neumann-Architektur“ (A), untersuchen und beurteilen Grenzen des Problemlösens mit Informatiksystemen (A). 	<p>geeigneter Modellrechner, z.B.</p> <ul style="list-style-type: none"> • Johnny • WinAli • Microprocessor Simulator • von-Neumann Rechner der Uni Hamburg (https://tams.informatik.uni-hamburg.de/applets/baukasten/DA/VNR_Einleitung.html) <p>Mögliches Vorgehen anhand des Buches „Informatik 5“ des Klett Verlages.</p>
<p>2. Grenzen der Automatisierbarkeit:</p> <ul style="list-style-type: none"> • Vorstellung des Halteproblems • Unlösbarkeit des Halteproblems • Beurteilung des Einsatzes von Informatiksystemen hinsichtlich prinzipieller Möglichkeiten und prinzipieller Grenzen 		<p>Thematisierung des Halteproblems an einem geeigneten Beispiel (z.B. Arbeitsblatt Halteproblem in Java)</p>

2.2 Grundsätze der fachmethodischen und fachdidaktischen Arbeit

Unter Berücksichtigung des Schulprogramms hat die Fachkonferenz Informatik des Siegtal-Gymnasiums die folgenden fachmethodischen und fachdidaktischen Grundsätze beschlossen.

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

Fachliche Grundsätze

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

2.3 Grundsätze der Leistungsbewertung und Leistungsrückmeldung

Auf der Grundlage von §13 - §16 der APO-GOST sowie Kapitel 3 des Kernlehrplans Informatik für die gymnasiale Oberstufe hat die Fachkonferenz Informatik des Siegtal-Gymnasiums Eitorf im Einklang mit dem entsprechenden schulbezogenen Konzept die nachfolgenden Grundsätze zur Leistungsbewertung und Leistungsrückmeldung beschlossen. Die nachfolgenden Absprachen stellen die Minimalanforderungen an das lerngruppenübergreifende gemeinsame Handeln der Fachgruppenmitglieder dar. Bezogen auf die einzelne Lerngruppe kommen ergänzend weitere der in den Folgeabschnitten genannten Instrumente der Leistungsüberprüfung zum Einsatz.

2.3.1 Beurteilungsbereich Klausuren

Bei der Formulierung von Aufgaben werden die für die Abiturprüfungen geltenden Operatoren des Faches Informatik schrittweise eingeführt, erläutert und dann im Rahmen der Aufgabenstellungen für die Klausuren benutzt.

Instrumente:

- Einführungsphase: 1 Klausur je Halbjahr, Dauer der Klausur: 2 Unterrichtsstunden
- Grundkurse Q 1: 2 Klausuren je Halbjahr, Dauer der Klausuren: 2 Unterrichtsstunden
- Grundkurse Q 2.1: 2 Klausuren, Dauer der Klausuren: 3 Unterrichtsstunden
- Grundkurse Q 2.2: 1 Klausur unter Abiturbedingungen

Anstelle einer Klausur kann gemäß dem Beschluss der Lehrerkonferenz in Q 1.2 eine Facharbeit geschrieben werden.

Die Aufgabentypen, sowie die Anforderungsbereiche I-III sind entsprechend den Vorgaben in Kapitel 3 des Kernlehrplans zu beachten. (Siehe Kernlehrplan, S. 40 *Überprüfungsformen*)

Bewertungskriterien:

Die Bewertung der schriftlichen Leistungen in Klausuren erfolgt über ein Raster mit Hilfspunkten, die im Erwartungshorizont den einzelnen Kriterien zugeordnet sind.

Spätestens ab der Qualifikationsphase orientiert sich die Zuordnung der Hilfspunktsumme zu den Notenstufen an dem Zuordnungsschema des Zentralabiturs.

Von diesem kann aber im Einzelfall begründet abgewichen werden, wenn sich z.B. besonders originelle Teillösungen nicht durch Hilfspunkte gemäß den Kriterien des Erwartungshorizontes abbilden lassen oder eine Abwertung wegen besonders schwacher Darstellung (APO-GOST §13 (2)) angemessen erscheint.

Die Note ausreichend (5 Punkte) soll bei Erreichen von 45 % der Hilfspunkte erteilt werden.

2.3.2 Beurteilungsbereich Sonstige Mitarbeit

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

Leistungsaspekte:

Mündliche und praktische Leistungen

- Beteiligung am Unterrichtsgespräch
- Zusammenfassungen zur Vor- und Nachbereitung des Unterrichts
- Präsentation von Arbeitsergebnissen
- Referate
- Mitarbeit in Partner-/Gruppenarbeitsphasen
- Praktische Leistungen am Computer (Implementierung, Test und Anwendung von Informatiksystemen)

Über die Durchführung von schriftliche Übungen entscheidet die Lehrkraft. Schriftliche Übung dauern ca. 20 Minuten und umfassen den Stoff der letzten ca. 4–6 Stunden.

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.

3 Entscheidungen zu fach- und unterrichtsübergreifenden Fragen

Die Fachkonferenz Informatik hat sich im Rahmen des Schulprogramms für folgenden zentralen Schwerpunkt entschieden:

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

Die Informatikfachschaft nutzt die sich flexibel ergebenden Gelegenheiten, mit anderen Fächern zu kooperieren.

4 Qualitätssicherung und Evaluation

Das schulinterne Curriculum stellt keine starre Größe dar, sondern ist als „lebendes Dokument“ zu betrachten. Dementsprechend sind die Inhalte stetig zu überprüfen, um ggf. Modifikationen vornehmen zu können. Die Fachkonferenz (als professionelle Lerngemeinschaft) trägt durch diesen Prozess zur Qualitätsentwicklung und damit zur Qualitätssicherung des Faches bei.

Das schulinterne Curriculum (siehe 2.1) ist zunächst bis 2017 für den ersten Durchgang durch die gymnasiale Oberstufe nach Erlass des Kernlehrplanes verbindlich. Jeweils vor Beginn eines neuen Schuljahres, d.h. erstmalig nach Ende der Einführungsphase im Sommer 2015 werden in einer Sitzung der Fachkonferenz für die nachfolgenden Jahrgänge zwingend erforderlich erscheinende Veränderungen diskutiert und ggf. beschlossen, um erkannten ungünstigen Entscheidungen schnellstmöglich entgegenwirken zu können.

Nach Abschluss des Abiturs 2017 wird eine Arbeitsgruppe aus den zu diesem Zeitpunkt in der gymnasialen Oberstufe unterrichtenden Lehrkräften auf der Grundlage ihrer Unterrichtserfahrungen eine Gesamtsicht des schulinternen Curriculums vornehmen und eine Beschlussvorlage für die erste Fachkonferenz des folgenden Schuljahres erstellen.