

Gesamtschule Marienheide

Schulinterner Lehrplan

Informatik

Stand: 01.07.2014

Die Fachgruppe Informatik an der Gesamtschule Marienheide

Das Fach Informatik wird an Oberstufe der Gesamtschule Marienheide als Wahlfach (Grundkurs) angeboten. Um insbesondere Schülerinnen und Schülern, die in der Sekundarstufe I keinen Informatikunterricht besucht haben, gerecht zu werden, wird in den Kursen der Oberstufe besonderer Wert darauf gelegt, dass keine Vorkenntnisse aus der Sekundarstufe I zum erfolgreichen Bestehen der Kurse erforderlich sind.

Durch projektartiges Vorgehen, offene Aufgaben und Möglichkeiten, Problemlösungen zu verfeinern und zu optimieren, entspricht der Informatikunterricht in besonderem Maße den Erziehungszielen, Leistungsbereitschaft zu fördern ohne zu überfordern. Darüber hinaus trägt er zu einer breitgefächerten Allgemeinbildung bei, bietet gleichzeitig Raum für individuelle Spezialisierungen und ermöglicht verantwortungsvolles Handeln in einer sich schnell wandelnden und von technischen Fortschritten geprägten Welt.

Zurzeit besteht die Fachschaft aus drei Lehrkräften mit Sek-II-Lehrbefähigung aufweisen.

Fachvorsitzender SII:	Manfred Knobloch
Stellvertreter:	Matthias Wörner

Entscheidungen zum Unterricht

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: Im "Übersichtsraster Unterrichtsvorhaben" wird die für alle Lehrerinnen und Lehrer gemäß Fachkonferenzbeschluss verbindliche Verteilung der Unterrichtsvorhaben dargestellt. Es verschafft einen schnellen Überblick über die Zuordnungen der Unterrichtsvorhaben zu den einzelnen Jahrgangsstufen sowie den im Kernlehrplan genannten Kompetenzen, Inhaltsfeldern und inhaltlichen Schwerpunkten zu verschaffen. Der ausgewiesene Zeitbedarf versteht sich als grober Orientierungswert, der bei Bedarf über- oder unterschritten werden kann.

Während der Fachkonferenzbeschluss zum "Übersichtsraster Unterrichtsvorhaben" zur Gewährleistung vergleichbarer Standards sowie zur Absicherung von Lerngruppenübertritten und Lehrkraftwechseln für alle Mitglieder der Fachkonferenz Bindekraft entfalten soll, besitzt die exemplarische Ausweisung "konkretisierter Unterrichtsvorhaben" empfehlenden Charakter. Abweichungen von den vorgeschlagenen Vorgehensweisen bezüglich der konkretisierten Unterrichtsvorhaben sind im Rahmen der pädagogischen Freiheit der Lehrkräfte jederzeit möglich. Sicherzustellen bleibt allerdings auch hier, dass im Rahmen der Umsetzung der Unterrichtsvorhaben insgesamt alle konkretisierten Kompetenzerwartungen des

Kernlehrplans Berücksichtigung finden.

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

Übersichtsraster Unterrichtsvorhaben

Einführungsphase

Einführungsphase	
<i>Unterrichtsvorhaben E-I</i>	<i>Unterrichtsvorhaben E-II</i>
<p>Thema: <i>Einführung in die Nutzung von Informatiksystemen und in grundlegende Begrifflichkeiten</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none">• Argumentieren• Darstellen und Interpretieren• Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none">• Informatiksysteme• Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none">• Einzelrechner• Dateisystem• Internet• Einsatz von Informatiksystemen	<p>Thema: <i>Geschichte der digitalen Datenverarbeitung und die Grundlagen des Datenschutzes</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none">• Argumentieren• Darstellen und Interpretieren• Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none">• Informatik, Mensch und Gesellschaft• Informatiksysteme <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none">• Wirkungen der Automatisierung• Geschichte der automatischen Datenverarbeitung• Digitalisierung

Einführungsphase

Unterrichtsvorhaben E-III

Thema:

Grundlagen der objektorientierten Analyse, Modellierung und Implementierung

Zentrale Kompetenzen:

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

Inhaltsfelder:

- Daten und ihre Strukturierung
- Formale Sprachen und Automaten

Inhaltliche Schwerpunkte:

- Objekte und Klassen
- Syntax und Semantik einer Programmiersprache

Unterrichtsvorhaben E-IV

Thema:

Grundlagen der objektorientierten Programmierung und algorithmischer Grundstrukturen in Java

Zentrale Kompetenzen:

- Argumentieren
- Modellieren
- Implementieren
- Kommunizieren und Kooperieren

Inhaltsfelder:

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

Inhaltliche Schwerpunkte:

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

Einführungsphase

Unterrichtsvorhaben E-V

Thema:

Modellierung und Implementierung von Klassen- und Objektbeziehungen

Zentrale Kompetenzen:

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

Inhaltsfelder:

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

Inhaltliche Schwerpunkte:

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

Unterrichtsvorhaben E-VI

Thema:

Such- und Sortieralgorithmen anhand kontextbezogener Beispiele

Zentrale Kompetenzen:

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

Inhaltsfelder:

- Algorithmen

Inhaltliche Schwerpunkte:

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

Konkretisierte Unterrichtsvorhaben

Im Folgenden sollen die zuvor aufgeführten Unterrichtsvorhaben konkretisiert werden.

Einführungsphase

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

Die Schülerinnen und Schüler

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

Unterrichtsvorhaben EF-I

Thema: Einführung in die Nutzung von Informatiksystemen und in grundlegende Begrifflichkeiten

Leitfragen: *Womit beschäftigt sich die Wissenschaft der Informatik? Wie kann die in der Schule vorhandene informatische Ausstattung genutzt werden?*

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
<p>1. Information, deren Kodierung und Speicherung</p> <ul style="list-style-type: none">(a) Informatik als Wissenschaft der Verarbeitung von Informationen(b) Darstellung von Informationen in Schrift, Bild und Ton(c) Speichern von Daten mit informatischen Systemen am Beispiel der Schulrechner(d) Vereinbarung von Richtlinien zur Datenspeicherung auf den Schulrechnern (z.B. Ordnerstruktur, Dateibezeichner usw.)	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none">• beschreiben und erläutern den Aufbau und die Arbeitsweise singularer Rechner am Beispiel der „Von-Neumann-Architektur“ (A),• nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D),• nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K).
<p>2. Informations- und Datenübermittlung in Netzen</p> <ul style="list-style-type: none">(a) Informatische Kommunikation in Rechnernetzen am Beispiel des Schulnetzwerks (z.B. Benutzeranmeldung, Netzwerkordner, Zugriffsrechte, Client-Server)(b) Grundlagen der technischen Umsetzung von Rechnerkommunikation am Beispiel des Internets (z.B. Netzwerkadresse, Paketvermittlung, Protokoll)(c) Richtlinien zum verantwortungsvollen Umgang mit dem Internet	
<p>3. Aufbau informatischer Systeme</p> <ul style="list-style-type: none">(a) Identifikation typischer Komponenten informatischer Systeme und anschließende Beschränkung auf das Wesentliche, Herleitung der „Von-Neumann-Architektur“(b) Identifikation des EVA-Prinzips (Eingabe-Verarbeitung-Ausgabe) als Prinzip der Verarbeitung von Daten und Grundlage der „Von-Neumann-Architektur“	

Unterrichtsvorhaben EF-II

Thema: Geschichte der digitalen Datenverarbeitung und die Grundlagen des Datenschutzes

Leitfrage: *Welche Entwicklung durchlief die moderne Datenverarbeitung und welche Auswirkungen ergeben sich insbesondere hinsichtlich neuer Anforderungen an den Datenschutz daraus?*

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
<p>1. Selbstständige Erarbeitung von Themen durch die Schülerinnen und Schüler</p> <p>(a) Mögliche Themen zur Erarbeitung in Kleingruppen:</p> <ul style="list-style-type: none">• „Eine kleine Geschichte der Digitalisierung: vom Morsen zum modernen Digitalcomputer“• „Eine kleine Geschichte der Kryptographie: von Caesar zur Enigma“• „Von Nullen, Einsen und mehr: Stellenwertsysteme und wie man mit ihnen rechnet“• „Kodieren von Texten und Bildern: ASCII, RGB und mehr“• „Auswirkungen der Digitalisierung: Veränderungen der Arbeitswelt und Datenschutz“• Greenfoot-Projekt „Wombats“	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none">• bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A),• erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (A),• stellen ganze Zahlen und Zeichen in Binärcodes dar (D),• interpretieren Binärcodes als Zahlen und Zeichen (D),• nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation. (K).
<p>2. Vertiefung des Themas Datenschutz</p> <p>(a) Erarbeitung grundlegender Begriffe des Datenschutzes</p> <p>(b) Problematisierung und Anknüpfung an die Lebenswelt der Schülerinnen und Schüler</p> <p>(c) Diskussion und Bewertung von Fallbeispielen aus dem Themenbereich „Datenschutz“</p>	

Unterrichtsvorhaben EF-III

Thema: Grundlagen der objektorientierten Analyse, Modellierung und Implementierung

Leitfrage: *Wie lassen sich Gegenstandsbereiche informatisch modellieren und im Sinne einer Simulation informatisch realisieren?*

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
<p>1. Identifikation von Objekten</p> <p>(a) Am Beispiel eines lebensweltnahen Beispiels (BlueJ „Figuren“ bzw. Greentfoot „Crab“) werden Objekte im Sinne der Objektorientierten Modellierung eingeführt.</p> <p>(b) Objekte werden mit Objektkarten visualisiert und mit sinnvollen Attributen und „Fähigkeiten“, d.h. Methoden versehen.</p> <p>(c) Manche Objekte sind prinzipiell typgleich und werden so zu einer Objektsorte bzw. Objektklasse zusammengefasst.</p> <p>(d) Vertiefung: Modellierung weiterer Beispiele ähnlichen Musters</p>	<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),• stellen die Kommunikation zwischen Objekten grafisch dar (M),• implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I),• stellen den Zustand eines Objekts dar (D).
<p>2. Analyse von Klassen</p> <p>(a) Objektorientierte Programmierung als modularisiertes Vorgehen (Entwicklung von Problemlösungen auf Grundlage vorhandener Klassen)</p> <p>(b) Teilanalyse der Klassen</p>	
<p>3. Implementierung</p> <p>(a) Grundaufbau einer Java-Klasse</p> <p>(b) Deklaration und Initialisierung von Objekten</p> <p>(c) Methodenaufrufe mit Parameterübergabe zur Manipulation von Objekteigenschaften</p>	

Unterrichtsvorhaben EF-IV

Thema: Grundlagen der objektorientierten Programmierung und algorithmischer Grundstrukturen in Java

Leitfragen: *Wie lassen sich Simulationen unter Berücksichtigung von äußeren Einflüssen realisieren?*

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
<p>1. Veränderung von Werten anhand einfacher Algorithmen</p> <ul style="list-style-type: none"> (a) Kontinuierliche Veränderungen mit Hilfe einer Schleife (if-Bedingung und FOR-Schleife (BlueJ Figuren: Tag/Nacht-Darstellung, Sonnenlauf) (b) Tastaturabfrage zur Realisierung einer Schleifenbedingung (c) Verschachtelte Schleifen (d) Einfache Berechnungen mit Hilfsvariablen (e) Abfangen von Fehlern und Verzweigungen (IF-Anweisungen) 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • analysieren und erläutern einfache Algorithmen und Programme (A), • entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M), • 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), • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M), • ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M), • modifizieren einfache Algorithmen und Programme (I), • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I), • implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I), • implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I), • testen Programme schrittweise anhand von Beispielen (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I).
<p>2. Erstellen und Verwalten größerer Mengen einfacher Objekte</p> <ul style="list-style-type: none"> (a) Erzeugung von Objekten mit Hilfe von Zählschleifen (FOR-Schleife) (b) Vertiefung: Verschiedene Graphische Objekte 	
<p>3. Modellierung</p> <ul style="list-style-type: none"> (a) Modellierung eines Simulationsprogramms mit eigenen Klassen, mit Hilfe eines Implementationsdiagramms (b) Implementierung eigener Methoden mit und ohne Parameterübergabe (c) Realisierung von Zustandsvariablen (d) Thematisierung des Geheimnisprinzips und des Autonomitätsprinzips von Objekten (e) Aufruf von selbstimplementierten Methoden (f) Vertiefung: Weitere Projekte (Adressverwaltung, Kontoführung) 	

Unterrichtsvorhaben EF-V

Thema: Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand komplexerer Anwendungen

Leitfrage: *Wie lassen sich komplexere Datenflüsse und Beziehungen zwischen Objekten und Klassen realisieren?*

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
<p>1. Entwicklung eines komplexeren Projektes</p> <ul style="list-style-type: none"> (a) Modellierung eines Projektes („Taschenrechner“) (incl. Nutzung eines Hilfsprogramms wie „jvicer“) und Erstellung eines Implementationsdiagramms (b) Dokumentation der Klassen des Projekts (c) Implementierung eines Grundmodells (d) Ergänzung durch zusätzliche Funktionen in Diagramm, Dokumentation und Quellcode (e) Vertiefung: Entwicklung spezieller Funktionen bzw. Projekte mit vergleichbarer Grundmodellierung 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • analysieren und erläutern eine objektorientierte Modellierung (A), • stellen die Kommunikation zwischen Objekten grafisch dar (M), • 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), • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M), • ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M),
<p>3. Vererbung von Methoden</p> <ul style="list-style-type: none"> (a) Analyse und Erläuterung einer Basisversion einer Klasse (b) Prinzip der Vererbung an Beispielen (z.B. wombats: Vererbung von der Klasse Actor) (c) Reflexion des Prinzips der Vererbung (d) Vertiefung: Entwicklung eines vergleichbaren Projekts 	<ul style="list-style-type: none"> • modellieren Klassen unter Verwendung von Vererbung (M), • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I), • testen Programme schrittweise anhand von Beispielen (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I), • modifizieren einfache Algorithmen und Programme (I), • stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D), • dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D).

Unterrichtsvorhaben EF-VI

Thema: Such- und Sortieralgorithmen anhand kontextbezogener Beispiele

Leitfragen: *Wie können Objekte bzw. Daten effizient sortiert werden, so dass eine schnelle Suche möglich wird?*

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
<p>1. Experimentelle Erarbeitung eines Sortierverfahrens</p> <ul style="list-style-type: none">(a) Sortierprobleme im Kontext informatischer Systeme und im Alltag (z.B. Dateisortierung, Tabellenkalkulation, Telefonbuch, Bundesligatabelle, usw.)(b) Vergleich zweier Elemente als Grundlage eines Sortieralgorithmus(c) Erarbeitung eines Sortieralgorithmus durch die Schülerinnen und Schüler	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none">• beurteilen die Effizienz von Algorithmen am Beispiel von Sortierverfahren hinsichtlich Zeit und Speicherplatzbedarf (A),• entwerfen einen weiteren Algorithmus zum Sortieren (M),• analysieren Such- und Sortieralgorithmen und wenden sie auf Beispiele an (D).
<p>2. Systematisierung von Algorithmen und Effizienzbetrachtungen</p> <ul style="list-style-type: none">(a) Formulierung (falls selbst gefunden) oder Erläuterung von mehreren Algorithmen im Pseudocode (auf jeden Fall: Sortieren durch Vertauschen, Sortieren durch Auswählen)(b) Anwendung von Sortieralgorithmen auf verschiedene Beispiele(c) Bewertung von Algorithmen anhand der Anzahl der nötigen Vergleiche(d) Variante des Sortierens durch Auswählen (Nutzung eines einzigen oder zweier Felder bzw. lediglich eines einzigen zusätzlichen Ablageplatzes oder mehrerer neuer Ablageplätze)(e) Effizienzbetrachtungen an einem konkreten Beispiel bezüglich der Rechenzeit und des Speicherplatzbedarfs(f) Analyse des weiteren Sortieralgorithmus (sofern nicht in Sequenz 1 und 2 bereits geschehen)	

Qualifikationsphase (Q1 und Q2) - GRUNDKURS

Qualifikationsphase 1	
<i>Unterrichtsvorhaben Q1-I</i>	<i>Unterrichtsvorhaben Q1-II</i>
<p>Thema: <i>Endliche Automaten und formale Sprachen</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> • Argumentieren • Modellieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> • Endliche Automaten und formale Sprachen <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Endliche Automaten • Grammatiken regulärer Sprachen • Möglichkeiten und Grenzen von Automaten und formalen Sprachen 	<p>Thema: <i>Wiederholung der objektorientierten Modellierung und Programmierung anhand einer kontextbezogenen Problemstellung</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> • Argumentieren • Modellieren • Implementieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> • Daten und ihre Strukturierung • Algorithmen • Formale Sprachen und Automaten • Informatiksysteme <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Objekte und Klassen • Analyse, Entwurf und Implementierung von Algorithmen • Syntax und Semantik einer Programmiersprache • Nutzung von Informatiksystemen

Qualifikationsphase 1

Unterrichtsvorhaben Q1-III

Thema:

Modellierung und Implementierung von Anwendungen mit dynamischen, linearen Datenstrukturen

Zentrale Kompetenzen:

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

Inhaltsfelder:

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

Inhaltliche Schwerpunkte:

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

Unterrichtsvorhaben Q1-IV

Thema:

Suchen und Sortieren auf linearen Datenstrukturen

Zentrale Kompetenzen:

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

Inhaltsfelder:

- Algorithmen
- Formale Sprachen und Automaten

Inhaltliche Schwerpunkte:

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

Qualifikationsphase 2

Unterrichtsvorhaben Q2-I

Thema:

Maschinennahe Programmierung am Beispiel HEXTOY

Zentrale Kompetenzen:

- Argumentieren
- Darstellen und Interpretieren
- Kommunizieren und Kooperieren

Inhaltsfelder:

- Zahlensysteme
- Maschinencode
- Informatik und Wirtschaft

Inhaltliche Schwerpunkte:

- Binär- und Hexadezimalsystem
- Nutzung der Assemblerprogrammierung
- Wirkungen der Automatisierung

Unterrichtsvorhaben Q2-II

Thema:

Modellierung und Implementierung von Anwendungen mit dynamischen, nichtlinearen Datenstrukturen

Zentrale Kompetenzen:

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

Inhaltsfelder:

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

Inhaltliche Schwerpunkte:

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

Qualifikationsphase 2

Unterrichtsvorhaben Q2-III

Thema:

*Prinzipieller Informationsaustausch zwischen unterschiedlichen
Programmmodulen*

Zentrale Kompetenzen:

- Argumentieren
- Kommunizieren und Kooperieren

Inhaltsfelder:

- Informatikmodule (Erkennende Automaten, String-Operationen)
- Informatik, Mensch und Gesellschaft, mathematische Grundlagen

Inhaltliche Schwerpunkte:

- String- und Termanalyse
- Komplexe Termauswertung

Konkretisierte Unterrichtsvorhaben

Im Folgenden sollen die zuvor aufgeführten Unterrichtsvorhaben konkretisiert werden.

Qualifikationsphase

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

Die Schülerinnen und Schüler

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

Unterrichtsvorhaben Q1-I:

Thema: Endliche Automaten und formale Sprachen

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

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
<p>1. Endliche Automaten</p> <p>(a) Vom Automaten in den Schülerinnen und Schülern bekannten Kontexten zur formalen Beschreibung eines endlichen Automaten</p> <p>(b) Untersuchung, Darstellung und Entwicklung endlicher Automaten</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • analysieren und erläutern die Eigenschaften endlicher Automaten einschließlich ih-res 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).
<p>2. Untersuchung und Entwicklung von Grammatiken regulärer Sprachen</p> <p>(a) Erarbeitung der formalen Darstellung regulärer Grammatiken</p> <p>(b) Untersuchung, Modifikation und Entwicklung von Grammatiken</p> <p>(c) Entwicklung von endlichen Automaten zum Erkennen regulärer Sprachen die durch Grammatiken gegeben werden</p> <p>(d) Entwicklung regulärer Grammatiken zu endlichen Automaten</p>	
<p>3. Grenzen endlicher Automaten</p>	

Unterrichtsvorhaben Q1-II:

Thema: Wiederholung der objektorientierten Modellierung und Programmierung

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

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
<p>1. Wiederholung und Erweiterung der objektorientierten Modellierung und Programmierung durch Analyse und Erweiterung eines kontextbezogenen Beispiels</p> <ul style="list-style-type: none">(a) Analyse der Problemstellung(b) Analyse der Modellierung (Implementationsdiagramm)(c) Erweiterung der Modellierung im Implementationsdiagramm (Vererbung, abstrakte Klasse)(d) Kommunikation zwischen mindestens zwei Objekten (grafische Darstellung)(e) Dokumentation von Klassen(f) Implementierung der Anwendung oder von Teilen der Anwendung	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none">• analysieren und erläutern objektorientierte Modellierungen (A),• beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),• modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M),• ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M),• modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M),• implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I),• nutzen die Syntax und Semantik einer Programmiersprache bei der 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).

Unterrichtsvorhaben Q1-III:

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

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

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
<p>1. Die Datenstruktur Schlange im Anwendungskontext unter Nutzung der Klasse Queue</p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <p>(b) Erarbeitung der Funktionalität der Klasse Queue</p> <p>(c) Modellierung und Implementierung der Anwendung unter Verwendung eines oder mehrerer Objekte der Klasse Queue</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none">• erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A),• analysieren und erläutern Algorithmen und Programme (A),• beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),
<p>2. Die Datenstruktur Stapel im Anwendungskontext unter Nutzung der Klasse Stack</p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <p>(b) Erarbeitung der Funktionalität der Klasse Stack</p> <p>(c) Modellierung und Implementierung der Anwendung unter Verwendung eines oder mehrerer Objekte der Klasse Stack</p>	<ul style="list-style-type: none">• 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),
<p>3. Die Datenstruktur lineare Liste im Anwendungskontext unter Nutzung der Klasse List</p> <p>(a) Erarbeitung der Vorteile der Klasse List im Gegensatz zu den bereits bekannten linearen Strukturen</p> <p>(b) Modellierung und Implementierung einer kontextbezogenen Anwendung unter Verwendung der Klasse List.</p>	<ul style="list-style-type: none">• nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I),• interpretieren Fehlermeldungen und korrigieren den Quellcode (I),• testen Programme systematisch anhand von Beispielen (I),• stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D).
<p>4. Vertiefung - Anwendungen von Listen, Stapeln oder Schlangen in mindestens einem weiteren Kontext</p>	

Unterrichtsvorhaben Q1-IV:

Thema: Suchen und Sortieren auf linearen Datenstrukturen

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

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
<p>1. Suchen von Daten in Listen und Arrays</p> <ul style="list-style-type: none"> (a) Lineare Suche in Listen und in Arrays (b) Binäre Suche in Arrays als Beispiel für rekursives Problemlösen (c) Untersuchung der beiden Suchverfahren hinsichtlich ihrer Effizienz (Laufzeitverhalten, Speicherbedarf) 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • analysieren und erläutern Algorithmen und Programme (A), • beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), • beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (A), • entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“ (M), • modifizieren Algorithmen und Programme (I), • implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), • implementieren und erläutern iterative und rekursive Such- und Sortierverfahren (I), • nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I), • testen Programme systematisch anhand von Beispielen (I), • stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D).
<p>2. Sortieren in Listen und Arrays - Entwicklung und Implementierung von iterativen und rekursiven Sortierverfahren</p> <ul style="list-style-type: none"> (a) Entwicklung und Implementierung eines einfachen Sortierverfahrens für eine Liste (b) Implementierung eines einfachen Sortierverfahrens für ein Feld (c) Entwicklung eines rekursiven Sortierverfahrens für ein Feld (z.B. Sortieren durch Mischen) 	
<p>3. Untersuchung der Effizienz der Sortierverfahren „Sortieren durch direktes Einfügen“ und „Quicksort“ auf linearen Listen</p> <ul style="list-style-type: none"> (a) Grafische Veranschaulichung der Sortierverfahren (b) Untersuchung der Anzahl der Vergleichsoperationen und des Speicherbedarf bei beiden Sortierverfahren (c) Beurteilung der Effizienz der beiden Sortierverfahren 	

Unterrichtsvorhaben Q2-I:

Thema: Prinzipielle Arbeitsweise eines Computers und Grenzen der Automatisierbarkeit,
Maschinennahe Programmierung am Beispiel HEXTOY

Leitfragen: *Was sind die strukturellen Hauptbestandteile eines Computers und wie kann man sich die Ausführung eines maschinennahen Programms mit diesen Komponenten vorstellen?*

Wie unterscheiden sich Programmstrukturen in höheren und niederen Programmiersprachen?

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
<p>1. Von-Neumann-Architektur und die Ausführung maschinennaher Programme</p> <p>a) prinzipieller Aufbau einer von Neumann-Architektur mit CPU, Rechenwerk, Steuerwerk, Register und Hauptspeicher, Aufbau des HEXTOY-Fensters</p> <p>b) Vertiefung des Umgangs mit dem Hexadezimalsystem</p> <p>c) einige maschinennahe Befehle und ihre Repräsentation in einem Binär-Code, der in einem Register gespeichert werden kann</p> <p>d) Analyse und Erläuterung der Funktionsweise einfacher maschinennaher Programme</p> <p>2. Programmierung unter Assembler anhand unterschiedlicher Beispiele</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none">• erläutern die Ausführung eines einfachen maschinennahen Programms sowie die Datenspeicherung auf einer „Von-Neumann-Architektur“ (A),• untersuchen und beurteilen Grenzen des Problemlösens mit Informatiksystemen (A).• beschreiben und erläutern die fundamentalen Befehle im Vergleich zu höheren Programmiersprachen (A),• untersuchen und bewerten Programmstrukturen anhand von Beispielen (A),• nutzen Programmregister und Datenspeicher (D)• nutzen das Eingabe- und Ausgabeband für die Datenkommunikation (K).

Unterrichtsvorhaben Q2-II:

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

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

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
<p>1. Analyse von Baumstrukturen in verschiedenen Kontexten</p> <p>(a) Grundlegende Begriffe (Grad, Tiefe, Höhe, Blatt, Inhalt, Teilbaum, Ebene, Vollständigkeit)</p> <p>(b) Aufbau und Darstellung von binären Bäumen anhand von Baumstrukturen in verschiedenen Kontexten</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A), • analysieren und erläutern Algorithmen und Programme (A), • beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), • ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M), • modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M), • verwenden bei der Modellierung geeigneter Problemstellungen die Möglichkeiten der Polymorphie (M), • entwickeln iterative und rekursive Algorithmen unter Nutzung der Konstruktionsstrategien „Modularisierung“ und „Teilen und Herrschen“ (M), • implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), • modifizieren Algorithmen und 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),
<p>2. Die Datenstruktur Binärbaum im Anwendungskontext unter Nutzung der Klasse BinaryTree</p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen im Anwendungskontext</p> <p>(b) Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramms</p> <p>(c) Erarbeitung der Klasse BinaryTree und beispielhafte Anwendung der Operationen</p> <p>(d) Implementierung der Anwendung oder von Teilen der Anwendung</p> <p>(e) Traversierung eines Binärbaums im Pre-, In- und Postorder- durchlauf</p>	
<p>3. Die Datenstruktur binärer Suchbaum im Anwendungskontext unter Verwendung der Klasse BinarySearchTree</p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <p>(b) Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramms,</p>	

grafische Darstellung eines binären Suchbaums und Erarbeitung der Struktureigenschaften

- stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D).

4. Übung und Vertiefungen der Verwendung von Binärbäumen oder binären Suchbäumen anhand einer weiteren Problemstellungen (Huffman-Codierung)

Unterrichtsvorhaben Q2-III:

Thema: *Prinzipieller Informationsaustausch zwischen unterschiedlichen Programmmodulen*

Leitfragen: *Wie gestalten sich Schnittstellen zwischen unterschiedlichen Programmmodulen? Wie wird die Interaktion zwischen verschiedenen Programmen hergestellt und synchronisiert?*

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
<p>1. Syntaktische Überprüfung komplexer Terme mit Hilfe von Erkennenden Automaten</p> <ul style="list-style-type: none">a) prinzipieller Aufbau von mathematischen Termen ohne Klammerb) Erweiterung der Syntaxprüfung um eine Klammerebenec) Realisierung der Syntaxprüfung mit beliebig vielen Klammerebenen	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none">• erläutern die Systematik und Symbolik des Erkennenden Automaten (A)• analysieren die Struktur von Eingabestrings• erstellen ein Zustandsdiagramm zur Stringanalyse (M)• definieren eine Schnittstelle zum Rechenmodul (I)• entwickeln rekursive Algorithmen zur Berechnung der Terme (M)• interpretieren Endzustände als Fehlercodes (A)• testen Programme und Schnittstellen systematisch anhand von Beispielen (I)
<p>2. Programmtechnische Umsetzung mathematisch-hierarchischer Gesetze</p> <ul style="list-style-type: none">a) Programmierung des Rechengesetzes „Punktrechnung vor Strichrechnung“b) Implementierung einer Klammerebenec) Ergänzung beliebig vieler Klammerebenen	

Grundsätze der Leistungsbewertung und Leistungsrückmeldung

Auf der Grundlage von §13 - §16 der APO-GOST hat die Fachkonferenz der Gesamtschule Marienheide 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.

Beurteilungsbereich Klausuren

Verbindliche Absprachen:

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 im ersten Halbjahr, 2 Klausuren im zweiten Halbjahr
Dauer der Klausur: 2 Unterrichtsstunden
- Grundkurse Q 1: 2 Klausuren je Halbjahr
Dauer der Klausuren: 3 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 zu beachten.

Beurteilungsbereich Sonstige Mitarbeit

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

Leistungsaspekte

Mündliche 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

Sonstige schriftliche Leistungen

- Lernerfolgsüberprüfung durch kurze schriftliche Übungen
- Bearbeitung von schriftlichen Aufgaben im Unterricht

Kriterien

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

Die Bewertungskriterien stützen sich auf

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

Besonderes Augenmerk ist dabei auf

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

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

Bei Gruppenarbeiten auch auf

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

Bei Projektarbeit darüber hinaus auf

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