



H a n n a h - A r e n d t - G y m n a s i u m

49525 Lengerich, Bahnhofstraße 110

Schulinterner Lehrplan

Informatik

Jahrgangsstufe EF

Stand: August 2014

Internes Curriculum Informatik-Grundkurs Sekundarstufe II

Einführungsphase

Grundzüge:

Die Inhalte werden weitgehend aus objektorientierter Sicht betrachtet.

Ziel ist dabei die Beherrschung der objektorientierten Techniken Analyse, Design und Programmierung.

Es wird projektorientiert gearbeitet.

Als didaktisches Konzept wird „Stifte und Mäuse“ verwendet.

Programmierwerkzeug ist zurzeit BlueJ mit der Programmiersprache Java. Andere Werkzeuge sind denkbar.

Zentrale Kompetenzen:

In jedem Unterrichtsvorhaben kommen folgende Kompetenzbereiche vor:

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

Jahrgangsstufe EF:

Thema	Inhaltsfelder	Inhaltliche Schwerpunkte
E-I Einführung in die Nutzung von Informatiksystemen und in grundlegende Begrifflichkeiten Zeitbedarf: 6 Stunden	<ul style="list-style-type: none">• Informatiksysteme• Informatik, Mensch und Gesellschaft	<ul style="list-style-type: none">• Einzelrechner• Dateisystem• Einsatz von Informatiksystemen
E-II Grundlagen der objektorientierten Programmierung und der Algorithmik anhand von grafischen Beispielen Zeitbedarf: 18 Stunden	<ul style="list-style-type: none">• Daten und ihre Strukturierung• Algorithmen• Formale Sprachen und Automaten	<ul style="list-style-type: none">• Objekte und Klassen• Analyse, Entwurf und Implementierung einfacher Algorithmen• Syntax und Semantik einer Programmiersprache
E-III Modellierung und Implementierung von Klassen sowie Klassen- und Objektbeziehungen anhand grafischer Beispiele Zeitbedarf: 25 Stunden	<ul style="list-style-type: none">• Daten und ihre Strukturierung• Algorithmen• Formale Sprachen und Automaten	<ul style="list-style-type: none">• Objekte und Klassen• Analyse, Entwurf und Implementierung einfacher Algorithmen• Syntax und Semantik einer Programmiersprache
E-IV Vorbereitung von dynamischen Datenstrukturen und Verwendung abstrakter Klassen Zeitbedarf: 16 Stunden	<ul style="list-style-type: none">• Daten und ihre Strukturierung• Algorithmen• Formale Sprachen und Automaten	<ul style="list-style-type: none">• Objekte und Klassen• Analyse, Entwurf und Implementierung einfacher Algorithmen• Syntax und Semantik einer Programmiersprache
E-V Analyse von Such- und Sortieralgorithmen und Auswirkungen von Informatiksystemen auf die Gesellschaft	<ul style="list-style-type: none">• Daten und ihre Strukturierung• Algorithmen• Formale Sprachen und Automaten	<ul style="list-style-type: none">• Algorithmen zum Suchen und Sortieren• Digitalisierung von Zeichen• Wirkungen der Automatisierung

Thema	Inhaltsfelder	Inhaltliche Schwerpunkte
Zeitbedarf: 12 Stunden	<ul style="list-style-type: none"><li data-bbox="815 199 1122 231">• Informatiksysteme<li data-bbox="815 252 1384 284">• Informatik, Mensch und Gesellschaft	<ul style="list-style-type: none"><li data-bbox="1464 199 2002 268">• Geschichte der automatischen Datenverarbeitung

E-I: Einführung in die Nutzung von Informatiksystemen und in grundlegende Begrifflichkeiten

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele und Materialien
<p>1. Speicherung von Informationen</p> <ul style="list-style-type: none"> • Informatik als Wissenschaft der Verarbeitung von Informationen • Speichern von Daten mit informatischen System am Beispiel der Schulrechner • Vorgehensweise bei der Installation von im Unterricht benötigten Programmen auf dem eigenen Rechner 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D). 	
<p>2. Informations- und Datenübermittlung in Netzen</p> <ul style="list-style-type: none"> • Informatische Kommunikation in Rechnernetzen (lokal) am Beispiel des Schulnetzwerks (Benutzer, Netzwerkordner, Zugriffsrechte) • Informatische Kommunikation in Rechnernetzen (global) am Beispiel einer Internet-Kommunikationsplattform (Benutzer, logische Struktur der Plattform) 	<ul style="list-style-type: none"> • nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung und gemeinsamen Verwendung von Daten unter Berücksichtigung der Rechteverwaltung (K). • nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K). 	<p>LogoDidact</p> <p>Moodle</p>
<p>3. Aufbau informatischer Systeme</p> <ul style="list-style-type: none"> • Komponenten informatischer Systeme, von-Neumann-Architektur und EVA-Prinzip 	<ul style="list-style-type: none"> • beschreiben und erläutern den strukturellen Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der „Von-Neumann-Architektur“ (A). 	<p>Recherche im Internet</p> <p>Kurzvortrag</p>

E-II: Grundlagen der objektorientierten Programmierung und der Algorithmik anhand von grafischen Beispielen

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele und Materialien
<p>1. Einführung in die Objektorientierung, geeignete Programmierumgebungen und Nutzen von Klassenbibliotheken</p> <ul style="list-style-type: none"> • Vorstellung der Programmierumgebung EOS • Grundbegriffe der Objektorientierung (Klasse, Objekt, Attribut, Methode) • Elementare Algorithmik • Vorstellung der Programmierumgebung BlueJ und der Programmiersprache Java • Vertiefung der Objektorientierung (Klasse, Objekt, Attribut, Methode, Auftrag, Anfrage) • Nutzung von Klassenbibliotheken und Verwendung von Dokumentationen 	<ul style="list-style-type: none"> • analysieren und erläutern eine objektorientierte Modellierung (A). • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I). • implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I). • interpretieren Fehlermeldungen und korrigieren den Quellcode (I). 	<p>Programmgesteuertes Erzeugen, Manipulieren und Animieren von Objekten in EOS</p> <p>Übertragung der Konzepte nach BlueJ und Java</p> <p>SuM-Klassen Stift, Bildschirm mit Dokumentationen</p> <p>Nachrichten an Objekte Direkteingabe von Java-Befehlen</p> <p>Erste Programme als Nachrichtenfolge</p>
<p>2. Einfache Algorithmen, Verwendung von Kontrollstrukturen</p> <ul style="list-style-type: none"> • Implementierung einfacher Algorithmen • Verwendung und Dokumentation von Kontrollstrukturen • Verwendung weiterer Klassen • Der Tastaturpuffer als Beispiel einer Schlange 	<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). • implementieren Algorithmen unter Verwendung von Kontrollstrukturen sowie Methodenaufrufen (I) 	<p>Projekt: Freihandzeichnen</p> <p>Schleifen, Verzweigung</p> <p>Struktogramme</p> <p>SuM: Maus, Tastatur</p>

3. Sicherung und Anwendung von einfachen Algorithmen und Kontrollstrukturen

- Eigenständige Anwendung des gelernten Wissens auf ein neues Projekt
- Verwendung weiterer Klassen
- Beobachtung einer Vererbung
- UML-Diagramm der bisherigen Klassen ohne Sichtbarkeiten

- entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M).
- implementieren Algorithmen unter Verwendung von Kontrollstrukturen sowie Methodenaufrufen (I)
- stellen Klassen- und Vererbungsbeziehungen in Diagrammen grafisch dar (D).

Projekt: Pfeilwurf

SuM: Buntstift

Projekt: Schatzsuche

Zufallszahlen, Abstandsberechnung

SuM: Rechner (Uhr)

E-III: Modellierung und Implementierung von Klassen sowie Klassen- und Objektbeziehungen anhand grafischer Beispiele

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele und Materialien
<p>1. Modellierung von Unterklassen</p> <ul style="list-style-type: none"> • Modellierung benötigter Klassen aufgrund einer Aufgabenbeschreibung • Erstellung der Dokumentation der Klassen 	<ul style="list-style-type: none"> • ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen (M). • modellieren Klassen unter Verwendung von Vererbung (M). • dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D). 	<p>Projekt: Mini-CAD</p>
<p>2. Entwurf eigener Klassen</p> <ul style="list-style-type: none"> • Entwicklung eines UML-Diagramms zum gewählten Projekt • Parameterkonzept • Einführung private und öffentlicher Methoden • Hole- und Setze-Methoden für Attribute • Assoziation von Klassen am Beispiel der Hat-Beziehung und ihre grafische Darstellung 	<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). • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen zu (M). • ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M). • stellen Klassen und Assoziationsbeziehungen in Diagrammen grafisch dar (D). • stellen die Kommunikation von Objekten grafisch dar (M). 	<p>Projekt: Billard</p> <p>UML-Editor für BlueJ, UMLEd</p>

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele und Materialien
<p>3. Implementation eigener Klassen</p> <ul style="list-style-type: none"> • Implementation der Klassen aus der zweiten Sequenz • Erweiterung um weitere Funktionalitäten • Erzeugen vieler Exemplare von einer Klasse 	<ul style="list-style-type: none"> • 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). • interpretieren Fehlermeldungen und korrigieren den Quellcode (I). 	<p>Projekt: Fortführung Billard</p> <p>Fehlermeldungen, Ursachen und mögliche Lösungen auf Moodle sammeln</p>
<p>4. Erweiterung des Modells</p> <ul style="list-style-type: none"> • Ergänzung um eine weitere Klasse, so dass Kennt-Beziehungen notwendig werden • Entwurf und Implementation spezieller Unterklassen (Spezialisierung) 	<ul style="list-style-type: none"> • stellen Klassen und Assoziationsbeziehungen in Diagrammen grafisch dar (D). • modellieren Klassen unter Verwendung von Vererbung (M). • modifizieren einfache Algorithmen und Programme (I). 	<p>Projekt: Fortführung Billard</p> <p>Tisch</p> <p>Drallkugel, Bremskugel, Nummernkugel, ...</p>
<p>5. Arbeitsteiliger Entwurf einer komplexeren Modellierung</p> <ul style="list-style-type: none"> • Entwicklung einzelner Klassendokumentationen in Kleingruppen • Zusammenstellung aller entworfenen Klassen zu einem Gesamtmodell als Grundlage für die Implementation • Entwurf einer abstrakten Oberklasse für Klassen mit gemeinsamen Eigenschaften und Methoden - Vererbung durch Generalisierung 	<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 unter Verwendung von Vererbung (M). 	<p>Projekt: Zeichenprogramm</p> <p>Beim Entwurf der Klassen darauf achten, dass Polymorphie und späte Bindung möglich sind</p>

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele und Materialien
<p>7. Implementation in arbeitsteiligen Kleingruppen</p> <ul style="list-style-type: none"> • Entwicklung der Dokumentation der entworfenen Klassen • Implementation der Klassen • Genauere Projektplanung (Arbeitsplan) • Polymorphie bei den Unterklassen der abstrakten Oberklasse 	<ul style="list-style-type: none"> • testen Programme schrittweise anhand von Beispielen (I). 	
<p>8. Späte Bindung</p> <ul style="list-style-type: none"> • Erzeugen von beliebigen unterschiedlichen Objekten während der Laufzeit • Einsatz des Debuggers zur Veranschaulichung der Laufzeit-Situation 	<ul style="list-style-type: none"> • stellen den Zustand eines Objekts dar (D). 	Debugger

E-IV: Vorbereitung von dynamischen Datenstrukturen und Verwendung abstrakter Klassen

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele und Materialien
<p>1. Modellierung der Ausgangssituation</p> <ul style="list-style-type: none"> Projekt mit Objekten vieler gemeinsamer Eigenschaften, die zu einer Modellierung einer abstrakten Oberklasse mit mehreren Unterklassen führen. Diese Objekte müssen so gewählt sein, dass eine anschauliche Verkettung möglich ist. 	<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). ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M). stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D). 	<p>Projekt: Zug</p>
<p>2. Implementation der Klassen</p> <ul style="list-style-type: none"> Gemeinsame Erstellung der abstrakten Oberklasse arbeitsteilige Erstellung der Unterklassen Zusammenstellung zu einem kombinierten Projekt 	<ul style="list-style-type: none"> 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). interpretieren Fehlermeldungen und korrigieren den Quellcode (I). 	<p>Projekt: Fortführung Zug</p> <p>Waggon</p> <p>Spezielle Waggon</p> <p>Zusammenstellung der Waggon zu einem Zug</p>

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele und Materialien
<p>3. Verkettung von Objekten</p> <ul style="list-style-type: none"> • Verkettung als Möglichkeit, beliebig viele Objekte hintereinander zu hängen • Gemeinsame Ergänzung der Oberklasse um die Verkettungseigenschaft • Zusammenstellung beliebiger Objekte unter Verwendung anonymer Objekte 	<ul style="list-style-type: none"> • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M). 	<p>Statt alle Waggons einzeln zu bewegen, zieht die Lok den ersten Waggon, dieser den zweiten usw.</p>

E-V: Analyse von Such- und Sortieralgorithmen und Auswirkungen von Informatiksystemen auf die Gesellschaft

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele und Materialien
<p>1. Speicherung von Zeichen und die Datenstruktur „Feld“</p> <ul style="list-style-type: none"> • Umwandlung von ganzen Zahlen in Binärcodes und Zeichen • Das Feld als Datenstruktur mit direktem Zugriff • Umwandlungen von Zeichen in Zahlen und umgekehrt 	<ul style="list-style-type: none"> • stellen ganze Zahlen und Zeichen in Binärcodes dar (D). • interpretieren Binärcodes als Zahlen und Zeichen (D). • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M). 	<p>Projekt: Kodieren/Dekodieren</p> <p>Tabelle des ASCII-Codes in Dezimaldarstellung</p> <p>„Geheimcode“ in binärer Darstellung</p>
<p>2. Such- und Sortieralgorithmen</p> <ul style="list-style-type: none"> • Analyse fertiger Programmtexte zu Such- und Sortieralgorithmen • Entwurf eines zusätzlichen Sortieralgorithmus (Pseudocode) • Zählen von Vergleichs- und Vertauschoperationen an realisierten Sortieralgorithmen 	<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). 	<p>Verdecktes Sortieren von Karten</p> <p>Eine bestimmte Kartenfolge wird nach den Verfahren sortiert und die Operationen werden ausgezählt.</p>
<p>3. Informatiksysteme und ihre Auswirkungen auf die Gesellschaft</p> <ul style="list-style-type: none"> • Auswirkungen der Automatisierung mit historischen Beispielen 	<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). 	<p>Code knacken (Enigma, NSA)</p> <p>Bibliotheksarchive</p> <p>Volkszählungen</p> <p>Buchausleihe</p> <p>ggf. Referate</p>