

Schulcurriculum Informatik

EF

Stand 8.9.2022

Inhalt		Seite
1	Die Fachgruppe Informatik am Mauritius-Gymnasium	2
2	Entscheidungen zum Unterricht	3
2.1	Unterrichtsvorhaben	3
2.1.1	<i>Übersichtsraster Unterrichtsvorhaben</i>	5
2.1.2	<i>Konkretisierte Unterrichtsvorhaben</i>	8
2.2	Grundsätze der fachmethodischen und fachdidaktischen Arbeit	18
2.3	Grundsätze der Leistungsbewertung und Leistungsrückmeldung	19
2.4	Lehr- und Lernmittel	19
3	Entscheidungen zur schulübergreifenden Kooperation	19

1 Die Fachgruppe Informatik am Mauritius-Gymnasium

Das Mauritius-Gymnasium ist ein privates Gymnasium im Herzen von Büren. Die Schule ist in der Sekundarstufe I in der Regel dreizügig und wird als Halbtagsgymnasium geführt. Etwa 75% seiner Schülerinnen und Schüler kommen aus dem Umland, sind also Fahrschüler.

Die Fachgruppe Informatik besteht zurzeit aus zwei Kolleginnen.

In der Sekundarstufe II bietet das Mauritius-Gymnasium Informatik als neu einsetzendes Fach an. Es wird also nicht auf den Stoff der Mittelstufe aufgebaut. Es gibt jedes Jahr ein bis zwei Informatik-Kurse in der Einführungsphase. Dieses Curriculum für diese Kurse wurde in Zusammenarbeit mit dem benachbarten Liebfrauengymnasium entwickelt, um in der folgenden Qualifikationsphase auch Liebfrauenschülerinnen und -schüler in unserem Leistungskurs willkommen heißen zu können.

Unsere Schule hat einen Schwerpunkt „Mauritius digital“, der auf unserer Homepage ausführlich dargestellt ist.

In der Jahrgangsstufe 5 wird 2-stündig Informationstechnische Grundbildung (IG) unterrichtet. In der Stufe 6 wird zweistündig das Fach Informatik unterrichtet. In der 8ten Stufe gibt es wieder eine Stunde IG. In den Stufen 9 und 10 gibt es Informatik als Wahlmöglichkeit im Differenzierungsbereich.

Den im Schulprogramm ausgewiesenen Zielen, Schülerinnen und Schüler ihren Begabungen und Neigungen entsprechend individuell zu fördern und ihnen Orientierung für ihren weiteren Lebensweg zu bieten, fühlt sich die Fachgruppe Informatik in besonderer Weise verpflichtet.

Schülerinnen und Schüler aller Klassen- und Jahrgangsstufen werden zur Teilnahme an den Wettbewerben im Fach Informatik angehalten und, wo erforderlich, begleitet. Insbesondere gilt dies für den Informatik-Biber und den Bundeswettbewerb Informatik.

Auch die Zusammenarbeit mit der Universität Paderborn und der Fachhochschule Münster zieht sich durch die Oberstufe. In der Einführungsphase erkunden wir das HNF, Europas größtes Computermuseum, zur Geschichte der Informatik. In der Q1 nimmt der Informatik-LK am Schüler-Krypto-Tag der Uni Paderborn teil. In der Q2 besteht die Möglichkeit des Schülerstudiums an der FH Münster, um zusammen mit dem Abitur bereits den ersten Uni-Schein in der Tasche zu haben. Dazu kommen Besuche ehemaliger Schülerinnen und Schüler, sowie in Büren und Paderborn ansässiger Informatik-Firmen, um die vielfältigen Möglichkeiten einer beruflichen Laufbahn im Informatik-Bereich vorzustellen.

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 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. Um Klarheit für die Lehrkräfte herzustellen und die Übersichtlichkeit zu gewährleisten, werden in der Kategorie „Kompetenzen“ an dieser Stelle nur die übergeordneten Kompetenzerwartungen ausgewiesen, während die konkretisierten Kompetenzerwartungen erst auf der Ebene konkreter Unterrichtsvorhaben Berücksichtigung finden. Der ausgewiesene Zeitbedarf versteht sich als grobe Orientierungsgröße, die nach Bedarf über- oder unterschritten werden kann. Um Spielraum für Vertiefungen, individuelle Förderung, besondere Schülerinteressen oder aktuelle Themen 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 Kurswechslern und Lehrkraftwechseln für alle Mitglieder der Fachkonferenz Bindekraft entfalten soll, besitzt die Ausweisung „konkreter Unterrichtsvorhaben“ (Kapitel 2.1.2) empfehlenden Charakter. 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 2.4 zu entnehmen sind. Begründete 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 prozess- und inhaltsbezogenen Kompetenzen des Kernlehrplans Berücksichtigung finden. Dies ist durch entsprechende Kommunikation innerhalb der Fachkonferenz zu gewährleisten.

Konkretisierte Unterrichtsvorhaben

Dieser Lehrplan hat den Anspruch, alle im Kernlehrplan geforderten Kompetenzen abzubilden.

Die Umsetzung des Lehrplans folgt pro Unterrichtsvorhaben jeweils auf einer Übersichts- und einer dazugehörigen Konkretisierungsebene. Die Übersichtsebene ist für die Kolleginnen und Kollegen verpflichtend, die hier dargestellten Konkretisierungen beziehen sich auf das Lehrwerk **Informatik 1** und können anhand der Lehrtexte, Projekte und Aufgaben umfassend umgesetzt werden.

2.1.1 Übersichtsraster Unterrichtsvorhaben

Einführungsphase	
<p><u>Unterrichtsvorhaben E-I</u></p> <p>Thema: <i>Was macht Informatik? - Einführung in die Inhaltsfelder der Informatik</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Kommunizieren und Kooperieren - Darstellen und Interpretieren - Argumentieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Informatiksysteme - Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> - Einsatz, Nutzung und Aufbau von Informatiksystemen - Wirkung der Automatisierung <p>Zeitbedarf: 6 Stunden</p>	<p><u>Unterrichtsvorhaben E-II</u></p> <p>Thema: <i>Grundlagen der objektorientierten Analyse, Modellierung und Implementierung</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Modellieren - Implementieren - Darstellen und Interpretieren - Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Daten und ihre Strukturierung - Formale Sprachen und Automaten <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> - Objekte und Klassen - Syntax und Semantik einer Programmiersprache <p>Zeitbedarf: 8 Stunden</p>
<p><u>Unterrichtsvorhaben E-III</u></p> <p>Thema: <i>Algorithmische Grundstrukturen in Java</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Argumentieren - Modellieren - Implementieren - Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Daten und ihre Strukturierung - Algorithmen - Formale Sprachen und Automaten <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> - Objekte und Klassen - Syntax und Semantik einer Programmiersprache - Analyse, Entwurf und Implementierung einfacher Algorithmen <p>Zeitbedarf: 18 Stunden</p>	<p><u>Unterrichtsvorhaben E-IV</u></p> <p>Thema: <i>Das ist die digitale Welt! – Einführung in die Grundlagen, Anwendungsgebiete und Verarbeitung binärer Codierung</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Kommunizieren und Kooperieren - Darstellen und Interpretieren - Argumentieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Informatiksysteme - Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> - Binäre Codierung und Verarbeitung - Besondere Eigenschaften der digitalen Speicherung und Verarbeitung von Daten <p>Zeitbedarf: 8 Stunden</p>

Einführungsphase

Unterrichtsvorhaben E-V

Thema:

Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand lebensnaher Anforderungsbeispiele

Zentrale Kompetenzen:

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

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

Zeitbedarf: 18 Stunden

Unterrichtsvorhaben E-VI

Thema:

Such- und Sortieralgorithmen

Zentrale Kompetenzen:

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

Inhaltsfelder:

- Algorithmen
- Daten und ihre Strukturierung

Inhaltliche Schwerpunkte:

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

Zeitbedarf: 9 Stunden

Unterrichtsvorhaben E-VII

Thema:

Leben in der digitalen Welt – Immer mehr Möglichkeiten und immer mehr Gefahren!?

Zentrale Kompetenzen:

- Kommunizieren und Kooperieren
- Darstellen und Interpretieren
- Argumentieren

Inhaltsfelder:

- Informatiksysteme
- Informatik, Mensch und Gesellschaft

Inhaltliche Schwerpunkte:

- Geschichte der automatischen Datenverarbeitung (Exkursion zum Computermuseum HNF)
- Wirkungen der Automatisierung
- Dateisystem

Zeitbedarf: 10 Stunden

**Summe Einführungsphase:
77 Stunden**

2.1.2 Konkretisierte Unterrichtsvorhaben

Unterrichtsvorhaben EF-I

Thema: Was macht Informatik? - Einführung in die Inhaltsfelder der Informatik

Leitfragen: Was macht Informatik? Welche fundamentalen Konzepte müssen Informatikerinnen und Informatiker in ihre Arbeit einbeziehen, damit informatische Systeme effizient und zuverlässig arbeiten können? Wo lassen sich diese Konzepte (in Ansätzen) in dem schuleigenen Netzwerk- und Computersystem wiederfinden?

Vorhabenbezogene Konkretisierung:

Im ersten Unterrichtsvorhaben werden die fünf Inhaltsfelder des Faches Informatik beispielhaft an einem Informatiksystem erarbeitet. Das Unterrichtsvorhaben ist so strukturiert, dass die Schülerinnen und Schüler anhand bekannter Alltagstechnik die Grundideen fundamentaler informatischer Konzepte (Inhaltsfelder) größtenteils selbstständig erarbeiten und nachvollziehen.

Ausgehend von dem bekannten Bedienungs- und Funktionalitätswissen eines Navigationsgerätes werden die Strukturierung von Daten, das Prinzip der Algorithmik, die Eigenheit formaler Sprachen, die Kommunikationsfähigkeit von Informatiksystemen und die positiven und negativen Auswirkungen auf Mensch und Gesellschaft thematisiert. Das am Navigationsgerät erworbene Wissen kann auf weitere den Schülerinnen und Schülern bekannte Informatiksysteme übertragen werden.

In einem letzten Schritt kann ausgehend von den Inhaltsfeldern das Schulnetzwerk in Ansätzen so analysiert werden, dass ein kompetenter Umgang mit diesem ermöglicht wird.

Zeitbedarf: 6 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Informatiksysteme und ihr genereller Aufbau (a) Daten und ihre Strukturierung (b) Algorithmen (c) Formale Sprachen und Automaten (d) Informatiksysteme (e) Informatik, Mensch und Gesellschaft	Die Schülerinnen und Schüler - bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A) - nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D)	Kapitel 1 „Was macht Informatik“ Als Anschauungsmaterial bieten sich Navigationsgeräte an
2. Der kompetente Umgang mit dem Schulnetzwerk (a) Erstellen und Anlegen von Ordnerstrukturen (b) Sortieren von Dateien und Ordnern (c) Eingabe von Befehlen über Eingabeaufforderung (d) Einzelrechner und Netzwerk (e) Sicherheit und Datenschutz		Kapitel 1 „Was macht Informatik“ Interview mit dem Netzwerkadministrator, Benutzer- und Datenschutzbestimmungen der Schule

Unterrichtsvorhaben EF-II

Thema: Grundlagen der objektorientierten Analyse, Modellierung und Implementierung

Leitfragen: *Wie lassen sich Gegenstandsbereiche informatisch modellieren und in einem Greenfoot-Szenario informatisch realisieren?*

Vorhabenbezogene Konkretisierung:

Ein zentraler Bestandteil des Informatikunterrichts der Einführungsphase ist die Objektorientierte Programmierung. Dieses Unterrichtsvorhaben führt in die Grundlagen der Analyse, Modellierung und Implementierung in diesem Kontext ein.

Dazu werden zunächst konkrete Gegenstandsbereiche aus der Lebenswelt der Schülerinnen und Schüler analysiert und im Sinne des objektorientierten Paradigmas strukturiert. Dabei werden die grundlegenden Begriffe der Objektorientierung und Modellierungswerkzeuge wie Objektdiagramme und Klassendiagramme eingeführt.

Im Anschluss wird die objektorientierte Analyse für das Greenfoot-Szenario Planetenerkundung durchgeführt. Die vom Szenario vorgegebenen Klassen werden von Schülerinnen und Schülern in Teilen analysiert und entsprechende Objekte anhand einfacher Problemstellungen erprobt. Die Lernenden implementieren und testen einfache Programme. Die Greenfoot-Umgebung ermöglicht es, Beziehungen zwischen Klassen zu einem späteren Zeitpunkt (Kapitel 4) zu thematisieren. So kann der Fokus hier auf Grundlagen wie der Unterscheidung zwischen Klasse und Objekt, Attribute, Methoden, Objektidentität und Objektzustand gelegt werden.

Da in Kapitel 2 zudem auf die Verwendung von Kontrollstrukturen verzichtet wird und der Quellcode aus einer rein linearen Sequenz besteht, ist auf diese Weise eine Fokussierung auf die Grundlagen der Objektorientierung möglich, ohne dass algorithmische Probleme ablenken. Natürlich kann die Arbeit an diesen Projekten unmittelbar zum nächsten Unterrichtsvorhaben (Kapitel 3) führen. Dort stehen Kontrollstrukturen im Mittelpunkt.

Zeitbedarf: 8 Stunden

Sequenzierung des Unterrichtsvorhabens: nächste Seite

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p>1. Identifikation von Objekten und Klassen</p> <p>(a) An einem lebensweltnahen Beispiel werden Objekte und Klassen im Sinne der objektorientierten Modellierung eingeführt.</p> <p>(b) Objekte werden durch Objektdiagramme, Klassen durch Klassendiagramme dargestellt.</p> <p>(c) Die Modellierungen werden einem konkreten Anwendungsfall entsprechend angepasst.</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> - ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften und ihre Operationen (M), - stellen den Zustand eines Objekts dar (D), - modellieren Klassen mit ihren Attributen und ihren Methoden (M), - implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I), - implementieren Klassen in einer Programmiersprache, auch unter Nutzung dokumentierter Klassenbibliotheken (I). 	<p>Kapitel 2 Einführung in die Objektorientierung</p> <p>2.1 Objektorientierte Modellierung</p>
<p>2. Analyse von Objekten und Klassen im Greenfoot-Szenario</p> <p>(a) Schritte der objektorientierten Analyse, Modellierung und Implementation</p> <p>(b) Analyse und Erprobung der Objekte im Greenfoot-Szenario</p>		<p>Kapitel 2 Einführung in die Objektorientierung</p> <p>2.2 Das Greenfoot-Szenario „Planetenerkundung“</p> <p>Von der Realität zu Objekten</p> <p>Von den Objekten zu Klassen, Klassendokumentation</p> <p>Objekte inspizieren</p> <p>Methoden aufrufen</p> <p>Objektidentität und Objektzustand</p>
<p>3. Implementierung einfacher Aktionen in Greenfoot</p> <p>(a) Quelltext einer Java-Klasse</p> <p>(b) Implementation eigener Methoden, Dokumentation mit JavaDoc</p> <p>(c) Programme übersetzen (Aufgabe des Compilers) und testen</p>		<p>Kapitel 2 Einführung in die Objektorientierung</p> <p>2.3 Programmierung in Greenfoot</p> <p>Methoden schreiben</p> <p>Programme übersetzen und testen</p>

Unterrichtsvorhaben EF-III

Thema: Algorithmische Grundstrukturen in Java

Leitfragen: *Wie lassen sich Aktionen von Objekten flexibel realisieren?*

Vorhabenbezogene Konkretisierung:

Das Ziel dieses Unterrichtsvorhabens besteht darin, das Verhalten von Objekten flexibel zu programmieren. Ein erster Schwerpunkt liegt dabei auf der Erarbeitung von Kontrollstrukturen. Die Strukturen Wiederholung und bedingte Anweisung werden an einfachen Beispielen eingeführt und anschließend anhand komplexerer Problemstellungen erprobt. Da die zu entwickelnden Algorithmen zunehmend umfangreicher werden, werden systematische Vorgehensweisen zur Entwicklung von Algorithmen thematisiert.

Ein zweiter Schwerpunkt des Unterrichtsvorhabens liegt auf dem Einsatz von Variablen. Beginnend mit lokalen Variablen, die in Methoden und Zählschleifen zum Einsatz kommen, über Variablen in Form von Parametern und Rückgabewerten von Methoden, bis hin zu Variablen, die die Attribute einer Klasse realisieren, lernen die Schülerinnen und Schüler die unterschiedlichen Einsatzmöglichkeiten des Variablenkonzepts anzuwenden.

Zeitbedarf: 18 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Algorithmen (a) Wiederholungen (While-Schleife) (b) bedingte Anweisungen (c) Verknüpfung von Bedingungen durch die logischen Funktionen UND, ODER und NICHT (d) Systematisierung des Vorgehens zur Entwicklung von Algorithmen zur Lösung komplexerer Probleme	Die Schülerinnen und Schüler - analysieren und erläutern einfache Algorithmen und Programme (A), - entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M), - ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen zu (M), - modifizieren einfache Algorithmen und Programme (I), - implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I), - implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (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).	Kapitel 3 Algorithmen 3.1 Wiederholungen 3.2 Bedingte Anweisungen 3.3 Logische Funktionen 3.4 Algorithmen entwickeln
2. Variablen und Methoden (a) Implementierung eigener Methoden mit lokalen Variablen, auch zur Realisierung einer Zählschleife (b) Implementierung eigener Methoden mit Parameterübergabe und/oder Rückgabewert (c) Implementierung von Konstruktoren (d) Realisierung von Attributen		Kapitel 4 Variablen und Methoden 4.1 lokale Variablen 4.2 Methoden 4.3 Attribute

Unterrichtsvorhaben EF-IV

Thema: Das ist die digitale Welt! - Einführung in die Grundlagen, Anwendungsgebiete und Verarbeitung binärer Codierung

Leitfragen: Wie werden binäre Informationen gespeichert und wie können sie davon ausgehend weiter verarbeitet werden? Wie unterscheiden sich analoge Medien und Geräte von digitalen Medien und Geräten? Wie ist der Grundaufbau einer digitalen Rechenmaschine?

Vorhabenbezogene Konkretisierung:

Das Unterrichtsvorhaben hat die binäre Speicherung und Verarbeitung sowie deren Besonderheiten zum Inhalt.

Im ersten Schritt erarbeiten die Schülerinnen und Schüler anhand ihnen bekannter technischer Gegenstände die Gemeinsamkeiten, Unterschiede und Besonderheiten der jeweiligen analogen und digitalen Version. Nach dieser ersten grundlegenden Einordnung des digitalen Prinzips wenden die Schülerinnen und Schüler das Binäre als Zahlensystem mit arithmetischen und logischen Operationen an und codieren Zeichen binär.

Zum Abschluss soll der grundlegende Aufbau eines Rechnersystems im Sinne der von-Neumann-Architektur erarbeitet werden.

Zeitbedarf: 8 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Analoge und digitale Aufbereitung und Verarbeitung von Daten (a) Erarbeitung der Unterschiede von analog und digital (b) Zusammenfassung und Bewertung der technischen Möglichkeiten von analog und digital	Die Schülerinnen und Schüler - bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A) - stellen ganze Zahlen und Zeichen in Binärcodes dar (D), - interpretieren Binärcodes als Zahlen und Zeichen (D)	<i>Exkurs „Analog und Digital“</i>
2. Der Umgang mit binärer Codierung von Informationen (a) Das binäre (und hexadezimale) Zahlensystem (b) Binäre Informationsspeicherung (c) Binäre Verschlüsselung (d) Implementation eines Binärumrechners	- 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)	<i>Exkurs „Binäre Welt“</i>
3. Aufbau informatischer Systeme (a) Identifikation des EVA-Prinzips als grundlegende Arbeitsweise informatischer Systemen (b) Nachvollziehen der von-Neumann-Architektur als relevantes Modell der Umsetzung des EVA-Prinzips	- implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I)	<i>Exkurs „Arbeitsweise eines Computers“</i>

Unterrichtsvorhaben EF-V

Thema: Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand lebensnaher Anforderungsbeispiele

Leitfragen: Wie werden realistische Systeme anforderungsspezifisch reduziert, als Entwurf modelliert und implementiert? Wie kommunizieren Objekte und wie wird dieses dargestellt und realisiert?

Vorhabenbezogene Konkretisierung:

Das Unterrichtsvorhaben hat die Entwicklung von Objekt -und Klassenbeziehungen zum Schwerpunkt. Dazu werden, ausgehend von der Realität, über Objektidentifizierung und Entwurf bis hin zur Implementation kleine Softwareprodukte in Teilen oder ganzheitlich erstellt.

Zuerst identifizieren die Schülerinnen und Schüler Objekte und stellen diese dar. Aus diesen Objekten werden Klassen und ihre Beziehungen in Entwurfsdiagrammen erstellt.

Nach diesem ersten Modellierungsschritt werden über Klassendokumentationen und der Darstellung von Objektkommunikationen anhand von Sequenzdiagrammen Implementationsdiagramme entwickelt. Danach werden die Implementationsdiagramme unter Berücksichtigung der Klassendokumentationen in Javaklassen programmiert. In einem letzten Schritt wird das Konzept der Vererbung sowie seiner Vorteile erarbeitet.

Schließlich sind die Schülerinnen und Schüler in der Lage, eigene kleine Softwareprojekte zu entwickeln. Ausgehend von der Dekonstruktion und Erweiterung eines Spiels wird ein weiteres Projekt von Grund auf modelliert und implementiert. Dabei können arbeitsteilige Vorgehensweisen zum Einsatz kommen. In diesem Zusammenhang wird auch das Erstellen von graphischen Benutzeroberflächen eingeführt.

Zeitbedarf: 18 Stunden

Sequenzierung des Unterrichtsvorhabens: nächste Seite

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p>1. Umsetzung von Anforderungen in Entwurfsdiagramme (a) Aus Anforderungsbeschreibungen werden Objekte mit ihren Eigenschaften identifiziert (b) Gleichartige Objekte werden in Klassen (Entwurf) zusammengefasst und um Datentypen und Methoden erweitert</p>	<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), - 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), - 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). - stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D), - dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D) 	<p>Kapitel 6 Klassenentwurf 6.1. Von der Realität zum Programm 6.2. Objekte identifizieren 6.3. Klassen und Beziehungen entwerfen</p>
<p>2. Implementationsdiagramme als erster Schritt der Programmierung (a) Erweiterung des Entwurfsdiagramms um Konstruktoren und get- und set-Methoden (b) Festelegung von Datentypen in Java, sowie von Rückgaben und Parametern (c) Entwicklung von Klassendokumentationen (d) Erstellung von Sequenzdiagrammen als Vorbereitung für die Programmierung</p>	<p>- ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M),</p> <p>- modellieren Klassen unter Verwendung von Vererbung (M),</p> <p>- implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I),</p>	<p>Kapitel 6 Klassenentwurf 6.4 Klassen und Beziehungen implementieren 6.5 Vererbung</p>
<p>3. Programmierung anhand der Dokumentation und des Implementations- und Sequenzdiagrammes (a) Klassen werden in Java-Quellcode umgesetzt (b) Das Geheimnisprinzip wird umgesetzt (c) Einzelne Klassen und das Gesamtsystem werden anhand der Anforderungen und Dokumentationen auf ihre Korrektheit überprüft.</p>	<p>- testen Programme schrittweise anhand von Beispielen (I),</p> <p>- interpretieren Fehlermeldungen und korrigieren den Quellcode (I),</p> <p>- analysieren und erläutern einfache Algorithmen und Programme (A)</p> <p>- modifizieren einfache Algorithmen und Programme (I),</p> <p>- entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M).</p>	<p>Kapitel 6 Klassenentwurf 6.4 Klassen und Beziehungen implementieren 6.5 Vererbung</p>
<p>4. Vererbungsbeziehungen (a) Das Grundprinzip der Vererbung wird erarbeitet (b) Die Vorteile der Vererbungsbeziehungen (c) Vererbung wird implementiert</p>	<p>- stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D),</p> <p>- dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D)</p>	<p>Kapitel 6 Klassenentwurf 6.5 Vererbung</p>
<p>5. Softwareprojekt (a) Analyse und Dekonstruktion eines Spiels (Modelle, Quelltexte) (b) Erweiterung des Spiels um weitere Funktionalitäten (c) Modellierung eines Spiels aufgrund einer Anforderungsbeschreibung, inklusive einer grafischen Benutzeroberfläche (d) (arbeitsteilige) Implementation des Spiels</p>		<p>Kapitel 8 Softwareprojekte 8.1 Softwareentwicklung 8.2 Oberflächen</p>

Unterrichtsvorhaben EF-VI

Thema: Such- und Sortieralgorithmen

Leitfragen: *Wie können Objekte bzw. Daten effizient gesucht und sortiert werden?*

Vorhabenbezogene Konkretisierung:

Dieses Unterrichtsvorhaben beschäftigt sich mit der Erarbeitung von Such- und Sortieralgorithmen. Der Schwerpunkt des Vorhabens liegt dabei auf den Algorithmen selbst und nicht auf deren Implementierung in einer Programmiersprache.

Zunächst lernen die Schülerinnen und Schüler das Feld als eine erste Datensammlung kennen. Optional können nun zunächst die wesentlichen Eigenschaften von Algorithmen wie z.B. Korrektheit, Terminiertheit, Effizienz und Verständlichkeit sowie die Schritte einer Algorithmenentwicklung erarbeitet werden (Klärung der Anforderung, Visualisierung, Zerlegung in Teilprobleme).

Daran anschließend lernen die Schülerinnen und Schüler zunächst Strategien des Suchens (lineare Suche, binäre Suche, Hashing) und dann des Sortierens (Selection Sort, Insertion Sort, Bubble Sort) kennen. Sie lernen, diese Strategien als Struktogramme darzustellen. Die Projekteinstiege dienen dazu, die jeweiligen Strategien handlungsorientiert zu erkunden und intuitive Effizienzbetrachtungen der Suchalgorithmen vorzunehmen.

Schließlich wird die Effizienz unterschiedlicher Sortierverfahren beurteilt.

Zeitbedarf: 9 Stunden

Sequenzierung des Unterrichtsvorhabens: nächste Seite

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p>1. Modellierung und Implementation von Datenansammlungen</p> <p>(a) Modellierung von Attributen als Felder (b) Deklaration, Instanziierung und Zugriffe auf ein Feld</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> - stellen Such- und Sortieralgorithmen als Struktogramme dar (D) - analysieren Such- und Sortieralgorithmen und wenden sie auf Beispiele an (D) 	<p>Kapitel 7 Sortieren und Suchen auf Feldern</p> <p>7.1 Das Feld – Eine Sammlung von Daten</p>
<p>2. Explorative Erarbeitung von Suchverfahren</p> <p>(a) Erkundung von Strategien für das Suchen auf unsortierten Daten, auf sortierten Daten und mithilfe einer Berechnungsfunktion. (b) Vergleich der drei Verfahren durch intuitive Effizienzbetrachtungen.</p>	<ul style="list-style-type: none"> - entwerfen einen weiteren Algorithmus zum Sortieren (M) - beurteilen die Effizienz von Algorithmen am Beispiel von Sortierverfahren hinsichtlich Zeit und Speicherplatzbedarf (A) 	<p>Kapitel 7 Sortieren und Suchen auf Feldern</p> <p>Projekteinstieg 1: Suchen 7.2 Suchen mit System Lineare Suche Binäre Suche Hashing</p>
<p>3. Systematisierung von Algorithmen und Effizienzbetrachtungen</p> <p>(a) Formulierung (falls selbst gefunden) oder Erläuterung von mehreren Algorithmen im Pseudocode und als Struktogramm (b) Anwendung von Sortieralgorithmen auf verschiedene Beispiele (c) Bewertung von Algorithmen anhand der Anzahl der nötigen Vergleiche (d) Effizienzbetrachtungen an einem konkreten Beispiel bezüglich der Rechenzeit und des Speicherplatzbedarfs (e) Analyse eines weiteren Sortieralgorithmus (sofern nicht in (a) bereits geschehen)</p>	<ul style="list-style-type: none"> - ordnen Attributen lineare Datenansammlungen zu (M) 	<p>Kapitel 7 Sortieren und Suchen auf Feldern</p> <p>Projekteinstieg 2: Sortieren 7.3 Ordnung ist das halbe Leben!? – Sortieren Sortieren Selection Sort Insertion Sort Bubble Sort</p>

Unterrichtsvorhaben EF-VII

Thema: Leben in der digitalen Welt – Immer mehr Möglichkeiten und immer mehr Gefahren!?

Leitfragen: Welche Entwicklungen, Ideen und Erfindungen haben zur heutigen Informatik geführt? Welche Auswirkungen hat die Informatik für das Leben des modernen Menschen?

Vorhabenbezogene Konkretisierung:

Das Unterrichtsvorhaben stellt die verschiedenen Entwicklungsstränge der Informatik in den Fokus. Darüber hinaus wird beispielhaft analysiert und bewertet, welche Möglichkeiten und Gefahren die moderne Informationsverarbeitung mit sich bringt.

Im ersten Schritt des Unterrichtsvorhabens wird anhand von Themenkomplexen entscheidende Entwicklungen der Informatik erarbeitet. Dabei werden auch übergeordnete Tendenzen identifiziert.

Ausgehend von dieser Betrachtung kann die aktuelle Informatik hinsichtlich ihrer Leistungsfähigkeit analysiert werden. Dabei soll herausgestellt werden, welche positiven und negativen Folgen Informatiksysteme mit sich bringen können.

Zeitbedarf: 12 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Schriftzeichen, Rechenmaschine, Computer (a) Anhand von Schwerpunkten, wie z.B. Datenspeicherung, Maschinen, Vernetzung sollen wichtige Entwicklungen der Informatik vorgestellt werden. (b) Anhand der unterschiedlichen Schwerpunkte sollen universelle Tendenzen der Entwicklung der Informationsverarbeitung erarbeitet werden.	Die Schülerinnen und Schüler - bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A), - erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (A) - nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K)	Exkurs „Geschichte der Informatik“ Exkursion zum Computermuseum HNF
2. Die Informationsverarbeitung und ihre Möglichkeiten und Gefahren (a) Ausgehend von 1. werden Tendenzen der Entwicklung der Informatik erarbeitet (b) Informatik wird als Hilfswissenschaft klassifiziert, die weit über ihren originären Bereich hinaus Effizienz- und Leistungssteigerungen erzeugt (c) Anhand von Fallbeispielen werden technische und organisatorische Vorteile, sowie deren datenschutzrechtlichen Nachteile betrachtet.		Exkurs „Informatik und Gesellschaft“

2.2 Grundsätze der fachmethodischen und fachdidaktischen Arbeit

Unter Berücksichtigung des Schulprogramms hat die Fachkonferenz Informatik die folgenden fachmethodischen und fachdidaktischen Grundsätze beschlossen. In diesem Zusammenhang beziehen sich die Grundsätze 1 bis 15 auf fächerübergreifende Aspekte, die auch Gegenstand der Qualitätsanalyse sind, die Grundsätze 16 bis 26 sind fachspezifisch angelegt.

Überfachliche Grundsätze:

- 1) Geeignete Problemstellungen zeichnen die Ziele des Unterrichts vor und bestimmen die Struktur der Lernprozesse.
- 2) Inhalt und Anforderungsniveau des Unterrichts entsprechen dem Leistungsvermögen der Schüler/innen.
- 3) Die Unterrichtsgestaltung ist auf die Ziele und Inhalte abgestimmt.
- 4) Medien und Arbeitsmittel sind schülernah gewählt.
- 5) Die Schüler/innen erreichen einen Lernzuwachs.
- 6) Der Unterricht fördert eine aktive Teilnahme der Schüler/innen.
- 7) Der Unterricht fördert die Zusammenarbeit zwischen den Schülern/innen und bietet ihnen Möglichkeiten zu eigenen Lösungen.
- 8) Der Unterricht berücksichtigt die individuellen Lernwege der einzelnen Schüler/innen.
- 9) Die Schüler/innen 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.
- 15) Wertschätzende Rückmeldungen prägen die Bewertungskultur und den Umgang mit Schülerinnen und Schülern.

Fachliche Grundsätze:

- 16) Im Unterricht werden fehlerhafte Schülerbeiträge produktiv im Sinne einer Förderung des Lernfortschritts der gesamten Lerngruppe aufgenommen.
- 17) Der Unterricht ermutigt die Lernenden dazu, auch fachlich unvollständige Gedanken zu äußern und zur Diskussion zu stellen.
- 18) Die Bereitschaft zu problemlösenden Arbeiten wird durch Ermutigungen und Tipps gefördert und unterstützt.
- 19) Die Einstiege in neue Themen erfolgen grundsätzlich mithilfe sinnstiftender Kontexte, die an das Vorwissen der Lernenden anknüpfen und deren Bearbeitung sie in die dahinter stehende Informatik führt.
- 20) Es wird genügend Zeit eingeplant, in der sich die Lernenden neues Wissen aktiv konstruieren und in der sie angemessene Grundvorstellungen zu neuen Begriffen entwickeln können.
- 21) Durch regelmäßiges wiederholendes Üben werden grundlegende Fertigkeiten „wachgehalten“.
- 22) Im Unterricht werden an geeigneter Stelle differenzierende Aufgaben eingesetzt.
- 23) Die Lernenden werden zu regelmäßiger, sorgfältiger und vollständiger Dokumentation der von ihnen bearbeiteten Aufgaben angehalten.
- 24) Im Unterricht wird auf einen angemessenen Umgang mit fachsprachlichen Elementen geachtet.
- 25) Digitale Medien werden regelmäßig dort eingesetzt, wo sie dem Lernfortschritt dienen.

2.3 Grundsätze der Leistungsbewertung und Leistungsrückmeldung

siehe Leistungsmessung_Informatik_2012.docx

2.4 Lehr- und Lernmittel

Die Unterrichtsvorhaben in Informatik hängen in ihrer konkreten Umsetzung von der gewählten Programmierumgebung ab.

In diesem Lehrplan wird zur Einführung in die objektorientierte Programmierung die didaktische Lernumgebung Greenfoot gewählt. Die Lehrtexte und Aufgaben des Lehrwerks „Informatik 1“ beziehen sich auch speziell auf ein bestimmtes Greenfoot-Szenario.

Nach der Einarbeitungsphase sollte auf eine komplexere Lernumgebung umgestellt werden. Hier bieten sich verschiedene Möglichkeiten (in alphabetischer Reihenfolge) an:

- BlueJ
- Eclipse
- Java Editor
- NetBeans

Im Lehrwerk sind die Lehrtexte und Aufgaben so verfasst, dass jede der Umgebungen eingesetzt werden kann. Zur lehrplankonformen und arbeitssparenden Umsetzung von Modellierung und Programmierung empfiehlt sich der Java Editor (<http://www.javaeditor.org/doku.php>).

3 Entscheidungen zur schulübergreifenden Kooperation

In der Informatik findet eine enge Kooperation zwischen dem Mauritius-Gymnasium und benachbarten dem Liebfrauen-Gymnasium statt. Dieses Curriculum wurde gemeinsam erstellt und wird gemeinsam umgesetzt. Dies wird durch regelmäßige gemeinsame Fachkonferenzen und dem Austausch von Material sichergestellt. Um auf den gemeinsamen LK in der Q vorzubereiten, wird die letzte Klausur der EF schulübergreifend parallel geschrieben.