



{informatik}am**rsg**

Reinoldus- und Schiller-Gymnasium (RSG), Dortmund

Schulinternes Curriculum Informatik

10. Juni 2014

Version vom 09.06.2014

Inhaltsverzeichnis

1. Einleitung	4
2. Unterrichtsinhalte und Kompetenzen	5
2.1. Differenzierung Jahrgangsstufen 8 und 9 (Wahlpflichtbereich II)	5
2.1.1. Schwerpunktmäßige Zielkompetenzen der Jahrgangsstufe 8	5
2.1.2. Unterrichtsinhalte Jahrgangsstufe 8	6
2.1.3. Schwerpunktmäßige Zielkompetenzen der Jahrgangsstufe 9	7
2.1.4. Unterrichtsinhalte Jahrgangsstufe 9	7
2.1.5. Zur Rolle von Hausaufgaben für das Fach Informatik in der Sekundarstufe I	8
2.2. Jahrgangsstufe EF	9
2.2.1. Übersicht Unterrichtsvorhaben	9
2.2.2. Konkretisierung der Unterrichtsvorhaben (nicht verbindlich)	11
2.3. Jahrgangsstufe Q1	16
2.3.1. Schwerpunktmäßige Zielkompetenzen der Jahrgangsstufe Q1	16
2.3.2. Unterrichtsinhalte Grundkurs	16
2.3.3. Unterrichtsinhalte Leistungskurs	17
2.4. Jahrgangsstufe Q2	19
2.4.1. Schwerpunktmäßige Zielkompetenzen der Jahrgangsstufe Q2	19
2.4.2. Unterrichtsinhalte Grundkurs	19
2.4.3. Unterrichtsinhalte Leistungskurs	20
2.5. Zur Rolle von Hausaufgaben für das Fach Informatik in der Sekundarstufe II . .	21
3. Leistungsbewertung	22
3.1. Sekundarstufe I	22
3.1.1. Schriftliche Arbeiten	22
3.1.2. Sonstige Leistungen im Unterricht („Sonstige Mitarbeit“)	22
3.2. Sekundarstufe II	23
3.2.1. Sonstige Mitarbeit	23
3.2.2. Klausuren	23
A. Kriteriales Raster Sonstige Mitarbeit Sek. I	25
B. Kriteriales Raster Sonstige Mitarbeit Sek. II	26

1. Einleitung

Das Fach Informatik wird am RSG sowohl in der Mittelstufe (Differenzierungsbereich der Jahrgangsstufen 8 und 9) als auch im Bereich der Oberstufe (Jahrgangsstufen EF, Q1 und Q2) unterrichtet. Somit kann Informatik am RSG als Abiturfach gewählt werden - bei ausreichenden Kurswahlzahlen auch als Leistungskurs. Nach Neukonstituierung des Fachbereichs Informatik am RSG im Jahre 2009 sowie der erfolgten Einführung der neuen Kernlehrpläne für die Oberstufe im Jahre 2014 präsentiert dieses schulinterne Curriculum eine Neuausrichtung der unterrichtlichen Grundlagen an unserer Schule

Der Unterricht in Informatik orientiert sich an den fachspezifischen Vorgaben des Landes Nordrhein-Westfalen, das heißt an den Lehrplänen für die Sekundarstufe I und Sekundarstufe II. Übergeordnetes Prinzip des Faches ist am RSG die Einführung und Vertiefung der **objektorientierten** Sichtweise, eines grundlegenden und modernen Prinzips der heutigen Informatik. Daher stellt der Informatikunterricht keine reine Anwenderschulung (Textverarbeitung, Tabellenkalkulation, usw.) dar, sondern verzahnt die Heranführung an solche Software mit der Vermittlung von Wissen zur generellen Funktionsweise von Informatiksystemen. Dazu zählt auch der verantwortungsvolle Umgang mit sowie Chancen und Risiken des Einsatzes von Informationstechnologie (Stichwort: Informatik, Mensch und Gesellschaft).

Für den Differenzierungsbereich existiert derzeit nur ein sehr alter Lehrplan, der vorwiegend Unterrichts**inhalte** vorgibt (ohne Ausweisung von Kompetenzen). Hinsichtlich der Kompetenzzuordnungen orientiert sich unser Curriculum daher an den in diesem Lehrplan beschriebenen obligatorischen *Inhalten* und *Methoden* des Fachs Informatik. — Für die Sekundarstufe II orientiert sich das schulinterne Curriculum stringent an den Vorgaben des *Kernlehrplans Informatik für die gymnasiale Oberstufe* aus dem Jahre 2014. Die Arbeit im Informatikunterricht am RSG ist geprägt durch Partner-, Gruppen- und Projektarbeiten und damit primär produkt- und outputorientiert.

Schülerinnen und Schüler, die das Fach Informatik wählen möchten, sollten sich bewusst sein, dass es sich dabei weder um eine reine Anwenderschulung (s.o.), noch um einen reinen Programmierkurs handelt. Außerdem: Der mathematische Anteil ist im Fach Informatik eher **hoch**.

2. Unterrichtsinhalte und Kompetenzen

2.1. Differenzierung Jahrgangsstufen 8 und 9 (Wahlpflichtbereich II)

In Übereinstimmung mit den inhaltlichen Vorgaben des *Lehrplan Informatik für die Sekundarstufe I an Gymnasien* werden am RSG in den Jahrgangsstufe 8 und 9 die folgenden Kompetenzen vermittelt – die zugrundeliegenden Unterrichtsinhalte finden sich weiter unten. Die Reihenfolge (Sequenzierung) der Unterrichtsinhalte innerhalb der jeweiligen Jahrgangsstufe liegt im Ermessen des unterrichtenden Lehrkraft.

2.1.1. Schwerpunktmäßige Zielkompetenzen der Jahrgangsstufe 8

- Textverarbeitungssysteme:
 - Nutzung von Textverarbeitungssystemen
 - Eingabe und Änderung von Texten
 - Gestaltung von Texten
- Tabellenkalkulationssysteme:
 - Nutzung von Tabellenkalkulationssystemen
 - Entwurf und Analyse eines Rechenblatts auf Basis der Analyse eines konkreten Anwendungsproblems
 - Anwendung von Datentypen in Rechenblättern
 - Anwendung der Verknüpfung von Zellen (absolute und relative Adressierung, einfache Formeln)
- Datenbanksysteme:
 - Nutzung von Dateiverwaltungssystemen (Datenbanken)
 - Entwurf und Strukturierung von Dateien auf der Basis der Analyse eines konkreten Anwendungsproblems
 - Anwendung von Datentypen in Dateien
 - Anwendung von elementaren Operationen (Einfügen, Löschen, Ändern, Suchen, Speichern, Drucken)
 - Sortieren von Datensätzen
 - Auswählen von Daten aus einem größeren Datenbestand (Selektion)
 - Ausblenden von Informationen (Projektion)
 - Kenntnis zentraler Aspekte des Datenschutzes
- Programmiersystem für Vektorgrafiken:
 - Nutzung von Systemen zur graphischen Darstellung
- Grundlagen Objektorientierung
 - Anwendung von objektorientierter Modellierung (als zeitgemäßer Ersatz für die im Lehrplan geforderten Aspekte »Top-Down-Entwurfstechnik« und »Methode der schrittweisen Verfeinerung (Strukturierung und Zerlegung in Module)«)
 - Abgrenzung von Anwender- und Programmiersystemen
 - Strukturierung von Algorithmen durch Verwendung von Methoden
- Objektorientierte Programmierung
 - Lösung von einfachen algorithmischen Problemen mit elementaren Befehlen im Direktausführungsmodus

- Bearbeitung algorithmischer Problemstellungen mithilfe des jeweils eingesetzten Programmierwerkzeugs unter Verwendung von bedingten Verzweigungen und Schleifenstrukturen
- Testen von Programmteilen
- Nutzung vernetzter Informations- und Kommunikationssysteme (Schulnetzwerk, Internet)

2.1.2. Unterrichtsinhalte Jahrgangsstufe 8

1. Umgang mit Software
 - a) Textverarbeitungssystem (z.B. OpenOffice.org Writer) in objektorientierter Sichtweise
 - b) Tabellenkalkulation (z.B. Openoffice.org Calc) in objektorientierter Sichtweise und Anwendungsmöglichkeiten (übersichtliche Ausführung von Rechnungen, Lösung von mathematisch orientierten Aufgaben aus anderen Fächern, etc.)
 - c) Objekt-relacionales Datenbanksystem (z.B. Openoffice.org Base) inkl. Entity-Relationship-Modell, einfacher SQL-Abfragen und Normalisierung
 - d) Einfaches Programmiersystem zur Erstellung, Manipulation und Animation von Vektorgrafiken (z.B. Object-Draw und EOS (einfache objektorientierte Sprache))
2. Funktionsweise von Software
 - a) Grundlagen der Objektorientierung (Klasse, Objekt, Attribut, Methode) und Algorithmen
 - b) Objektorientierte Programmierung

Eine klare Aufteilung der Inhaltspunkte 1 (Umgang mit Software) und 2 (Funktionsweise von Software) nach Halbjahren der Jahrgangsstufe 8 wird am RSG nicht vorgenommen, da beide Inhalte über die objektorientierte Sichtweise auf Software integriert vermittelt werden.

2.1.3. Schwerpunktmäßige Zielkompetenzen der Jahrgangsstufe 9

- Einführung in die Robotik am Beispiel des LegoNXT-Roboters
 - Messen und Steuern bei technischen Prozessen
 - Umgang mit Interfaces: Digitale Ein- und Ausgänge, Analog-Digital-Wandler, Digital-Analog-Wandler
 - Nutzung einer Programmiersprachenerweiterung für die Interfacesteuerung
 - Umgang mit Sensoren und Aktoren
 - Regeln bei technischen Prozessen (z.B. Regelkreis (Sollwert, Istwert, Stellgröße))
- Grundlagen der technischen Informatik
 - Verständnis und Anwendung digitaler Informationsdarstellung: Bit, Byte, Codierung von Zahlen und Zeichen
 - Konstruktion elementarer logischer Schaltungen
 - Verständnis der Funktionsweise eines von-Neumann-Computers:
 - * Aufbau eines von-Neumann-Computers aus den Funktionsblöcken Rechenwerk, Steuerwerk, Speicherwerk
 - * Informationsfluss zwischen den Funktionsblöcken: Bussystem, von-Neumann-Zyklus
- Softwareprojekte
 - Anwendung von objektorientierter Modellierung (als zeitgemäßer Ersatz für die im Lehrplan genannten Aspekte »Methode der schrittweisen Verfeinerung« und »Strukturierung eines Problems in Teilprobleme, Modularisierung«)
 - Anwendung von Problemabstraktion bzw. Problemreduktion
 - Anwendung von strukturierten Datentypen
 - Zugriff auf strukturierte Daten
- Nutzung vernetzter Informations- und Kommunikationssysteme (Schulnetzwerk, Internet)

2.1.4. Unterrichtsinhalte Jahrgangsstufe 9

1. Funktionsweise von Hardware, Prozessdatenverarbeitung
 - a) Einführung in die Robotik am Beispiel des LegoNXT-Roboters
 - b) Grundlagen der technischen Informatik (digitale Schaltungen, boolesche Algebra)
2. Softwareprojekte (unter Verwendung eines didaktisch-reduzierten Programmiersystems)

2.1.5. Zur Rolle von Hausaufgaben für das Fach Informatik in der Sekundarstufe I

Das Fach Informatik ist durch die Notwendigkeit der Einübung des professionellen Umgangs mit Informatiksystemen übungintensiv. Zum Erwerb der Zielkompetenzen des Fachs reicht die bloße im Unterricht zur Verfügung stehende Zeit allerdings in der Regel aus. Eine erfolgreiche Teilnahme am Unterricht setzt jedoch eine kontinuierliche häusliche Vor- und Nachbereitung (auch in Form der Anfertigung von Hausaufgaben) voraus. Mehrfach nicht angefertigte Hausaufgaben werden dabei als Leistungsverweigerung gewertet.

Verfügt eine Schülerin/ein Schüler zuhause **nachweislich** nicht über die notwendigen technischen Voraussetzungen zur Bearbeitung gestellter Hausaufgaben, so wird ihr/ihm nach Absprache mit dem Fachlehrer im Nachmittagsbereich (oder bei Unterrichtsschluss vor der 6. Stunde auch vormittags) Zugang zu den Informatikräumen der Schule gewährt – sofern das Verhalten der Schülerin/des Schülers dies nach Ansicht der unterrichtenden Lehrkraft zulässt. Die Schülerin/Der Schüler hat dafür über das Nicht-Vorliegen entsprechender Voraussetzungen auf Verlangen des Fachlehrers eine Bescheinigung ihrer/seiner Erziehungsberechtigten vorzulegen.

2.2. Jahrgangsstufe EF

Die Jahrgangsstufe EF bereitet den Unterricht in der Qualifikationsphase inhaltlich und methodisch entsprechend der Vorgaben der Richtlinien und des Lehrplans vor. Zentrale Prinzipien des Faches werden im Anwendungskontext zur Erlangung der aufgeführten Zielkompetenzen einflührend erarbeitet. Zur Zeit orientiert sich der Unterricht inhaltlich am „E-Book“ *Informatik mit BlueJ - Ein Kurs für die Jahrgangsstufe 11* von Herrn Ulrich Helmich (Söderblom-Gymnasium, Espelkamp). Ein festes Lehrwerk existiert nicht.

Die am RSG eingesetzte Programmiersprache in der Sekundarstufe II ist Java. Als Programmierumgebung wird BlueJ verwendet.

Die Darstellung folgende der Unterrichtsvorhaben im schulinternen Lehrplan deckt sämtliche im Lehrplan Kernlehrplan für die Sekundarstufe II angeführten Kompetenzen ab. Die entsprechende Umsetzung erfolgt auf zwei Ebenen: der Übersichts- und der Konkretisierungsebene.

- In der folgenden »Übersicht Unterrichtsvorhaben« (Abschnitt 2.2.1) wird die Verteilung der Unterrichtsvorhaben dargestellt. Diese wurde von der Fachkonferenz Informatik hinsichtlich der Gegenstände als **verbindlich** festgelegt, wobei die Reihenfolge der Unterrichtsvorhaben durchaus variiert werden darf. Auch soll es der einzelnen Lehrkraft ausdrücklich gestattet sein, die Kompetenzschwerpunkte (s.u. »zentrale Kompetenzen«) anders über die Unterrichtsvorhaben zu verteilen, so lange alle im Kernlehrplan genannten Kompetenzen abgedeckt wrden.
- Die »konkretisierten Unterrichtsvorhaben« (Abschnitt 2.2.2) enthalten Beispiele und Materialien, die lediglich orientierenden Charakter haben und von jeder Lehrkraft nach eigener Vorstellung umgesetzt oder alternativ gestaltet werden können..

Da die Inhalte in den folgend genannten Unterrichtsvorhaben in der Regel anhand von kontextualisierten Problemstellungen erarbeitet werden, ergeben sich häufig Bezüge zu jeweils mehreren Inhaltsfeldern des Kernlehrplans.

2.2.1. Übersicht Unterrichtsvorhaben

Die folgende Grobübersicht der Unterrichtsvorhaben ist für die unterrichtenden Lehrkräfte verbindlich.

Unterrichtsvorhaben EF 1:

Gegenstand:

Grundlagen der objektorientierten Modellierung, Implementierung und Analyse sowie von Objektbeziehungen und eine ersten Umsetzung in UML

Zentrale Kompetenzen:

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

Zentrale Inhaltsfelder:

- Daten und ihre Strukturierung
- Algorithmen
- Formale Sprachen und Automaten (Bezug: Syntax der Programmiersprache Java)
- Informatiksysteme (Bezug: Nutzung der Entwicklungsumgebung BlueJ)

Inhaltliche Schwerpunkte:

- Objekte und Klassen
- Syntax und Semantik einer Programmiersprache
- Einzelrechner
- Dateisystem

Unterrichtsvorhaben EF 2:**Gegenstand:**

Grundlagen der objektorientierten Programmierung und algorithmischer Grundstrukturen (Fallunterscheidung und Schleifen) in Java

Zentrale Kompetenzen:

- Argumentieren
- Modellieren
- Implementieren
- Kommunizieren und Kooperieren

Zentrale Inhaltsfelder:

- Daten und ihre Strukturierung
- Algorithmen
- Formale Sprachen und Automaten (Bezug: Syntax der Programmiersprache Java)

Inhaltliche Schwerpunkte:

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

Unterrichtsvorhaben EF 3:**Gegenstand:**

Modellierung und Implementierung von Klassen- und Objektbeziehungen (Vertiefung) anhand anwendungsbezogener Projekte

Zentrale Kompetenzen:

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

Zentrale Inhaltsfelder:

- Daten und ihre Strukturierung
- Algorithmen
- Formale Sprachen und Automaten (Bezug: Syntax der Programmiersprache Java)

Inhaltliche Schwerpunkte:

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

Unterrichtsvorhaben EF 4:

Gegenstand:

Such- und Sortierverfahren anhand kontextualisierter Beispiele, verzahnt mit ihrer Implementierung auf Arrays

Zentrale Kompetenzen:

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

Zentrale Inhaltsfelder:

- Daten und ihre Strukturierung
- Algorithmen
- Formale Sprachen und Automaten (Bezug: Syntax der Programmiersprache Java)

Inhaltliche Schwerpunkte:

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

Unterrichtsvorhaben EF 5:

Gegenstand:

Entwicklung der digitalen Datenverarbeitung und Grundlagen des Datenschutzes

Zentrale Kompetenzen:

- Argumentieren
- Darstellen und Interpretieren
- Kommunizieren und Kooperieren

Zentrale Inhaltsfelder:

- Informatik, Mensch und Gesellschaft
- Informatiksysteme

Inhaltliche Schwerpunkte:

- Wirkungen der Automatisierung
- Geschichte der automatischen Datenverarbeitung
- Digitalisierung

2.2.2. Konkretisierung der Unterrichtsvorhaben (nicht verbindlich)

Die folgende Konkretisierung der Unterrichtsvorhaben soll für die unterrichtenden Lehrkräfte keine Bindungswirkung entfalten. Sie ist als Hilfestellung und Orientierung – v.a. für neue Kolleginnen und Kollegen sowie Referendarinnen und Referendare – zu betrachten.

Unterrichtsvorhaben EF 1:

Gegenstand:

Grundlagen der objektorientierten Modellierung, Implementierung und Analyse sowie von Objektbeziehungen und eine ersten Umsetzung in UML

Sequenzierung:

- Interaktive Erzeugung von Objekten aus vorgefertigten Klassen (»Figuren«-Projekt nach Barnes und Kölling), die nach Instantiierung visualisiert werden
- Veränderung von Attribute dieser Objekte über den Aufruf von Methoden, die das Erscheinungsbild (Position, Größe, etc.) der visuellen Repräsentation der Objekte beeinflussen
- Erarbeitung (anhand dieses Beispiels) eines ersten Verständnisses der Grundbegriffe der objektorientierten Modellierung: Objekte, Klassen, Attribute, verändernde und sondierende Methoden
- Gestaltung ein Bild nach Vorgabe durch Instantiierung und Methodenaufrufe wie oben beschrieben
- Erstellung von Objektkarten in UML zu den Objekten des Bildes
- ein erstes Erarbeitung eines ersten Verständnisses für Datentypen über die vorgefertigten Klassen
- Erarbeitung eines ersten Klassenbegriffs erarbeiten die Objektkarten zu ähnlichen Objekten
- Implementierung des erzeugten Bild als neue Klasse mit eigenen Attributen, Bezugsobjekten und Methoden zwecks Herstellung von Permanenz

Bezüge zu Kompetenzerwartungen des Kernlehrplans:

Die Schülerinnen und Schüler...

- ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M)
- modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M)
- 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)
- nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D)

Beispiele:

- Entsprechung: Helmich E-Book, Folgen 1 und 2
 - Ein zweidimensionales Bild mit BlueJ (nach Barnes und Kölling)
 - Analogien zum Objektbegriff

Unterrichtsvorhaben EF 2:

Gegenstand:

Grundlagen der objektorientierten Programmierung und algorithmischer Grundstrukturen (Fallunterscheidung und Schleifen) in Java

Sequenzierung:

- Entwicklung von Methoden mit Wertrückgabe, die arithmetisch (ohne Iteration) einen Wert berechnen
- Erarbeitung des Prinzips der Datenkapselung
- Entwicklung von Methoden, die Attributwerte und/oder berechnete Werte in der Konsole ausgeben
- Optimierung der Konsolenausgabe durch Formatierung der Ausgabe
- Erweiterung der Funktionalität der o.g. Methoden durch Verwendung von Fallunterscheidung (if, if-else, geschachtelte if-else-Abfragen, optional: switch-Anweisung)

- Erarbeitung der Notwendigkeit, Anweisungen mehrfach zu wiederholen im Anwendungskontext von Wertberechnungen
- Entwicklung von Methoden, die unter Verwendung von Iteration (durch Schleifen) Werte berechnen und/oder wiederholte Ausgaben produzieren (while-Schleifen, for-Schleifen, fakultativ: do-while-Schleifen)
- Darstellung von Methoden als Nassi-Shneiderman-Diagramme (Struktogramme)

Bezüge zu Kompetenzerwartungen des Kernlehrplans:

Die Schülerinnen und Schüler...

- 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)
- nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D)

Beispiele:

- Entsprechung: Helmich E-Book, Folgen 3 und 4
 - Projekt »Waage«
 - Projekt »Auto«

Unterrichtsvorhaben EF 3:

Gegenstand:

Modellierung und Implementierung von Klassen- und Objektbeziehungen (Vertiefung) anhand anwendungsbezogener Projekte

Sequenzierung:

- Entwicklung eigener Klassen mit Bezugsobjekten vom Typ vorgefertigter Java-Klassen
- Entwicklung eigener Klassen als Unterklassen vorgefertigter Java-Klassen (zum Beispiel: `JApplet`)
- Überschreibung von Methoden der Oberklassen in den Unterklassen (zum Beispiel: `paint(Graphics g)` aus `JApplet`)
- Verwendung von Schleifenstrukturen zur iterierten Instantiierung von Bezugsobjekten

- Erarbeitung der Beziehungen *Assoziation* und (in Ansätzen) *Vererbung* in Modellierung (insb. auch Darstellung in UML) und Implementierung
- **Optionale** Erarbeitung von Grundlagen der Ereignissteuerung (z.B. über Projekt »Roboter«, siehe unten)

Bezüge zu Kompetenzerwartungen des Kernlehrplans:

- 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),
- 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)
- nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D)

Beispiele:

- Entsprechung: Helmich E-Book, Folgen 5 und 6
 - Projekt »Java-Applets mit grafischer Ausgabe«
 - Projekt »Roboter«

Unterrichtsvorhaben EF 4:

Gegenstand:

Such- und Sortierverfahren anhand kontextualisierter Beispiele, verzahnt mit der Implementierung von *Bubblesort* auf Arrays

Sequenzierung:

- Verwendung von *Arrays* zur Speicherung mehrere Werte gleichen (primitiven) Typs
- Iteration über *Arrays* mittels for- und/oder while-Schleifen und Füllen von *Arrays* mit Zufallszahlen
- Speichern von Objekt-Referenzen in *Arrays*
- Kennenlernen/Erarbeitung der Sortierverfahren *Bubblesort*, *Selectionsort* und *Insertionsort* im Anwendungskontext (ohne Implementierung)
- Anschaulicher Vergleich dieser Sortierverfahren hinsichtlich ihrer Effizienz
- Implementierung von *Bubblesort* im Anwendungskontext auf einem *Array* primitiven Datentyps

- Erarbeitung der binären Suche an einem Alltagsbeispiel, anschaulicher Vergleich der Effizienz mit *linearer Suche* und Implementierung von *binärer Suche* auf einem Array primitiven Datentyps

Bezüge zu Kompetenzerwartungen des Kernlehrplans:

- 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)
- nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D)

Beispiele:

- Entsprechung: Helmich E-Book, Folgen 7 und 8

Unterrichtsvorhaben EF 5:

(eng angelehnt an den entsprechenden Abschnitt in der Modellversion des schulinternen Lehrplans)

Gegenstand:

Entwicklung der digitalen Datenverarbeitung und Grundlagen des Datenschutzes

Sequenzierung:

- Selbstständige Erarbeitung von Themen und Präsentation als Referate, z.B.
 - »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«
- Vertiefung zentraler Aspekte zum Thema Datenschutz, z.B.
 - grundlegende Begriffe
 - Bundesdatenschutzgesetz
 - Fallbeispiele

Bezüge zu Kompetenzerwartungen des Kernlehrplans:

Die Schülerinnen und Schüler

- beschreiben und erläutern den Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der »Von-Neumann-Architektur« (A)
- 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)
- nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D)

Beispiele:

- nach Schülerwahl

2.3. Jahrgangsstufe Q1

Die derzeitige Ausgestaltung des Fachs Informatik in der Qualifikationsphase orientiert sich notwendigerweise an den Vorgaben des Zentralabiturs NRW sowie den übrigen Punkten der Obligatorik des Lehrplans in der Fassung von 1999. Entsprechend den veröffentlichten »Vorgaben zu den unterrichtlichen Voraussetzungen für die schriftlichen Prüfungen im Abitur in der gymnasialen Oberstufe« sowie den Festlegungen der Fachkonferenz wurden die folgenden Zielkompetenzen und Sequenzierungen der Unterrichtsinhalte für die Jahrgangsstufen 11 und 12 festgelegt.

Konkretisierungen ergeben sich vor allem über die veränderlichen weiteren Vorgaben zum Zentralabitur NRW (<http://www.standardsicherung.nrw.de>) zum Fach Informatik. **Zusätzliche Inhalte fließen nach Ermessen der unterrichtenden Lehrkraft ein**, wo es sinnvoll erscheint (z.B. Thematisierung von Heapsort im Zusammenhang mit Baumstrukturen). Die folgenden Auflistungen an Kompetenzen und Inhalten ist ausgerichtet auf den Abiturjahrgang 2012, sie ist jeweils zu Ersetzen durch die Fassung des entsprechenden Abiturjahrgangs.

Zum Unterschied von Grund- und Leistungskursen sei zusätzlich auf Abschnitt 3.3 des Lehrplans Informatik für die Sekundarstufe II verwiesen. Beide Kursartenarten basieren unverzichtbar auf dem Unterricht der Jahrgangsstufe EF.

2.3.1. Schwerpunktmäßige Zielkompetenzen der Jahrgangsstufe Q1

- Ein Informatikmodell gewinnen: Probleme eingrenzen und spezifizieren, reduzierte Systeme definieren
 - Problemstellungen eingrenzen und Probleme strukturieren
 - Anforderungen an ein Modell aufstellen
 - ein reduziertes Modell für die Problemstellung definieren
 - eine erste Lösungsstrategie entwerfen
- Lösungen nach einem Programmierkonzept realisieren, überprüfen und weiterentwickeln
 - Lösungskonzepte implementieren und testen
 - Lösungen dokumentieren
 - Lösungen nach vorgegebenen Kriterien bewerten
 - Problemlösungen optimieren und weiterentwickeln
- Daten und Algorithmen abstrahieren
 - allgemeine Strategien und Standardlösungen kennen und anwenden
 - Programmierkonzepte allgemeiner und spezieller Art verstehen und benutzen
 - Formen des Strukturierens einsetzen
 - problembezogene Objekte und ihre Wechselwirkungen spezifizieren
 - ein Lösungskonzept als Denkschema (Modellierung) entwickeln

2.3.2. Unterrichtsinhalte Grundkurs

1. Konzepte des objektorientierten Modellierens (integrierende Wiederholung)
 - a) Klasse, Objekt, Attribut, Methode, Geheimnisprinzip
 - b) Klassendiagramme
 - c) Beziehungen zwischen Klassen: (Gerichtete) Assoziation mit Multiplizität, Vererbung
 - d) Abstrakte Klassen, Polymorphie
2. Datenstrukturen

- a) Lineare Strukturen mit den Akzenten
 - i. Schlange und Stapel
 - A. Anwendung der Standardoperationen
 - B. Implementation der Standardoperationen
 - ii. Lineare Liste
 - A. Anwendung der Standardoperationen
- b) Such- und Sortieralgorithmen für Felder und Listen
 - i. Suchen
 - ii. Sortieren durch direktes Einfügen
- c) Baumstrukturen mit den Akzenten
 - i. Binärbaum
 - A. Anwendung der Standardoperationen
 - B. Traversierungsalgorithmen
 - ii. Binärer Suchbaum
 - A. Anwendung der Standardoperationen

2.3.3. Unterrichtsinhalte Leistungskurs

1. Konzepte des objektorientierten Modellierens (integrierende Wiederholung)
 - a) Klasse, Objekt, Attribut, Methode, Geheimnisprinzip
 - b) Klassendiagramme
 - c) Beziehungen zwischen Klassen: (Gerichtete) Assoziation mit Multiplizität, Vererbung
 - d) Abstrakte Klassen, Polymorphie
2. Datenstrukturen
 - a) Lineare Strukturen mit den Akzenten
 - i. Schlange und Stapel
 - A. Anwendung der Standardoperationen
 - B. Implementation der Standardoperationen
 - ii. Lineare Liste
 - A. Anwendung der Standardoperationen
 - b) Such- und Sortieralgorithmen für Felder und Listen
 - i. Suchen
 - ii. Sortieren durch direktes Einfügen
 - iii. Quicksort
 - c) Baumstrukturen mit den Akzenten
 - i. Binärbaum
 - A. Anwendung der Standardoperationen
 - B. Traversierungsalgorithmen
 - ii. Binärer Suchbaum
 - A. Anwendung der Standardoperationen
 - B. Implementation von insert und search
 - iii. Ungerichteter gewichteter Graph mit den Akzenten

- A. Anwendung der Standardoperationen
- B. Breiten- und Tiefensuche
- C. Suche des kürzesten Weges zwischen zwei Knoten: Backtracking, Dijkstra-Algorithmus

2.4. Jahrgangsstufe Q2

Einleitende Kommentare siehe Abschnitt 2.3 (JgSt. Q1)

2.4.1. Schwerpunktmäßige Zielkompetenzen der Jahrgangsstufe Q2

- Algorithmen, Sprachkonzepte und Automatenmodelle beurteilen
 - den Algorithmenbegriff und den Begriff der Berechenbarkeit verstehen
 - Grenzen von Verfahren und Methoden abschätzen
 - formale Sprachen und Grammatiken untersuchen
 - Syntaxregeln und Beschreibungssysteme beurteilen
 - Automatenmodelle und akzeptierte Sprachen analysieren und beurteilen
 - Effizienzuntersuchungen durchführen
- Typische Einsatzbereiche, Möglichkeiten, Grenzen, Chancen und Risiken der Informations- und Kommunikationssysteme untersuchen und einschätzen
 - die Entwicklung von Informatiksystemen kennen lernen und verstehen
 - den Einsatz von Informations- und Kommunikationssystemen in verschiedenen gesellschaftlichen Bereichen untersuchen und bewerten
 - den Strukturwandel in Industrie und Gesellschaft erkennen und beschreiben
 - die Notwendigkeit des verantwortungsbewussten Umgangs mit Informationen einschätzen
- Technische, funktionale und organisatorische Prinzipien von Hard- und Softwaresystemen kennen und einordnen
 - Kommunikations- und Vernetzungsstrukturen einordnen
 - Anwendungssoftware klassifizieren
 - Benutzerführung, Funktionsumfang und Schnittstellen untersuchen und bewerten
- Technische, funktionale und organisatorische Prinzipien von Hard- und Softwaresystemen kennen und einordnen
 - die Struktur und Funktionsweise eines von-Neumann-Rechners analysieren
 - alternative Rechnerkonzepte und Maschinenmodelle beschreiben
 - die Informationsdarstellung auf der Maschinenebene analysieren
 - Funktionen und Komponenten der Systemsoftware kennen lernen und beschreiben

2.4.2. Unterrichtsinhalte Grundkurs

1. Endliche Automaten und formale Sprachen
 - a) Modellieren kontextbezogener Problemstellungen als deterministische endliche Automaten
 - b) Darstellung von deterministischen endlichen Automaten als Graph und als Tabelle
 - c) Formale Sprachen: Reguläre Sprachen und ihre Grammatiken
2. Modellieren und implementieren kontextbezogener Problemstellungen als Netzwerkanwendungen
 - a) Netzwerkprotokolle
 - b) Client-Anwendungen

- c) Client-Server-Anwendungen
 - d) Kryptografie
 - i. Symmetrische Verschlüsselungsverfahren (Caesar, Vigenère)
 - ii. Asymmetrische Verschlüsselungsverfahren (RSA)
 - iii. Schlüsselaustausch (Diffie-Hellmann)
3. Stufen zwischen Hard- und Software (*gem. Obligatorik des Lehrplans*)
- a) von-Neumann-Rechnerarchitektur
 - b) Zahlensysteme (Basis 2, 10 und 16)
 - c) interne Repräsentation von Daten
 - d) einfache bitweise Operationen
 - e) einfache Modellrechner (z.B. Registermaschine)
 - f) Übertragung einfacher Methoden (algorithmische Grundstrukturen (Verzweigung, Schleifen)) in eine Assembler-Sprache

2.4.3. Unterrichtsinhalte Leistungskurs

1. Endliche Automaten und formale Sprachen
 - a) Modellieren kontextbezogener Problemstellungen als deterministische endliche Automaten
 - b) Darstellung von deterministischen endlichen Automaten als Graph und als Tabelle
 - c) Entwicklung eines Parsers für eine einfache formale Sprache (nur Leistungskurs)
2. Modellieren und implementieren kontextbezogener Problemstellungen als Netzwerkanwendungen
 - a) Netzwerkprotokolle
 - b) Client-Anwendungen
 - c) Client-Server-Anwendungen
 - d) Kryptografie
 - i. Symmetrische Verschlüsselungsverfahren (Caesar, Vigenère)
 - ii. Asymmetrische Verschlüsselungsverfahren (RSA)
 - iii. Schlüsselaustausch (Diffie-Hellmann)
3. Stufen zwischen Hard- und Software (*gem. Obligatorik des Lehrplans*)
 - a) von-Neumann-Rechnerarchitektur
 - b) Zahlensysteme (Basis 2, 10 und 16)
 - c) interne Repräsentation von Daten
 - d) einfache bitweise Operationen
 - e) einfache Modellrechner (z.B. Registermaschine)
 - f) Übertragung einfacher Methoden (algorithmische Grundstrukturen (Verzweigung, Schleifen)) in eine Assembler-Sprache

2.5. Zur Rolle von Hausaufgaben für das Fach Informatik in der Sekundarstufe II

Das Fach Informatik ist durch die Notwendigkeit der Einübung von Programmier Techniken sehr übungsintensiv. **Zum Erwerb der Zielkompetenzen des Fachs reicht die bloße im Unterricht zur Verfügung stehende Zeit daher nicht aus.** Eine erfolgreiche Teilnahme am Unterricht setzt somit eine kontinuierliche häusliche Vor- und Nachbereitung sowie kontinuierliche, gründliche Erledigung von Hausaufgaben unabdingbar voraus. Nach Ermessen der Fachlehrkraft können Hausaufgaben jederzeit bewertet werden und in die Bildung der Note zur sonstigen Mitarbeit angemessen gewichtet einfließen. Nicht angefertigte Hausaufgaben sind mit ungenügend zu bewerten, sofern der Grund des Versäumnisses von der Schülerin/dem Schüler zu vertreten ist.

Verfügt eine Schülerin/ein Schüler zuhause **nachweislich** nicht über die notwendigen technischen Voraussetzungen zur Bearbeitung gestellter Hausaufgaben, so wird ihr/ihm nach Absprache mit dem Fachlehrer im Nachmittagsbereich (oder in Freistunden auch vormittags) Zugang zu den Informatikräumen der Schule gewährt – sofern das Verhalten der Schülerin/des Schülers dies nach Ansicht der unterrichtenden Lehrkraft zulässt.

3. Leistungsbewertung

3.1. Sekundarstufe I

Die Leistungsbewertung im Fach Informatik folgt den in § 6 APO-S I und § 48 SchulG aufgeführten Grundsätzen. Dementsprechend gliedert sich die Leistungsbewertung in die Unterbereiche *Schriftliche Arbeiten* und *Sonstige Leistungen im Unterricht*. Beide Teile sollen zu ca. 50% in die Leistungsbewertung mit eingehen. Die Bildung der Gesamtnote erfolgt nicht rein rechnerisch, sondern liegt im Ermessen der unterrichtenden Lehrkraft.

Grundlagen der Leistungsbewertung bilden alle im Unterricht vermittelten Kompetenzen. Bewertet werden demzufolge alle von Schülern/innen einzeln oder in einer Gruppe im Unterricht erbrachten Leistungen sowie die Art und Form der Darstellung.

3.1.1. Schriftliche Arbeiten

Pro Schulhalbjahr werden zwei schriftliche Arbeiten (»Kursarbeiten«) im Umfang von 1-2 Schulstunden geschrieben. Einmal pro Schuljahr kann eine Arbeit durch eine andere Form der Leistungsüberprüfung (Projekt, Gruppenarbeit, etc.) ersetzt werden. Die Entscheidungen hinsichtlich zeitlicher Dauer und Alternative zu einer Kursarbeit trifft die unterrichtende Lehrkraft.

Für die Bewertung schriftlicher Arbeiten wird eine Orientierung an folgendem Schema zur Bildung einer Gesamtnote empfohlen:

Note	Prozent (%)
sehr gut (1)	100,0 – 87,5
gut (2)	87,0 – 75,0
befriedigend (3)	74,5 – 62,5
ausreichend (4)	62,0 – 50,0
mangelhaft (5)	49,5 – 25,0
ungenügend (6)	24,5 – 0,0

Es wird auf die Vorrangigkeit der Notendefinitionen gem. § 3 Abs. 3 SchulG hingewiesen. Insbesondere bedeutet dies, dass Abweichungen vom o.g. Schema nach Ermessen der unterrichtenden Lehrkraft möglich sind. Die Bildung der Gesamtnote einer schriftlichen Arbeit muss darüber hinaus generell nicht unbedingt unflexibel aus einem solchen Schema ableitbar sein.

3.1.2. Sonstige Leistungen im Unterricht („Sonstige Mitarbeit“)

Der Bereich *Sonstige Leistungen* umfasst i.d.R. vor allem die Felder

- mündliche Beiträge
- praktische Leistungen
- allg. Mitarbeit im Unterricht

Der konkrete Beurteilungs- und Bewertungsmodus unterliegt dem Ermessen der unterrichtenden Lehrkraft und basiert sowohl auf langfristigen Beobachtungen als auch punktuellen Überprüfungen einzelner Kompetenzen (z.B. durch schriftliche Übungen, Abfragen, o.ä.).

Konkrete Grundlage zur Bewertung der sonstigen Leistungen (»Sonstige Mitarbeit«) ist das in Anhang A dargestellte kriteriale Raster.

3.2. Sekundarstufe II

Die Leistungsbewertung im Fach Informatik folgt den in §§ 13-19 APO-GOST und § 48 SchulG aufgeführten Grundsätzen unter Berücksichtigung von Abschnitt 3 des *Kernlehrplans Informatik für die gymnasiale Oberstufe*.

Mehr noch als in der Sekundarstufe I setzt der Informatikunterricht in der Sekundarstufe II auf die Eigenständigkeit und Selbstverantwortung der Schülerinnen und Schüler, indem diese ihren Lernprozess selbst steuern und größtenteils in Partnerarbeit oder Kleingruppen gemeinsam die Lösung dargestellter Probleme erarbeiten. Eine regelmäßige Abgabe von eigenen Lösungen bzw. Bearbeitungen von gestellten Aufgaben ist entscheidende Bewertungsgrundlage des Bereichs Sonstige Mitarbeit.

3.2.1. Sonstige Mitarbeit

Der Bereich *Sonstige Mitarbeit* kann gem. Abschnitt 3 des Kernlehrplans u.a. die folgenden Bestandteile umfassen

- unterschiedliche Formen der selbstständigen und kooperativen Aufgabenerfüllung
- Beiträge zum Unterricht
- von der Lehrkraft abgerufene Leistungsnachweise wie z.B. die schriftliche Übung
- von der Schülerin oder dem Schüler vorbereitete, in abgeschlossener Form eingebrachte Elemente zur Unterrichtsarbeit, die z.B. in Form von
 - Präsentationen
 - Protokollen
 - Referaten
 - Portfolios

wobei nicht zu erwarten ist, dass jeder Bestandteil von allen Schülern in jedem Halbjahr erbracht werden (sollen). Die obige Auflistung ist außerdem nicht abschließend. Der konkrete Beurteilungs- und Bewertungsmodus unterliegt dem Ermessen der unterrichtenden Lehrkraft und basiert sowohl auf langfristigen Beobachtungen als auch punktuellen Überprüfungen einzelner Kompetenzen (z.B. durch schriftliche Übungen, Abfragen, o.ä.).

Konkrete Grundlage zur Bewertung der sonstigen Mitarbeit ist das in Anhang B dargestellte kriteriale Raster.

3.2.2. Klausuren

Die Bewertung von Klausuren soll mittels eines kriterialen Punkterasters erfolgen. Insbesondere ab der Jahrgangsstufe Q1 wird als Bewertungsschema hinsichtlich der Notenvergabe eine Orientierung an der Prozent->Notenzuordnung des Zentralabiturs empfohlen:

Note	ab Prozent
sehr gut plus (1+)	95%
sehr gut (1)	90%
sehr gut minus (1-)	85%
gut plus (2+)	80%
gut (2)	75%
gut minus (2-)	70%
befriedigend plus (3+)	65%
befriedigend (3)	60%
befriedigend minus (3-)	55%
ausreichend plus (4+)	50%
ausreichend (4)	45%
ausreichend minus (4-)	39%
mangelhaft plus (5+)	33%
mangelhaft (5)	27%
mangelhaft minus (5-)	20%
ungenügend (6)	0%

Neben der fachlich-inhaltlichen Leistung kann auch die *Art und Form der Darstellung* in angemessenem Umfang (bis 10% der Gesamtpunktzahl) als Bewertungskriterium herangezogen werden. Es wird auf die Vorrangigkeit der Notendefinitionen gem. § 3 Abs. 3 SchulG hingewiesen. Insbesondere bedeutet dies, dass Abweichungen vom o.g. Schema nach Ermessen der unterrichtenden Lehrkraft möglich sind.

A. Kriteriales Raster Sonstige Mitarbeit Sek. I

Beurteilungskriterien im Fach Informatik für Sonstige Mitarbeit, Sek. I

Note	Unterrichtsbeiträge	Arbeit am Rechner	Programmierungsumgebung und -sprache	Arbeitsverhalten
sehr gut 1	Fördert den U. mit häufigen, gut durchdachten Beiträgen. Beteiligt sich erfolgreich beim Lösen komplizierter Probleme. Erkennt schnell Fehler und deren Quelle.	Löst die gestellten Aufgaben schnell und sicher. Löst selbst gewählte Zusatzaufgaben.	Kann sicher Projekte speichern, löschen oder kompilieren. Abgaben sind strukturiert, enthalten Kommentare und sinnvolle Bezeichnungen. Kann Fehlermeldungen des Informatiksystems richtig deuten.	Arbeitet sehr konzentriert. Beendet begonnene Aufgabenstellung selbständig zu Hause.
gut 2	Beteiligt sich regelmäßig durch gute Beiträge. Hat die U.-inhalte der letzten Stunden voll verstanden.	Löst die gestellten Aufgaben fast immer schnell und richtig.	Kann Projekte speichern, löschen oder kompilieren. Abgaben sind in der Regel strukturiert, enthalten Kommentare und sinnvolle Bezeichnungen. Kann Fehlermeldungen des Informatiksystems meistens richtig deuten.	Arbeitet konzentriert.
befriedigend 3	Beteiligt sich häufiger unaufgefordert durch sachbezogene Beiträge. Verfügt über ein zufriedenstellendes Grundwissen.	Löst die Aufgaben, benötigt aber relativ viel Zeit dazu.	Hat teilweise Probleme mit dem Informatiksystem. Abgaben sind nicht immer strukturiert/übersichtlich.	Lässt sich nur selten ablenken. Versucht ernsthaft, die Aufgaben zu lösen.
ausreichend 4	Beteiligt sich hin und wieder ohne Anforderung. Kann auf Nachfrage die grundlegende Problematik des aktuellen U.-stoffes erklären.	Kann die Aufgaben nur mit Hilfen lösen.	Hat Schwierigkeiten mit dem Informatiksystem. Unübersichtliche/unstrukturierte Abgaben. Versteht oft die Fehlermeldungen des Informatiksystems nicht.	Lässt sich öfters ablenken. Muss zur Konzentration aufgefordert werden.
mangelhaft 5	Beteiligt sich praktisch nie ohne Anforderung. Kann die Grundlagen des aktuellen Unterrichtsstoffes nicht korrekt wiedergeben.	Hat große Probleme bei der Bearbeitung der gestellten Aufgaben. Kann diese nur mit massiver Hilfe bearbeiten.	Hat Schwierigkeiten mit dem Informatiksystem. Versteht die Fehlermeldungen des Informatiksystems nicht. Kann das Informatiksystem nicht zielführend/sinnvoll bedienen bzw. verwenden	Lässt sich leicht ablenken. Beschäftigt sich öfters mit anderen Dingen.
ungenügend 6	Beteiligt sich praktisch überhaupt nicht. Kann dem aktuellen Unterrichtsstoff nicht folgen. Kann auf Nachfragen keine fachlich richtigen Aussagen machen.	Weiß überhaupt nicht, wie man die gestellten Aufgaben lösen soll. Bemüht sich auch nicht.	Hat größte Schwierigkeiten mit dem Informatiksystem. Versteht die Fehlermeldungen des Informatiksystems überhaupt nicht. Kann nicht programmieren.	Ist absolut passiv. Arbeitet nur nach massiver Aufforderung. Spielt z.B. Computerspiele oder surft im Internet.

Zusätzlich können schriftliche Tests (allerdings keinesfalls mehr als drei pro Halbjahr) und Referate zur Beurteilung der Sonstigen Mitarbeit herangezogen werden.

B. Kriteriales Raster Sonstige Mitarbeit Sek. II

Beurteilungskriterien im Fach Informatik für Sonstige Mitarbeit, Sek. II

Note	Unterrichtsbeiträge	Arbeit am Rechner	Programmierungsumgebung und -sprache	Arbeitsverhalten
sehr gut 1	Fördert den U. mit häufigen, gut durchdachten Beiträgen. Beteiligt sich erfolgreich beim Lösen komplizierter Probleme. Erkennt schnell Programmierfehler.	Löst die gestellten Aufgaben schnell und sicher. Löst selbst gewählte Zusatzaufgaben.	Kann sicher Programme speichern, löschen oder kompilieren. Programmtext enthält Einrückungen, Kommentare und sinnvolle Variablennamen. Kann Fehlermeldungen des Compilers richtig deuten.	Arbeitet sehr konzentriert. Beendet begonnene Aufgabenstellung selbständig zu Hause.
gut 2	Beteiligt sich regelmäßig durch gute Beiträge. Hat die U.-inhalte der letzten Stunden voll verstanden.	Löst die gestellten Aufgaben fast immer schnell und richtig.	Kann Programme speichern, löschen oder kompilieren. Programmtext enthält in der Regel Einrückungen, Kommentare und sinnvolle Variablennamen. Kann Fehlermeldungen des Compilers meistens richtig deuten.	Arbeitet konzentriert.
befriedigend 3	Beteiligt sich häufiger unaufgefordert durch sachbezogene Beiträge. Verfügt über ein zufriedenstellendes Grundwissen.	Löst die Aufgaben, benötigt aber relativ viel Zeit dazu.	Hat teilweise Probleme mit der Programmierungsumgebung. Programme sind nicht immer übersichtlich geschrieben (ohne Einrückungen).	Lässt sich nur selten ablenken. Versucht ernsthaft, die Aufgaben zu lösen.
ausreichend 4	Beteiligt sich hin und wieder ohne Aufforderung. Kann auf Nachfrage die grundlegende Problematik des aktuellen U.-stoffes erklären.	Kann die Aufgaben nur mit Hilfen lösen.	Hat Schwierigkeiten mit der Programmierungsumgebung. Unübersichtliche Programmierung. Versteht oft die Fehlermeldungen des Compilers nicht.	Lässt sich öfters ablenken. Muss zur Konzentration aufgefordert werden.
mangelhaft 5	Beteiligt sich praktisch nie ohne Aufforderung. Kann die Grundlagen des aktuellen Unterrichtsstoffes nicht korrekt wiedergeben.	Hat große Probleme bei der Bearbeitung der gestellten Aufgaben. Kann diese nur mit massiver Hilfe bearbeiten.	Hat Schwierigkeiten mit der Programmierungsumgebung. Versteht die Fehlermeldungen des Compilers nicht. Kann nicht programmieren.	Lässt sich leicht ablenken. Beschäftigt sich öfters mit anderen Dingen.
ungenügend 6	Beteiligt sich praktisch überhaupt nicht. Kann dem aktuellen Unterrichtsstoff nicht folgen. Kann auf Nachfragen keine fachlich richtigen Aussagen machen.	Weiß überhaupt nicht, wie man die gestellten Aufgaben lösen soll. Bemüht sich auch nicht.	Hat größte Schwierigkeiten mit der Programmierungsumgebung. Versteht die Fehlermeldungen des Compilers überhaupt nicht. Kann nicht programmieren.	Ist absolut passiv. Arbeitet nur nach massiver Aufforderung. Spielt z.B. Computerspiele oder surft im Internet.

Zusätzlich können schriftliche Tests (allerdings keinesfalls mehr als drei pro Halbjahr) und Referate zur Beurteilung der Sonstigen Mitarbeit herangezogen werden.