Sekundarstufe II

Lehrplan Informatik
Städt. Röntgen-Gymnasium

Inhalt

Rahmenbedingungen der fachlichen Arbeit 3
Allgemeines zur Schule 3
Unterrichtliche Situation Sekundarstufe I und II 3
Zieldefinition 3
Planungstechnische Rahmenbedingungen 3
Entscheidungen zum Unterricht 4
Grundsätze der fachmethodischen und fachdidaktischen Arbeit 4
Verwendete Software 5
Überblick über Lernmodule Einführungs- und Qualifikationsphase 6
Überblick über Lernmodule in der Einführungsphase 6
Überblick über Lernmodule in der Qualifikationsphase (GK) 9
Grundsätze der Leistungsbewertung und Leistungsrückmeldung 16
Beurteilungsbereich Klausuren 16
Beurteilungsbereich sonstige Mitarbeit 16
Qualitätssicherung und Evaluation 18

Rahmenbedingungen der fachlichen Arbeit

Allgemeines zur Schule

Das Röntgen-Gymnasium ist eines von vier öffentlichen Gymnasien der Stadt. Das Röntgen-Gymnasium ist in der Sekundarstufe I in der Regel dreizügig und wird als offenes Ganztagsgymnasium geführt.
In die Einführungsphase der Sekundarstufe II wurden in den letzten Jahren regelmäßig etwa 20 Schülerinnen und Schüler neu aufgenommen.
Der Unterricht findet im 67,5-Minuten-Takt statt, die Kursblockung sieht grundsätzlich für Grundkurse zwei Unterrichtseinheiten (UE) und für Leistungskurse 3 UE vor, von denen eine um 22,5 Minuten verlängert wird.

Unterrichtliche Situation Sekundarstufe I und II

Im Differenzierungsbereich der Sekundarstufe I, d.h. in den Jahrgangsstufen 8 und 9 wird grundsätzlich ein Informatikkurs angeboten. Da dieser nicht verpflichtend ist, können die fachlichen Inhalte in der Einführungsphase (EF) nicht als Vorwissen vorausgesetzt werden.
Durch das Wahlverhalten der Schülerinnen und Schüler konnten in der Vergangenheit meist 2 Grundkurse in der Einführungsphase zum Abitur entstehen. Die Zusammensetzung reduzierte sich zur Qualifikationsphase typischerweise, so dass nur nur ein Kurs bis zum Abitur zu Ende geführt werden konnte.

Zieldefinition

Für Referendare, die zu Ausbildungszwecken nur eine kurze Zeit an unserer Schule verbringen, ist es notwendig, sie schnell und präzise über die abgesprochenen fachlichen Inhalte und Umgangsformen z.B. zur Leistungsbewertung zu informieren.
Die durch das Curriculum vorgenommenen Standardisierung ist weiterhin positiv für Schülerinnen und Schüler, die aus verschiedenen Gründen während ihrer Schullaufbahn in eine andere Lerngruppe wechseln. Bei schulinternen Wechseln kann die durchgängige thematisch gleichbleibende Ausbildung gewährleistet und bei Wechseln aus anderen Schulen zumindest eine leichtere Vorbereitung ermöglicht werden.

Planungstechnische Rahmenbedingungen

Die Planungen zur Umsetzung des Curriculums beinhalten eine Aufteilung in verschiedene Module. Aus didaktischen Gründen kann es sinnvoll sein, von der vorgegebenen Reihenfolge der Module abzuweichen, dies widerspricht jedoch dem Zielgedanken der Standardisierung und soll daher nur in Ausnahmefällen und begründet geschehen.

Bei der Unterrichtsplanung muss sichergestellt werden, das sämtliche im Lehrplan[1] vorgeschriebenen inhaltlichen und methodischen Kompetenzen von Schülerinnen und Schülern erreicht werden können. Im Lehrplan wird dabei zuerst auf weit abstrahierender Ebene von Inhaltsfeldern und Kompetenzbereichen gesprochen. Diese werden danach miteinander verknüpft und es werden nach Inhaltsbereich sortierte **konkretisierte Kompetenzerwartungen** geschaffen.
Die ausgearbeitete Modulübersicht enthält alle geforderten konkretisierten Kompetenzen, aus Gründen der Übersicht wurde jedoch auf eine Mehrfachnennung der Kompetenzen verzichtet (z.B. wäre *„interpretieren Fehlermeldungen und korrigieren den Quellcode (I)“* an vielen Modulen als behandelte Kompetenz einzutragen, teilweise liegt aber nicht der Fokus auf dieser Kompetenz, sondern sie wird eher nebenläufig trainiert). Die vorliegende Modulübersicht stellt daher nur eine Schwerpunktsetzung bzw. ein Mindestmaß der behandelten Kompetenzen dar.

Entscheidungen zum Unterricht

Grundsätze der fachmethodischen und fachdidaktischen Arbeit

1. Der Unterricht unterliegt der Wissenschaftsorientierung und ist dementsprechend eng verzahnt mit seiner Bezugswissenschaft.
2. Der Unterricht ist problemorientiert und soll von realen Problemen ausgehen und sich auf solche rückbeziehen.
3. Der Unterricht folgt dem Prinzip der Exemplarizität und soll ermöglichen, informatische Strukturen und Gesetzmäßigkeiten in den ausgewählten Problemen und Projekten zu erkennen.
4. Der Unterricht ist anschaulich sowie gegenwarts- und zukunftsorientiert und gewinnt dadurch für die Schülerinnen und Schüler an Bedeutsamkeit.
5. Der Unterricht ist handlungsorientiert, d.h. projekt- und produktorientiert angelegt.
6. Im Unterricht werden sowohl für die Schule didaktisch reduzierte als auch reale Informatiksysteme aus der Wissenschafts-, Berufs- und Lebenswelt eingesetzt.
7. Der Unterricht beinhaltet reale Begegnung mit Informatiksystemen.
8. Der Unterricht fördert die Zusammenarbeit zwischen den Schülern/innen und bietet ihnen Möglichkeiten zu eigenen Lösungen.
9. Die Schüler/innen erhalten Gelegenheit zu selbstständiger Arbeit und werden dabei unterstützt.
10. Der Unterricht fördert strukturierte und funktionale Partner- bzw. Gruppenarbeit.

[1] http://www.schulentwicklung.nrw.de/lehrplaene/lehrplannavigator-s-ii/gymnasiale-oberstufe/informatik/

Verwendete Software

Bei der verwendeten Software wurde darauf geachtet, dass sämtliche Programme kostenfrei im Netz verfügbar und möglichst auf verschiedenen Betriebssystem-Plattformen (Linux, Mac, Windows) einsetzbar sind.

Die angegebenen Download-Links sind zuletzt überprüft worden am 10.6.2015.

- Programmiersprache **Java** unter Verwendung der didaktischen Entwicklungsumgebung **BlueJ** und ausgewählten Komponenten der **„Stifte-und-Mäuse"-(SuM)-Bibliothek** Downloadlink für BlueJ/SuM: http://www.mg-werl.de/sum
- Zur einfachen Visualisierung von selbst zusammengestellten endlichen Automaten: **Jflap** http://www.jflap.org/jflaptmp/
- Zur Erstellung von verschiedenen UML-Diagrammtypen: **Umlet**. http://www.umlet.com Vor der Verwendung eines Informatiksystems soll die Erstellung mit Papier und Stift stehen.
- **WAMP**-Server als lokaler Webserver zum Umgang mit SQL-Datenbanken, http://www.wampserver.com/en/
- Java-**Bibliotheken** zum Zentralabitur http://www.schulentwicklung.nrw.de/lehrplaene/upload/klp_SII/if/MaterialZABI/Abitur_2017_GK_Implementationen_Dynamische_Datenstrukturen.zip

Überblick über Lernmodule Einführungs- und Qualifikationsphase

Überblick über Lernmodule in der Einführungsphase

Modul	Inhaltsfeld/Unterthemen	Erwartete Kompetenzen: Schülerinnen und Schüler...	Anzahl 67,5er UE
EF-IMG	**Informatik, Mensch und Gesellschaft** • Klärung der Begriffe Informatik, Computer • Überblick über die Inhalte der Fachwissenschaft Informatik • Syntax/Semantik/Pragmatik mit Syntaxdiagrammen, Anwendungsbeispiele von Informatiksystemen auf unterschiedlichen Ebenen des Informationsbegriffs	• nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D). • bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A).	3 2
EF-IS-1	**Informatiksysteme: Digitalisierung** • Binärsystem zur Digitalisierung von Zahlen • Binärcodes zur Digitalisierung von Zeichen	• stellen ganze Zahlen und Zeichen in Binärcodes dar (D). • interpretieren Binärcodes als Zahlen und Zeichen (D).	3
EF-IS-2	**Informatiksysteme: Dateisysteme, Internet** • Einführung in eine Lernplattform, u.a. eigenständiges Anlegen von Gruppenverzeichnissen • Sichere Passwörter • Speicherung von Dateien in Verzeichnisstrukturen (Diese Modul wird unterrichtsbegleitend thematisiert.)	• 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).	2
EF-IS-3	**Informatiksysteme: Einzelrechner** • Praktische Hardwareanalyse • Theoretische Hardwareanalyse (Von-Neumann-Architektur) • Vergleich Desktop/Notebook/Mobiltelefon	• beschreiben und erläutern den strukturellen Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der „Von-Neumann-Architektur“ (A). • erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung.	4

EF-OO-1	**Daten und ihre Strukturierung: Modellierung** • Analyse von realen Objekten bezüglich ihrer Attribute und Attributwerte (Objektdiagramme) • Kategorisierung von Objekten in Form von Klassen (Klassendiagramme ohne Programmierung) • Analyse und Modifikation von grafischen Primitive mit Hilfe von Punktnotationen (ohne Syntaxdiagramm) • Variablenkonzept mit Konstruktoraufruf, um grafische Primitive über Kommandozeile anzuordnen.	• ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu. • stellen den Zustand eines Objekts dar (D). • implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I). • interpretieren Fehlermeldungen und korrigieren den Quellcode (I).	
EF-OO-2	**Daten und ihre Strukturierung: Modellierung und Programmierung** • Syntax der Programmiersprache Java in Form von Syntaxdiagrammen (Deklaration, Zuweisung, Punktnotation) • Modellierung einer Klasse „Mobiltelefon“ nach ausformulierten Anforderungen (Neu: Deklaration von Methoden mit Parameter) • Dokumentation der Funktionalität von Klassen und Methoden • Implementierung • Rückblick mit Hilfe von Zustandsdiagrammen	• ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M). • dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D)	
EF-OO-3	**Daten und ihre Strukturierung: Assoziation I** • Zusammenstellung mehrere grafischer Primitive zu einem komplexeren Objekt. • Deklaration des Konstruktors (mit Befehlen zur Objekterzeugung) • Methoden zur Steuerung enthaltener Objekte (Punktnotation) **Daten und ihre Strukturierung: Assoziation II** • Erweiterung des Mobiltelefonprojekt um Netzproviderobjekte (Pro Provider vordefinierte Anzahl Verbindungs-“Slots“, pro Slot eine Assoziation einzeichnen) • Assoziationen herstellen • Objektspiel **//** Sequenzdiagramme	• Modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M). • stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D). • analysieren und erläutern eine objektorientierte Modellierung (A) • stellen die Kommunikation zwischen Objekten grafisch dar (M).	

EF-OO-4	**Daten und ihre Strukturierung: Lineare Datenstrukturen** (Übergang zum Inhaltsfeld Algorithmen) • Weiterentwicklung des Netzproviders zum Umgang mit (prinzipiell) beliebig vielen Mobiltelefonen unter Verwendung von Arrays • Umgang mit Arrays • Durchlaufen von Arrays mit Hilfe von Wiederholungsanweisungen	• Implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I). • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen ode rlineare Datensammlungen zu (M).	
EF-OO-5	**Daten und ihre Strukturierung: Vererbung I** • Modellierung verschiedene ähnlicher realer Gegenstände • Zusammenfassung der erstellten Klassen unter Verwendung von Generalisierung • Änderung der Perspektive zur Spezialisierung • Vergleich zwischen Generalisierung und Spezialisierung	• Modellieren Klassen unter Verwendung von Vererbung (M).	
EF-OO-6	**Daten und ihre Strukturierung: Vererbung II** • Verschiedene Varianten einer Klasse „Haus“ modellieren und implementieren • Erweiterung aller „Haus“-Klassen zur Lichtschaltung, daran Motivation des Vererbungskonzepts mit Fokus auf Spezialisierungen • Das Schlüsselwort „super“ am Beispiel des Überschreibens von Methoden	• implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I). • testen Programme schrittweise anhand von Beispielen (I).	
EF-A-1	**Algorithmen: Suchen und Sortieren** • Kartensortieren als intuitive Einführung • Definition des Begriff „Algorithmus“ • Min-Max-Sortierung • Bubble-Sort • Vergleich der Sortierverfahren bezüglich Laufzeit • Merge-Sort als Beispiel für einen Divide-and-Conquer-Algorithmus	• Modifizieren einfache Algorithmen und Programme (I). • entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M). • analysieren und erläutern einfache Algorithmen und Programme (A). • 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).	
		Summe UE á 67,5min	

Überblick über Lernmodule in der Qualifikationsphase (GK)

Modul	Inhaltsfeld/Unterthemen	Erwartete Kompetenzen: Schülerinnen und Schüler	Anzahl 67,5er UE
Q-IS	**Modulübergreifend werden in der gesamten Qualifikationsphase folgende Kompetenzen zur** **Nutzung von Informatiksystemen vermittelt:** • nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung und gemeinsamen Verwendung von Daten unter Berücksichtigung der Rechteverwaltung (K). • wenden eine didaktisch orientierte Entwicklungsumgebung zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an (I). • nutzen bereitgestellte Informatiksysteme und das Internet zur Erschließung, Aufbereitung und Präsentation fachlicher Inhalte (D). **Konkretisiert werden die Kompetenzen an folgenden Handlungsbeispielen:** • Internetbrowser zur Internetrecherche • BlueJ zur Programmierung in Java • Verwendung eines gemeinsamen Dropboxverzeichnis • Zur Anwendung von Datenbanken wird ein lokaler Webserver mit phpmyadmin verwendet.		
Q-A	**In den Modulen Q-A-1 bis Q-A-3 werden übergreifend folgende Kompetenzen vermittelt.** Schülerinnen und Schüler... • nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I) • beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A). • interpretieren Fehlermeldungen und korrigieren den Quellcode (I).		
Q-A-1	**Algorithmen: Sortieren** • Anknüpfung an Suchalgorithmen aus der EF • Implementation von Bubble-/MinMax-Sort auf Grundlage von Arrays (iterativ) • Implementation Mergesort auf Grundlage von Arrays (rekursiv) • *optional*: An Beispielen Rekursion und Iteration vergleichen (Fibonacci-Zahlen, Fakultät...)	• Stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D). • implementieren und erläutern iterative und rekursive Such- und Sortierverfahren (I). • entwickeln iterative und	

		rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“ (M).	
Q-A-2	**Algorithmen: Suchen** Suchen in Arrays (lineare und binäre Suche) beschreiben und implementieren Vergleich der Verfahren bezüglich Anzahl Operation	• beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (A).	
Q-A-3	**Algorithmen: Dynamische Datenstrukturen** • Rechercheauftrag mit digitaler Präsentation (einfach verkettete Liste, optional doppelt verkettet; Queue; Stack; Baum) => Kooperativ • Exkurs IMG: Urheberrechtsproblematik bei potenzieller Veröffentlichung der Präsentationen • Vorstellung der Klassenbibliotheken (mit Klassendiagrammen zur Übersicht) zum Zentralabitur • Implementieren einfacher Algorithmen auf dynamischen Datenstrukturen (z.B. Umdrehen der Inhalte eines Stacks/einer Queue) • *optional:* Implementierung einer Baum-Ausgabe-Methode mit vorgegebenen Inhaltsobjekten (rekursiv)	• stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D). • erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A). • implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I). • modifizieren Algorithmen und Programme (I). • analysieren und erläutern Algorithmen und Programme (A). • untersuchen und bewerten anhand von Fallbeispielen Auswirkungen des Einsatzes von Informatiksystemen sowie Aspekte der Sicherheit von Informatiksystemen, des Datenschutzes und des Urheberrechts (A).	
Q-OO-1	**Daten und ihre Strukturierung: Objektorientierung** • Erweiterung des Mobiltelefonprojekts um eine Speicherung in einer Datenstruktur (mit Diskussion, welche Struktur sinnvoll ist) => bewusst offen, ob unterschiedliche	• verwenden bei der Modellierung geeigneter Problemstellungen Möglichkeiten der Polymorphie (M). • modellieren Klassen mit	

	Strukturen implementiert werden sollen • Erweiterung der Klasse Handy als Spezialisierung der Klasse Item, dadurch Thematisierung des Konzepts abstrakter Klassen • Nutzung der Klassenbibliotheken zum Zentralabitur unter Verwendung der jeweiligen Dokumentation (BinarySearchTree, Stack, Queue, Item) • Klassendiagramm des Projekts um Multiplizitäten erweitern.	ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M). • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I). • modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M). • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M). • testen Programme systematisch anhand von Beispielen (I). • (Die Kompetenzen zur Objektorientierung aus der Einführungsphase werden begleitend wiederholt.*)	
Q-IMG	**Informatik, Mensch und Gesellschaft** • Durchführen des Planspiel Datenschutz • Analyse des Datenmaterials bezüglich vorgegebener Falldaten durch Rasterfahndung • Reflexion mit Fokus auf Grundsatz der Datensparsamkeit und Probleme bei der Rasterfahndung	• Ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen (D). • erläutern Eigenschaften, Funktionsweisen und Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nutzung (A). • untersuchen und bewerten Problemlagen, die sich aus dem Einsatz von Informatiksystemen ergeben, hinsichtlich rechtlicher Vorgaben,	

		ethischer Aspekte und gesellschaftlicher Werte unter Berücksichtigung unterschiedlicher Interessenlagen (A) • untersuchen und bewerten anhand von Fallbeispielen Auswirkungen des Einsatzes von Informatiksystemen sowie Aspekte der Sicherheit von Informatiksystemen, des Datenschutzes und des Urheberrechts (A).	
Q-DB-1	**Daten und ihre Strukturierung: Datenbanken** • Vermeidung von Anomalien (Einfügen, Ändern, Löschen) anhand eines vorgegebenen Negativbeispiels • Primärschlüssel zur Realisierung von Beziehungen zwischen Relationen • Normalisierung als Technik zur Vermeidung von Anomalien bei der Datenbankmodellierung • ER-Diagramme als Hilfsmittel zur Modellierung: Attribute, Relationen und Kardinalitäten erkennen und im ER-Diagramm darstellen. • Überführung eines ER-Diagramms in ein Datenbankschema	• Analysieren und erläutern eine Datenbankmodellierung (A). • modifizieren eine Datenbankmodellierung (A). • bestimmen Primär- und Sekundärschlüssel (M). • erläutern die Eigenschaften normalisierter Datenbankschemata (A). • überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D). • überführen Datenbankschemata in die 1. bis 3. Normalform (M). • ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten (M). • stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten mit Kardinalitäten in einem ER-Diagramm grafisch dar (D).	

		• modellieren zu einem ER-Diagramm ein relationales Datenbankschema (M)	
Q-DB-2	**Daten und ihre Strukturierung: Datenbanken/SQL** • Datenbanken und Tabellen sind vorgegeben, nur INSERT-Befehl zum Einfügen • SELECT-Befehl inklusive JOIN • *optional:* Planspiel Datenschutz erneut durchführen mit Datenbankeinsatz	• analysieren und erläutern die Syntax und Semantik einer Datenbankabfrage (A). • verwenden die Syntax und Semantik einer Datenbankabfragesprache, um Informationen aus einem Datenbanksystem zu extrahieren (I).	
Q-IS-1	**Informatiksysteme am Beispiel des E-Mail-Versands** • Netzwerktopologie (vgl. zu Hause und Provider) • Client-Server-Struktur • ISO/OSI Schichtenmodell: Zuordnung der Schichten bei der Übertragung einer E-Mail • Protokolle, SMTP-Protokoll an Telenet-Sitzung nachvollziehbar machen (lokalen Mailserver auf Schulrechner verwenden) • Verschlüsselung, vgl. Symmetrisch und Asymmetrisch	• Beschreiben und erläutern Netzwerk-Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken (A). • analysieren und erläutern Eigenschaften und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren (A).	
Q-FSA-1	**Formale Sprachen und Automaten: Fokus auf deterministische endliche Automaten** • Weitgehend selbstständige Arbeit mit Skript aus dem Netz mit begleitenden Fragestellungen zu: • Darstellung von Automaten in Tabellenform • Darstellung von Automaten als Graph • Nutzung von jflap zum interaktiven Test eigener Automaten • *optional:* Implementierung eines ausgewählten Automaten mit Java/BlueJ (z.B. Rückmeldung einer Wort-Eingabe-Methode „Wort akzeptiert/nicht akzeptiert“)	• analysieren und erläutern die Eigenschaften endlicher Automaten einschließlich ihres Verhaltens bei bestimmten Eingaben (A). • entwickeln und modifizieren zu einer Problemstellung endliche Automaten (M). • stellen endliche Automaten in Tabellen oder Graphen dar und überführen sie in die jeweils andere Darstellungsform (D).	
Q-FSA-2	**Formale Sprachen und Automaten** • Definition formaler Sprachen, Schreibweise	• analysieren und erläutern Grammatiken	

	von Grammatiken, Kurz-Notation von Sprachen (evtl. EBNF, noch in Konferenz festzulegen) • Zusammenhang Grammatik/Automat durch gemeinsame Sprache (evtl. Arbeitsteilig als Umwandlungsaufgabe Grammatik ←→ Automat) • Sprache eines Automaten (Nutzung von JFLAP an Schulrechnern) • Grenzen endlicher Automaten/regulärer Sprachen, (z.B. beim Zählen gleich vieler Klammern)	regulärer Sprachen (A). • modifizieren Grammatiken regulärer Sprachen (M). • ermitteln die formale Sprache, die durch eine Grammatik erzeugt wird (A). • entwickeln zu einer regulären Sprache eine Grammatik, die die Sprache erzeugt (M). • ermitteln die Sprache, die ein endlicher Automat akzeptiert • entwickeln zur Grammatik einer regulären Sprache einen zugehörigen endlichen Automaten	
Q-FSA-3	**Formale Sprachen und Automaten** • Grafische Darstellung und Implementierung eines Stringmatching-Automaten	• entwickeln zur akzeptierten Sprache eines Automaten eine zugehörige Grammatik (M). • beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken.	
Q-IS-2	**Informatiksysteme** • Von-Neumann-Architektur/Geschichte • Maschinennahes Programmieren • (Know-How-Computer)	• Erläutern die Ausführung eines einfachen maschinennahen Programms sowie die Datenspeicherung auf einer „Von-Neumann-Architektur“ (A).	
Q-IS	**Nutzung von Informatiksystemen** • Internetbrowser zur Internetrecherche • BlueJ zur Programmierung in Java • Erarbeitung von gemeinsamen Verhaltensweisen zur Konfliktvermeidung bei der Verwendung eines gemeinsamen Dropboxverzeichnis • Zur Anwendung von Datenbanken wird ein lokaler Webserver mit phpmyadmin verwendet. (Diese Modul wird unterrichtsbegleitend thematisiert.)	• nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung und gemeinsamen Verwendung von Daten unter Berücksichtigung der Rechteverwaltung (K). • wenden eine didaktisch orientierte Entwicklungsumgebung	

		zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an (I). • nutzen bereitgestellte Informatiksysteme und das Internet zur Erschließung, Aufbereitung und Präsentation fachlicher Inhalte(D).	
		Summe UE á 67,5min	

* Kompetenzen, die sich durch begleitende Wiederholung der Inhalte der Einführungsphase ergeben:

- stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D).
- analysieren und erläutern objektorientierte Modellierungen (A).
- ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M).
- dokumentieren Klassen.
- stellen die Kommunikation zwischen Objekten grafisch dar.
- ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M).

Grundsätze der Leistungsbewertung und Leistungsrückmeldung im Fach Informatik

Beurteilungsbereich Klausuren

Bei der Formulierung von Aufgaben werden die für die Abiturprüfungen geltenden Operatoren[2] des Faches Informatik schrittweise eingeführt, erläutert und dann im Rahmen der Aufgabenstellungen für die Klausuren benutzt.

Einführungsphase: 1 Klausur je Halbjahr, Dauer der Klausur: 90min
Grundkurse Q 1: 2 Klausuren je Halbjahr, Dauer der Klausuren: 90min
Grundkurse Q 2.1: 2 Klausuren, Dauer der Klausuren: 135min
Grundkurse Q 2.2: 1 Klausur unter Abiturbedingungen

Anstelle einer Klausur kann in Q 1.2 eine Facharbeit geschrieben werden.

Die Aufgabentypen, sowie die Anforderungsbereiche I-III sind entsprechend den Vorgaben in Kapitel 3 des Kernlehrplans zu beachten.

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

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

Von diesem kann aber im Einzelfall begründet abgewichen werden, wenn sich z.B. besonders originelle Teillösungen nicht durch Hilfspunkte gemäß den Kriterien des Erwartungshorizontes abbilden lassen oder eine Abwertung wegen besonders schwacher Darstellung (APO-GOSt[3] §13 (2)) angemessen erscheint.

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

Beurteilungsbereich sonstige Mitarbeit

Bewertungskriterien:

- Beteiligung am Unterrichtsgespräch
- Zusammenfassungen zur Vor- und Nachbereitung des Unterrichts
- Präsentation von Arbeitsergebnissen
- Referate
- Mitarbeit in Partner-/Gruppenarbeitsphasen
- Implementierung, Test und Anwendung von Informatiksystemen
- Sonstige schriftliche Leistungen wie Lernzielkontrollen

[2]Vgl. https://www.standardsicherung.schulministerium.nrw.de/abitur-gost/fach.php?fach=15 bzw. direkt https://www.standardsicherung.schulministerium.nrw.de/abitur-gost/getfile.php?file=172

[3]https://www.schulministerium.nrw.de/docs/Recht/Schulrecht/APOen/ bzw. direkt https://www.schulministerium.nrw.de/docs/Recht/Schulrecht/APOen/GY-Oberstufe-SekII/APO_GOSt_Oberstufe2011.pdf

Die Bewertungskriterien stützen sich auf

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

Besonderes Augenmerk ist dabei auf

- die sachliche Richtigkeit,
- die angemessene Verwendung der Fachsprache,
- die Darstellungskompetenz,
- die Komplexität und den Grad der Abstraktion,
- die Selbstständigkeit im Arbeitsprozess,
- die Präzision und
- die Differenziertheit der Reflexion zu legen.

Bei Gruppenarbeiten auch auf

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

Bei Projektarbeit darüber hinaus auf

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

Die Grundsätze der Leistungsbewertung werden zu Beginn eines jeden Halbjahres den Schülerinnen und Schülern transparent gemacht. Leistungsrückmeldungen können erfolgen

- nach einer mündlichen Überprüfung,
- bei Rückgabe von schriftlichen Leistungsüberprüfungen,
- nach Abschluss eines Projektes,
- nach einem Vortrag oder einer Präsentation,
- bei auffälligen Leistungsveränderungen,
- auf Anfrage,
- als Quartalsfeedback und
- zu Eltern- oder Schülersprechtagen.

Die Leistungsrückmeldung kann

- durch ein Gespräch mit der Schülerin oder dem Schüler,
- durch einen Feedbackbogen,
- durch die schriftliche Begründung einer Note oder

- durch eine individuelle Lern-/Förderempfehlung

erfolgen.

Leistungsrückmeldungen erfolgen auch in der Einführungsphase im Rahmen der kollektiven und individuellen Beratung zur Wahl des Faches Informatik als fortgesetztes Grund- oder Leistungskursfach in der Qualifikationsphase.

Qualitätssicherung und Evaluation

Durch Diskussion der Aufgabenstellung von Klausuren in Fachdienstbesprechungen und eine regelmäßige Erörterung der Ergebnisse von Leistungsüberprüfungen wird ein hohes Maß an fachlicher Qualitätssicherung erreicht.
Es werden jährlich in Fachkonferenzen Erfahrungen ausgetauscht und ggf. Änderungen für den nächsten Durchgang der Einführungsphase beschlossen, um erkannten ungünstigen Entscheidungen schnellstmöglich entgegenwirken zu können.
Außerhalb dieses Curriculums werden ausführliche Modulbeschreibungen entwickelt, die sich auf die aktuell gemachten Unterrichtserfahrungen, dort verwendete Arbeitsblätter und Lernzielkontrollen bzw. auf die entwickelten Klausuren stützen.