

# **Schulinterner Lehrplan zum Kernlehrplan, Lise-Meitner-Gymnasium, Willich**

## **Informatik**

# Inhalt

Seite

<b>1</b>	<b>Rahmenbedingungen der fachlichen Arbeit .....</b>	<b>3</b>
<b>2</b>	<b>Entscheidungen zum Unterricht .....</b>	<b>4</b>
2.1	<i>Übersichtsraster Unterrichtsvorhaben.....</i>	5
	<i>Einführungsphase .....</i>	5
2.2	<i>Konkretisierte Unterrichtsvorhaben .....</i>	6
2.3	Grundsätze der Leistungsbewertung und Leistungsrückmeldung	14
2.4	Lehr- und Lernmittel.....	17

# 1 Rahmenbedingungen der fachlichen Arbeit

Das Lise-Meitner-Gymnasium liegt in Anrath, einer von vier ehemals selbstständigen Gemeinden (Willich, Neersen, Schiefbahn, Anrath), die im Zuge der Kommunalreform 1972 zur Gesamtgemeinde Willich zusammengeschlossen wurden. Im Bewusstsein der Einwohner ist das Bewusstsein der Eigenständigkeit aber immer noch relativ stark ausgeprägt. Dazu trägt auch bei, dass die Verkehrsverbindungen im ÖPNV zwischen den einzelnen Ortsteilen und dem mittlerweile neu hinzugekommenen Ortsteil Wekeln nicht optimal sind.

Die Kommune ist noch stark ländlich geprägt, besitzt aber sehr attraktive Gewerbegebiete. Wegen der Nähe zu den Großstädten Düsseldorf, Köln, Krefeld, Mönchengladbach und zum Ruhrgebiet ist der Anteil der Berufspendler hoch.

Die Kommune war eine der ersten in NRW, die im weiterführenden Schulbereich auf das „Zwei-Säulen-Modell“ gesetzt hat, sodass sich das Angebot an Schulformen auf zwei Gesamtschulen und zwei Gymnasien beschränkt.

Das Lise-Meitner-Gymnasium ist als Reaktion auf die derzeit stark steigende Einwohnerzahl 1998 gegründet worden. 1999 wurde ein neues Schulgebäude bezogen. Die Schule ist eine vierzügige „Halbtagschule“ mit zahlreichen Angeboten in der Über-Mittag-Betreuung. Die Schülerschaft kommt aus allen Willicher Ortsteilen, vornehmlich aus Anrath und Alt-Willich. Der Anteil der Fahrschüler beträgt ca. 60%.

Ab Klasse 8 wird am LMG im Differenzierungsbereich das Fach Informatik (IF) in der Sekundarstufe I angeboten, wobei ein Kernlehrplan des Landes bis zum heutigen Tag nicht existiert. In der Jahrgangsstufe 5 nehmen wir mit einer Klasse am Pilotprojekt Modellvorhaben „Informatik in der Erprobungsstufe eines Gymnasiums“ des Landes NRW teil. In der Oberstufe wird in den Jahrgangsstufen EF bis Q2 das Informatik (IF) als ordentliches Fach im naturwissenschaftlichen Aufgabenfeld seit 2017 auch als Leistungskurs angeboten; es werden in der Regel in den Stufen EF zwei Informatikkurse und in der Q1 und Q2 jeweils ein Informatikkurs mit ca. 20 Schülerinnen und Schülern eingerichtet. In der Sek II besteht eine Kooperation mit der Robert-Schumann-Europaschule in Willich.

Pro Abiturjahrgang entscheiden sich einige Schülerinnen und Schüler für Informatik als Abiturfach, wobei es die überwiegende Zahl als schriftliches Prüfungsfach wählt.

Die Fachgruppe besteht aus drei Fachkollegen, die eine Fakultas für das Fach IF in der Sek I und Sek II besitzen.

Für den Informatikunterricht in der Sek II ist als Lehrwerk Informatik 2 von Schöningh und zusätzlich für den LK Einführung in die Informatik *eingeführt worden*.

## 2 Entscheidungen zum Unterricht

### **Einführungsphase:**

Für den Einstieg in die EF ist es vorgesehen, die Arbeitsweise eines Computers zu verstehen. Ziel ist es zu begreifen, wie man mit Strom rechnen und Informationen darstellen kann.

In der Jahrgangsstufe EF steht nach wie vor das Erlernen einer modernen objektorientierten Programmiersprache im Vordergrund. Diese Programmiersprache (Java) bildet die Basis und das Werkzeug für Beispiele und Aufgaben im gesamten Durchlauf durch die Sek.II und verdeutlicht das objektorientierte Denken sehr konsequent.

Ebenso steht die Modellierung, d.h. das anforderungsspezifische Abbilden von Sachverhalten der realen Welt auf ein Modell, im Vordergrund. Hierbei und bei der Methode der objektorientierten Analyse kommt die Modellierungs- und Dokumentationssprache UML (Unified Modelling Language) mit dem Klassendiagramm zum Einsatz, wobei sowohl Implementations- und Entwurfsdiagramme, als auch die unterschiedlichen Objektbeziehungen „Ist-Beziehung“, „hat-Beziehung“ und kennt-Beziehung die Basis stellen.

### **Qualifikationsphase1:**

In der Jahrgangsstufe Q1 stehen vor allem elementare Algorithmen und Datenstrukturen und deren Umsetzung in Java im Vordergrund. Die Vorstellung der Algorithmen und Datenstrukturen sollte einführend als Kurzreferat erfolgen. Als Beispiele wären vor allem Sortierverfahren auf lineare Strukturen wie z.B. Listen zu nennen und die Datenstrukturen Stapel (Stack) und Warteschlange (Queue), die sich in vielen Anwendungsbereichen (z.B. Druckerwarteschlange) wiederfinden. Daneben werden Grundoperationen und Anwendungssituationen auf verzweigte Baumstrukturen sowie Such- und Verschlüsselungsverfahren behandelt. Darüber hinaus werden im Leistungskurs auch Graphen und komplexe Sortierverfahren behandelt

### **Qualifikationsphase2:**

In der Jahrgangsstufe Q2 werden der Aufbau und die Funktionsweise von Informatiksystemen untersucht. Neben den physikalischen Grundlagen stehen vor allem Protokolle als Kommunikationsvereinbarung in Theorie (ISO-OSI-Schichtenmodell) und Praxis (TCP, IP, Ethernet) im Vordergrund.

Das Inhaltsfeld Datenbanken wird von der Modellierung im ER-Diagramm bis zur normalisierten Datenbank mit Anbindung an eine graphische Benutzerschnittstelle in Java aufbereitet.

Automaten und deren Einordnung und Überführung sind ebenfalls Themen, die im Grundkurs wie im Leistungskurs behandelt werden

## 2.1 Übersichtsraster Unterrichtsvorhaben:

### Übersicht der Abdeckung der Inhaltsfelder und Kompetenzen durch die Unterrichtsvorhaben der EF:

Einführungsphase	Daten und ihre Strukturierung	Algorithmen	Formale Sprachen und Automaten	Informatiksysteme	IF, Mensch und Gesellschaft
Argumentieren	UV3	UV4	UV3	UV1	UV1
Modellieren	UV3	UV4			
Implementieren	UV3	UV2, UV4, UV5	UV2, UV3		UV4
Darstellen und Interpretieren	UV1	UV2	UV2, UV3, UV5	UV1	
Kommunizieren und Kooperieren	UV3	UV4	UV5		

### Übersicht der Abdeckung der Inhaltsfelder und Kompetenzen durch die Unterrichtsvorhaben der Q1 Grundkurs:

Einführungsphase	Daten und ihre Strukturierung	Algorithmen	Formale Sprachen und Automaten	Informatiksysteme	IF, Mensch und Gesellschaft
Argumentieren	UV1	UV3	UV2	UV4	UV4
Modellieren	UV2	UV2			
Implementieren	UV1, UV2, UV3	UV1, UV3	UV1, UV2		UV4
Darstellen und Interpretieren	UV1, UV3	UV1, UV3, UV4	UV2		UV2, UV4
Kommunizieren und Kooperieren	UV1, UV2, UV3	UV3, UV4	UV1, UV2	UV4	UV4

### Übersicht der Abdeckung der Inhaltsfelder und Kompetenzen durch die Unterrichtsvorhaben der Q2<sup>1</sup> Grundkurs:

Einführungsphase	Daten und ihre Strukturierung	Algorithmen	Formale Sprachen und Automaten	Informatiksysteme	IF, Mensch und Gesellschaft
Argumentieren				UV1	UV1
Modellieren			UV2	UV1	
Implementieren	UV2	UV1	UV2	UV1	UV2
Darstellen und Interpretieren	UV2	UV2	UV2	UV1	UV1
Kommunizieren und Kooperieren			UV2	UV2	UV1

<sup>1</sup> Da das Abschlussprojekt UV3 eine Vertiefungsmöglichkeit in der Auswahl der SuS darstellt, erfolgt hier keinerlei Zuordnung der Inhaltsfelder.

## 2.2 Konkretisierte Unterrichtsvorhaben

### 2.2.1 Einführungsphase

#### UV 1: Vom Bit zum Computer. Kann man mit Strom rechnen?

Inhaltsfelder und inhaltliche Schwerpunkte	Kompetenzen	Zeit
<p>Ausgehend von der geschichtlichen Entwicklung der EDV und der wiederholenden, kooperativen Aufbereitung des binären Zahlensystems wird über die Disjunktive Normalform der binären Addition und über die logischen Schaltungen Und (Reihenschaltung), Oder (Parallelschaltung) und Nicht (Transistorschaltung), nachgewiesen, dass mit Strom addiert werden kann. Ebenso soll erfahren werden, dass sich hinter einer binären Codierung diverse Informationen verbergen können (Zeichenketten, negative ganze Zahlen, Fließkommazahlen, Bilder,..).</p> <p>Repräsentierte Inhaltsfelder: Informatiksysteme (prinzipieller Aufbau singulärer Rechnersysteme), Daten und ihre Strukturierung (binäre Darstellung von Daten), Informatik, Mensch und Gesellschaft.</p>	<p>Darstellen und Interpretieren (LogikSim, DNF, binäre Darstellungen) Argumentieren (Erläutern u. Begründen eines Schaltnetzes, ...)</p>	18 Std.

#### UV 2: Syntax und Semantik der Programmiersprache Java

Inhaltsfelder und inhaltliche Schwerpunkte	Kompetenzen	Zeit
<p>Strukturierungsmerkmale und der formale Aufbau der Programmiersprache Java werden an einfachen Quelltextbeispielen in der Entwicklungsumgebung BlueJ analysiert und von den SuS modifiziert. Die Kontrollstrukturen der Programmiersprache und die Verwendung dokumentierter Programmbibliotheken werden anhand einer Turtlegrafik (Aplu.jar) erprobt und eigene Lösungen für gestellte Probleme werden implementiert.</p> <p>Repräsentierte Inhaltsfelder: Formale Sprachen und Automaten, Algorithmen</p>	<p>Implementieren Darstellen und Interpretieren Kooperieren</p>	27 Std.

#### UV 3: Vom Modell zum Programm: Objektorientiertes Modellieren und Objektorientiertes Programmieren in Java

Inhaltsfelder und inhaltliche Schwerpunkte	Kompetenzen	Zeit
<p>Einführung in das OO-Paradigma am gängigen Modell des Onlinebankings. Analyse und Modellierung der Klasse Konto mit den zugehörigen Attributen und Methoden im UML-Klassendiagramm. Implementation der Klasse Konto nach dem Modell. Deklaration und Initialisierung von Variablen. Unterscheidung von Klassen und Objekten durch die Erfassung der Zustände eines Objektes in BlueJ. Einführung in das Geheimnisprinzip. Initialisierung von Objekten in Java</p>	<p>Implementieren Darstellen und Interpretieren Modellieren Kooperieren Argumentieren</p>	36 Std.

<p>und Verwendung unterschiedlicher Konstruktoren. Verwendung von Sammlungsobjekten in Java am Beispiel von Arrays im Rahmen des Lottoprojekts. Einsatz von Dienstklassen und Darstellung der Hat-Beziehung im UML-Klassendiagramm. Einsatz der Vererbung in Java und Darstellung der „ist-Beziehung“ im UML-Klassendiagramm. Repräsentierte Inhaltsfelder: Formale Sprachen und Automaten, Daten und ihre Strukturierung</p>		
---	--	--

**UV 4: „Sortieren geht über Probieren“. Sortierung von Datenmengen: Implementierung naiver Sortier- und Suchverfahren.**

Inhaltsfelder und inhaltliche Schwerpunkte	Kompetenzen	Zeit
<p>Ausgehend vom Problem der Sortierung eines reellen Kartenstapels sollen die SuS kooperativ algorithmische Beschreibungen zur Lösung dieses Problems finden und diese schriftlich fixieren, um später Kontrollstrukturen zur Implementierung zu identifizieren. Die Implementierung erfolgt zunächst am Beispiel des Arrays aus dem Lottoprojekt und wird dann auf größere Datenmengen übertragen. Aufwandsabschätzungen bezüglich Zeitaufwand und Speicherbedarf können dann zunächst auf größeren unsortierten Datenmengen und dann auf entsprechende sortierte Datenmengen erfahren werden. Die gemachten Erfahrungen sollen dann für die Fälle Best-Case, Average-Case und Worst-Case auf die bereits kennengelernten Sortierverfahren übertragen werden. Repräsentierte Inhaltsfelder: Formale Sprachen und Automaten, Algorithmen</p>	<p>Implementieren Darstellen und Interpretieren Kooperieren Argumentieren</p>	<p>18 Std.</p>

**UV 5: „Klick mich“: GUI-Programmierung in Java**

Inhaltsfelder und inhaltliche Schwerpunkte	Kompetenzen	Zeit
<p>Am Beispiel des Projektes Fuchs und Hase (Barnes &amp; Kölling) werden die objektorientierten Begriffe abstrakte Methoden und - Klassen eingeführt und der Begriff der Vererbung wiederholt. Als abstrakteste Klasse wird das Interface im Rahmen der GUI-Programmierung aufgegriffen. Das Java-Listener-Konzept wird nun an einfachen GUI-Beispielen kooperativ implementiert (flüchtiger Button, Bildbetrachter, ...) Repräsentierte Inhaltsfelder: Formale Sprachen und Automaten, Algorithmen</p>	<p>Implementieren Darstellen und Interpretieren Kooperieren Argumentieren</p>	<p>21 Std.</p>

## 2.2.2.1 Qualifikationsphase 1 GK

### UV 1: „Hinten anstellen!“ Algorithmen und Datenstrukturen im Alltag

Inhaltsfelder und inhaltliche Schwerpunkte	Kompetenzen	Zeit
Ausgehend von rekursiven Arbeitsweisen und Strukturen werden an ausgewählten Alltagssituationen (Druckerschlange, Abstellgleis) die linearen Datenstrukturen Stack und Queue (zunächst Arrayimplementationen) und die Datenstruktur List in Kurzreferaten grafisch dargestellt und kooperativ, entsprechend der Dokumentation der Materialien für das Zentralabitur, in Java implementiert. Die Array-Implementationen von Stack und Queue werden durch linear rekursive Datenstrukturen abgelöst. Inhaltsfelder: Formale Sprachen, Daten und ihre Strukturierung, Algorithmen	Implementieren Darstellen und Interpretieren Modellieren Kooperieren Argumentieren	30 Std.

### UV2: Das Kartenspielprojekt

Inhaltsfelder und inhaltliche Schwerpunkte	Kompetenzen	Zeit
Auf der Grundlage der Substantiv-Verb-Methode (Barnes, Kölling Kap. 13) und den der anschließenden Modellierung (Implementations- und Entwurfsdiagramm) wird ein einfaches Kartenspiel beschrieben, modelliert und als Anwendung mit graphischer Darstellung in Kleinstgruppen implementiert und bewertet.	Implementieren Darstellen und Interpretieren Modellieren Kooperieren Argumentieren	40 Std.

### UV3: Sortieren komplex! Binäre Suche und Suchbäume und verzweigt rekursive Lösungsstrategien.

Inhaltsfelder und inhaltliche Schwerpunkte	Kompetenzen	Zeit
Am Problem der Sortierung von Kartenblättern in Kartenspielen werden die naiven Sortierverfahren wiederholt und komplexe verzweigte Lösungsansätze dargestellt. Die Darstellung und Implementation der Klasse BinarySearchTree veranschaulicht dabei in der Methode insert(Object pObject) das Prinzip der binären Suche. Auf der Grundlage der Traversierungsmethoden (inorder, postorder und preorder) werden alle rekursiv möglichen Durchläufe grafisch dargestellt und auf der Grundlage der Methode inorder selbstständig implementiert. Die komplexen Sortierverfahren Quick-Sort und Merge-Sort werden abschließend von den SuS kooperativ graphisch dargestellt, erläutert und hinsichtlich der Effizienz und Komplexität bewertet.	Implementieren Darstellen und Interpretieren Argumentieren Kooperieren	25 Std.



## UV4: Wie werden Informationen sicher? Historische und aktuelle Ansätze der Kryptographie.

Inhaltsfelder und inhaltliche Schwerpunkte	Kompetenzen	Zeit
Ausgehend von aktuellen Problemen der Datensicherheit werden chronologisch zunächst die symmetrischen Verschlüsselungsverfahren nach Cäsar und Vignere dargestellt und anschließend implementiert. Die Beurteilung der Sicherheit der Verschlüsselungsverfahren durch die SuS erfolgt nach der Erprobung von einfachen Brute-Force Methoden zur Entschlüsselung. Die asymmetrischen Verschlüsselungsverfahren insbesondere RSA werden zunächst an einfachen Beispielen erprobt und kooperativ arbeitsteilig in Teilschritten dargestellt. Die SuS untersuchen und bewerten im Anschluss Möglichkeiten dieses Verfahren mit einfachen Möglichkeiten zu entschlüsseln.	Implementieren Darstellen und Interpretieren Argumentieren Kooperieren	25 Std.

### 2.2.2.2 Qualifikationsphase 2 GK

## UV1: Wie können Computer miteinander kommunizieren?

Inhaltsfelder und inhaltliche Schwerpunkte	Kompetenzen	Zeit
Ausgehend vom Versuch innerhalb eines SuS-Netzwerks binär, analog, non-akustisch und non-visuell eine Nachricht von Sender zum Empfänger zu schicken, werden Netzwerktopologien und grundlegende Aufgaben von Protokollen entwickelt. Theoretische Modelle (ISO-OSI-Schichtenmodell) und praktischen Protokollen werden grafisch dargestellt und unterschiedliche Zustände in Graph- bzw. Tabellenform erfasst. Auf der Basis hierauf aufsetzender einfacher Protokolle (POP3) wird ein Protokoll in der für das Zentralabitur vorgesehenen Form implementiert (Client-Server) und auch hinsichtlich der Sicherheit bewertet. SuS entwickeln auf dieser Grundlage eigene Protokolle für unterschiedliche Anwendungssituationen. Bei der Implementation ist die Verwendung von bereits behandelten linearen Datenstrukturen intendiert.	Implementieren Darstellen und Interpretieren Argumentieren Formale Sprachen und Automaten	35 Std.

## UV2: Umgang und Verwaltung von Datenmassen?

Inhaltsfelder und inhaltliche Schwerpunkte	Kompetenzen	Zeit
Die Modellierung und Realisierung einer Schuldatenbank über die Schritte ER-Diagramm, Relationenschema und Datenbankentwurf ist die Grundlage für die Entwicklung von einfachen und komplexeren Abfragen in SQL. Die Anwendung von Normalisierungsformen (1-3) zur Optimierung der Datenbank, sowie die Anbindung dieser an Informationssysteme bildet den Abschluss dieses Unterrichtsvorhabens.	Implementieren Darstellen und Interpretieren Argumentieren Formale Sprachen und Automaten	40 Std.

### UV3: Welche Vorgänge lassen sich automatisieren?

Inhaltsfelder und inhaltliche Schwerpunkte	Kompetenzen	Zeit
Wo liegen die Grenzen der Berechenbarkeit, wie lassen sich Vorgänge automatisieren und wie lassen sich Automaten beschreiben? Zustandsdiagramme Mealy- und Kellerautomat sowie NEA und DEA, sowie das Thema formale Sprachen bis hin zum Compiler sollen die Übergänge von Soft- zur Hardware ausloten und die Arbeitsweise von Programmiersprachen verdeutlichen.	Implementieren Darstellen und Interpretieren Argumentieren Formale Sprachen und Automaten	25 Std.

### 2.2.2.3 Qualifikationsphase 1 LK

#### UV 1: „Hinten anstellen!“ Algorithmen und Datenstrukturen im Alltag

Inhaltsfelder und inhaltliche Schwerpunkte	Kompetenzen	Zeit
Ausgehend von rekursiven Arbeitsweisen und Strukturen werden an ausgewählten Alltagssituationen (Druckerschlange, Abstellgleis) die linearen Datenstrukturen Stack und Queue (zunächst Arrayimplementationen) und die Datenstruktur List in Kurzreferaten grafisch dargestellt und kooperativ, entsprechend der Dokumentation der Materialien für das Zentralabitur, in Java implementiert. Die Array-Implementationen von Stack und Queue werden durch linear rekursive Datenstrukturen abgelöst. Inhaltsfelder: Formale Sprachen, Daten und ihre Strukturierung, Algorithmen	Implementieren Darstellen und Interpretieren Modellieren Kooperieren Argumentieren	35 Std.

#### UV2: Das Kartenspielprojekt

Inhaltsfelder und inhaltliche Schwerpunkte	Kompetenzen	Zeit
Auf der Grundlage der Substantiv-Verb-Methode (Barnes, Kölling Kap. 13) und den der anschließenden Modellierung (Implementations- und Entwurfsdiagramm) wird ein einfaches Kartenspiel beschrieben, modelliert und als Anwendung mit graphischer Darstellung in Kleinstgruppen implementiert und bewertet. Die Verwendung der Datenstrukturen aus UV1 ist hierbei verbindlich.	Implementieren Darstellen und Interpretieren Modellieren Kooperieren Argumentieren	55 Std.

#### UV3: Sortieren komplex! Binäre Suche und Suchbäume und verzweigt rekursive Lösungsstrategien.

Inhaltsfelder und inhaltliche Schwerpunkte	Kompetenzen	Zeit
Am Problem der Sortierung von Kartenblättern in Kartenspielen werden die naiven Sortierverfahren wiederholt und komplexe verzweigte Lösungsansätze dargestellt. Die Darstellung und Implementation der Klasse BinarySearchTree veranschaulicht dabei in der Methoden insert(Object pObject) das Prinzip der binären Suche. Auf der Grundlage der Traversierungsmethoden (inorder, postorder und preorder) werden alle rekursiv möglichen Durchläufe grafisch dargestellt und auf der Grundlage der Methode	Implementieren Darstellen und Interpretieren Argumentieren Kooperieren	35 Std.

inorder selbstständig implementiert. Die komplexen Sortierverfahren Quick-Sort und Merge-Sort werden abschließend von den SuS kooperativ graphisch dargestellt, erläutert, implementiert und hinsichtlich der Effizienz und Komplexität bewertet. Die Implementation der Datenstruktur Graph ist obligatorisch.		
---	--	--

#### **UV4: Wie werden Informationen sicher? Historische und aktuelle Ansätze der Kryptographie.**

Inhaltsfelder und inhaltliche Schwerpunkte	Kompetenzen	Zeit
Ausgehend von aktuellen Problemen der Datensicherheit werden chronologisch zunächst die symmetrischen Verschlüsselungsverfahren nach Cäsar und Vignere dargestellt und anschließend implementiert. Die Beurteilung der Sicherheit der Verschlüsselungsverfahren durch die SuS erfolgt nach der Erprobung von einfachen Brute-Force Methoden zur Entschlüsselung. Die asymmetrischen Verschlüsselungsverfahren insbesondere RSA werden zunächst an einfachen Beispielen erprobt und kooperativ arbeitsteilig in Teilschritten dargestellt. Die SuS untersuchen und bewerten im Anschluss Möglichkeiten dieses Verfahren mit einfachen Möglichkeiten zu entschlüsseln. RSA oder Diffie & Hellmann sollen in arbeitsteiligen Schritten implementiert werden.	Implementieren Darstellen und Interpretieren Argumentieren Kooperieren	35 Std.

#### **2.2.2.4 Qualifikationsphase 2 LK**

#### **UV1: Wie können Computer miteinander kommunizieren?**

Inhaltsfelder und inhaltliche Schwerpunkte	Kompetenzen	Zeit
Ausgehend vom Versuch innerhalb eines SuS-Netzwerks binär, analog, non-akustisch und non-visuell eine Nachricht von Sender zum Empfänger zu schicken, werden Netzwerktopologien und grundlegende Aufgaben von Protokollen entwickelt. Theoretische Modelle (ISO-OSI-Schichtenmodell) und praktischen Protokollen werden grafisch dargestellt und unterschiedliche Zustände in Graph- bzw. Tabellenform erfasst. Auf der Basis hierauf aufsetzender einfacher Protokolle (POP3) wird ein Protokoll in der für das Zentralabitur vorgesehenen Form implementiert (Client-	Implementieren Darstellen und Interpretieren Argumentieren Formale Sprachen und Automaten	40 Std.

Server) und auch hinsichtlich der Sicherheit bewertet. SuS entwickeln auf dieser Grundlage eigene Protokolle für unterschiedliche Anwendungssituationen. Bei der Implementation ist die Verwendung von bereits behandelten linearen Datenstrukturen intendiert.		
---	--	--

## UV2: Umgang und Verwaltung von Datenmassen?

Inhaltsfelder und inhaltliche Schwerpunkte	Kompetenzen	Zeit
Die Modellierung und Realisierung einer Schuldatenbank über die Schritte ER-Diagramm, Relationenschema und Datenbankentwurf ist die Grundlage für die Entwicklung von einfachen und komplexeren Abfragen in SQL. Die Anwendung von Normalisierungsformen (1-3) zur Optimierung der Datenbank, sowie die Anbindung dieser an Informationssysteme bildet den Abschluss dieses Unterrichtsvorhabens.	Implementieren Darstellen und Interpretieren Argumentieren Formale Sprachen und Automaten	40 Std.

## UV3: Welche Vorgänge lassen sich automatisieren?

Inhaltsfelder und inhaltliche Schwerpunkte	Kompetenzen	Zeit
Wo liegen die Grenzen der Berechenbarkeit, wie lassen sich Vorgänge automatisieren und wie lassen sich Automaten beschreiben? Zustandsdiagramme Mealy- und Kellerautomat sowie NEA und DEA, sowie das Thema formale Sprachen bis hin zum Compiler sollen die Übergänge von Soft- zur Hardware ausloten und die Arbeitsweise von Programmiersprachen verdeutlichen.	Implementieren Darstellen und Interpretieren Argumentieren Formale Sprachen und Automaten	35 Std.

## 2.3 Grundsätze der Leistungsbewertung und Leistungsrückmeldung

Auf der Grundlage von § 48 SchulG, APO-GOst sowie Kapitel 3 des Kernlehrplans Informatik hat die Fachkonferenz im Einklang mit dem entsprechenden schulbezogenen Konzept die nachfolgenden Grundsätze zur Leistungsbewertung und Leistungsrückmeldung beschlossen.

### Verbindliche Absprachen:

1. In der EF soll im ersten Halbjahr nur eine 90-minütige Klausur durchgeführt und im Plenum ausgewertet werden. Im zweiten Halbjahr werden zwei 90-minütige Klausuren geschrieben.
2. In der Q1 werden pro Halbjahr im GK zwei zweistündige Klausuren geschrieben. Im Leistungskurs werden je zwei dreistündige Klausuren geschrieben.
3. In der Q2 werden im ersten Halbjahr zwei dreistündige Klausuren geschrieben und im LK jeweils zwei 4,25 Stunden dauernde Klausuren. Die Vorabiturklausuren werden unter Abiturbedingungen geschrieben.
4. Ein (kleineres) Programmierprojekt soll im Rahmen jedes Jahrgangs der Sek. 2 durchgeführt werden.

Die Grundsätze der Leistungsfeststellung werden den Schülerinnen und Schülern (zum Schuljahresbeginn) sowie den Erziehungsberechtigten (u.a. im Rahmen des Elternsprechtages und der Jahrgangsstufenpflegschaftssitzungen) transparent gemacht und erläutert. Sie finden Anwendung im Rahmen der grundsätzlichen Unterscheidung von Lern- und Leistungssituationen, die ebenfalls im Unterrichtsverlauf an geeigneter Stelle transparent gemacht wird.

### Verbindliche Instrumente:

#### *Überprüfung der schriftlichen Leistung*

- Im 1. Halbjahr der Einführungsphase wird lediglich eine Klausur zur Überprüfung der schriftlichen Leistung geschrieben.
- Das Format der Aufgaben des schriftlichen Abiturs wird schrittweise entwickelt und schwerpunktmäßig eingeübt.

#### *Überprüfung der sonstigen Leistung*

Neben den o. g. obligatorischen Formen der Leistungsüberprüfung werden weitere Instrumente der Leistungsbewertung genutzt, u. a.:

- mündliche Beiträge zum Unterricht (z. B. Beiträge zum Unterrichtsgespräch, Referate, Präsentationen, Kurzvorträge)
- Mitarbeit in Partner- und Gruppenarbeiten
- projektbezogenes Arbeiten (Quelltexte, Ergebnisprotokolle, Dokumentationen, UML-Diagramme)
- weitere schriftliche Beiträge zum Unterricht (z. B. Protokolle, Materialsammlungen, Hefte/Mappen)
- Beiträge im Rahmen eigenverantwortlichen, schüleraktiven Handelns (z. B. Präsentation, Rollenspiel)

### Übergeordnete Kriterien:

Die Bewertungskriterien für eine Leistung werden den Schülerinnen und Schülern zu Beginn der jeweiligen Kurshalbjahre transparent gemacht. Die folgenden – an die Bewertungskriterien des Kernlehrplans für die Abiturprüfung angelehnten – allgemeinen Kriterien gelten sowohl für die schriftlichen als auch für die sonstigen Formen der Leistungsüberprüfung:

- Umfang und Differenzierungsgrad der Ausführungen
- sachliche Richtigkeit und Schlüssigkeit der Ausführungen
- Angemessenheit der Abstraktionsebene
- Herstellen geeigneter Zusammenhänge
- argumentative Begründung eigener Urteile, Stellungnahmen und Wertungen
- Eigenständigkeit der Auseinandersetzung mit Sachverhalten und Problemstellungen
- Klarheit und Strukturiertheit in Aufbau von Darstellungen
- Sicherheit im Umgang mit Fachmethoden
- Verwendung von Fachsprache und geklärter Begrifflichkeit
- Erfüllung standardsprachlicher Normen

Der Grad der Anwendung der angeführten Maßstäbe hängt insgesamt von der Komplexität der zu erschließenden und darzustellenden Gegenstände ab.

### Konkretisierte Kriterien:

#### *Kriterien für die Bewertung der schriftlichen Leistung*

Die Bewertung der schriftlichen Leistungen, insbesondere von Klausuren, erfolgt anhand von jeweils zu erstellenden Bewertungsrastern (Erwartungshorizonte), die sich an den Vorgaben für die Bewertung von Schülerleistungen im Zentralabitur orientieren.

### *Kriterien für die Überprüfung der sonstigen Leistungen*

- inhaltliche Qualität und gedankliche Stringenz der Beiträge
- Selbständigkeit der erbrachten Leistung
- Bezug der Beiträge zum Unterrichtsgegenstand
- Verknüpfung der eigenen Beiträge mit bereits im Unterricht erarbeiteten Sachzusammenhängen sowie mit den Beiträgen anderer Schülerinnen und Schüler
- funktionale Anwendung fachspezifischer Methoden
- sprachliche und fachterminologische Angemessenheit der Beiträge
- Qualität und Selbständigkeit der Implementation in Java

### Grundsätze der Leistungsrückmeldung und Beratung:

Die Leistungsrückmeldung erfolgt in mündlicher Form.

- **Intervalle**
  - punktuelles Feedback auf im Unterricht erbrachte spezielle Leistungen
  - Quartalsfeedback
- **Formen**
  - Einstufung der Beiträge im Hinblick auf den deutlich werdenden Kompetenzerwerb,
  - individuelle Lern-/Förderempfehlungen in der EF bei Versetzungsgefährdung
  - Beratung am Eltern- oder Schülersprechtag



## 2.4 Lehr- und Lernmittel:

BARNES, D.J. KÖLLING, M.: Java lernen mit *BlueJ*: Eine Einführung in die objektorientierte Programmierung. Pearson Studium. München 2007.

DRESCH, P.; FROBEL G.; KOSCHORREK, H.J.: Informatik 2, Schöningh. ISBN: 978-3-14-037127-8

KEMPE, T.; LÖHR A.; TEPÄßE, D.: Informatik 1, Schöningh. ISBN: 978-3-14-037126-1

HEROLD, H.; LURZ, B.; WOHLRAB, J.: Grundlagen der Informatik. Praktisch – Technisch – Theoretisch. Pearson Studium. München 2007.

KEMPER, A.; Eickler, A.: Datenbanken. ISBN: 978-3486257069