

Schulinterner Lehrplan  
des Freiherr-vom-Stein-Gymnasiums  
im Fach Informatik  
für die Einführungsphase

## **Erstes Unterrichtsvorhaben (EF.I):**

**Thema:** Einführung in Informatiksysteme

**Leitfragen:** Was ist Informatik und wie funktionieren Computer(systeme)? (I)

### **Inhaltsfelder:**

- Informatiksysteme
- Informatik, Mensch und Gesellschaft

### **Inhaltliche Schwerpunkte:**

- Einzelrechner
- Dateisystem
- Einsatz von Informatiksystemen

### **Zu entwickelnde Kompetenzen:**

*Leitkompetenzen:* Argumentieren, Darstellen/Interpretieren, Kommunizieren/Kooperieren  
Die Schülerinnen und Schüler

- erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (A).
- nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung und gemeinsamen Verwendung von Daten unter Berücksichtigung der Rechteverwaltung (K).
- nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D)
- nutzen das Internet zu Recherche, Datenaustausch und Kommunikation (K).
- bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A).
- stellen ganze Zahlen und Zeichen in Binärcodes dar (D),
- interpretieren Binärcodes als Zahlen und Zeichen (D).
- beschreiben und erläutern den strukturellen Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der „Von-Neumann-Architektur“ (A)

### **Mögliche konkrete Realisierung und Sequenzierung des Unterrichtsvorhabens:**

1. Was ist Informatik?
2. Einführung in das schuleigene Computersystem und dessen Ordnerablage, Training einer sinnvollen Dateiablage, Besprechung der Nutzerordnung für die Computerräume
3. Informatik in Vergangenheit, Gegenwart und Zukunft: Internetrecherche und Diskussion über Chancen und Risiken der Digitalisierung
4. Binärcodierung
5. Verzweigungen als Neuerung der von-Neumann-Architektur: Anwendung am Know-How-Computer

**Zeitbedarf:** ca. 3 Stunden (= 1 Woche)

## Zweites Unterrichtsvorhaben (EF.I):

**Thema:** Einführung in die Objektorientierte Programmierung

**Leitfragen:** Wie kommt die Welt in den Rechner? / Wie programmiert man ein erstes Spiel?

### Inhaltsfelder:

- Daten und ihre Strukturierung
- Formale Sprachen und Automaten
- Algorithmen

### Inhaltliche Schwerpunkte:

- Objekte und Klassen
- Syntax und Semantik einer Programmiersprache

### Zu entwickelnde Kompetenzen:

*Leitkompetenzen:* Modellieren, Implementieren, Kommunizieren und Kooperieren  
Die Schülerinnen und Schüler

- ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen und Objekttypen zu (M)
- stellen den Zustand eines Objekts dar (D)
- stellen die Kommunikation zwischen Objekten und Klassen-, Assoziations- und Vererbungsbeziehungen in Diagrammen graphisch dar (M, D)
- ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M)
- analysieren und erläutern eine objektorientierte Modellierung (A),
- modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen sowie unter Verwendung von Vererbung (M)
- ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M)
- implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I)
- dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D)
- implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I)
- stellen die Kommunikation zwischen Objekten und Klassen-, Assoziations- und Vererbungsbeziehungen in Diagrammen graphisch dar (D)
- analysieren und erläutern eine objektorientierte Modellierung (A)
- implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I)
- interpretieren Fehlermeldungen des Compilers und korrigieren den Quellcode (I)

### Mögliche konkrete Realisierung und Sequenzierung des Unterrichtsvorhabens:

1. Unterschiede und Gemeinsamkeiten zwei verschiedener Stühle -> Objekte, Klassen, Vererbung
2. UML-Diagramme der Klasse „Stuhl“ sowie anderer Klassen mit Datentypen in Java
3. erste einfache Implementierungen in Greenfoot oder BlueJ (Klassendeklaration, Konstruktor, Attribute, getter- und setter-Methoden, Methodenaufrufe, Variablen und Wertzuweisungen), z.B. Projekte *Schule*, *Shapes*
4. komplexere Implementierungen und Modellierungen von Algorithmen mit Kontrollstrukturen (if-statements, for- und while-Schleifen) und Zufallszahlen, z.B. Projekte *Wombats and Leaves*, *Schatzräuber*, *Schiebepuzzle*

**Zeitbedarf:** ca. 54 Stunden (= ca. 18 Wochen)

(Stand: Oktober 2023)

## Drittes Unterrichtsvorhaben (EF.II):

**Thema:** Arrays

**Leitfragen:** Wie kann man mehrere Daten desselben Typs effizient speichern und abrufen?

### Inhaltsfelder:

- Daten und ihre Strukturierung
- Formale Sprachen und Automaten
- Algorithmen

### Inhaltliche Schwerpunkte:

- Objekte und Klassen
- Syntax und Semantik einer Programmiersprache

### Zu entwickelnde Kompetenzen:

*Leitkompetenzen:* Darstellen und Interpretieren, Modellieren, Implementieren, Kommunizieren und Kooperieren

Die Schülerinnen und Schüler

- stellen die Kommunikation zwischen Objekten und Klassen- und Vererbungsbeziehungen in Diagrammen graphisch dar (D)
- implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I)
- dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D)
- implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I)
- stellen die Kommunikation zwischen Objekten und Klassen-, Assoziations- und Vererbungsbeziehungen in Diagrammen graphisch dar (D)
- ordnen Attributen, Parametern und Rückgaben von Methoden (statische) lineare Datensammlungen zu.
- analysieren und erläutern einfache Algorithmen und Programme (A)
- modifizieren einfache Algorithmen und Programme (I)
- entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M)
- implementieren Algorithmen und stellen sie umgangssprachlich und grafisch dar (M)
- interpretieren Fehlermeldungen des Compilers und korrigieren den Quellcode (I)
- testen Programme schrittweise anhand von Beispielen (I)

### Mögliche konkrete Realisierung und Sequenzierung des Unterrichtsvorhabens:

1. Kennenlernen von Arrays als statische lineare Datensammlung, z.B. zum Speichern aller Namen der Kursteilnehmer:innen
2. Kennenlernen der wesentlichen Eigenschaften der Datenstruktur Array (vorgegebene Länge, vorgegebener Daten- bzw. Objekttyp, feste Positionszuweisung etc.)
3. Deklaration und Initialisierung von Arrays und Implementation elementarer Operationen (löschen, tauschen, suchen):
  - a) mit einem primitiven Datentyp (z.B. Projekte *Zahlenliste* oder *Würfelbecher* mit dem Datentyp `int`)
  - b) mit dem komplexen Datentyp `String` und Kennenlernen der grundlegenden Methoden der Klasse `String` (z.B. Projekt *Wortsalat*)
  - c) mit einem komplexen Objekttyp (z.B. Projekt *Telefonbuch*)

**Zeitbedarf:** ca. 33 Stunden (= ca. 11 Wochen)

## **Viertes Unterrichtsvorhaben (EF.II):**

**Thema:** Suchen und Sortieren

**Leitfragen:** Wie kann man Daten sortieren, z.B. um ausgewählte Daten schnell zu finden?

**Inhaltsfeld:**

- Algorithmen

**Inhaltliche Schwerpunkte:**

- Algorithmen zum Suchen und Sortieren

**Zu entwickelnde Kompetenzen:**

*Leitkompetenzen:* Darstellen und Interpretieren, Modellieren, Argumentieren, Kommunizieren und Kooperieren

Die Schülerinnen und Schüler

- entwerfen einen weiteren Algorithmus zum Sortieren (M)
- analysieren Such- und Sortieralgorithmen und wenden sie auf Beispiele an (D)
- beurteilen die Effizienz von Algorithmen am Beispiel von Sortierverfahren hinsichtlich Zeitaufwand und Speicherplatzbedarf (A)
- testen Programme schrittweise anhand von Beispielen (I)

**Mögliche konkrete Realisierung und Sequenzierung des Unterrichtsvorhabens:**

1. Sortieren von Karten → Erkennen des Unterschieds Sortiervorgang Computer / Mensch
2. Entwicklung eines Sortier-Algorithmus
3. Kennenlernen und Beurteilung der Algorithmen Selection Sort / Min-Sort und Bubble-Sort

**Zeitbedarf:** ca. 18 Stunden (= ca. 6 Wochen)

## **Fünftes Unterrichtsvorhaben (EF.II):**

**Thema:** Funktionsweise und Geschichte von Informatiksystemen

**Leitfragen:** Was ist Informatik und wie funktionieren Computer(systeme)? (II)

### **Inhaltsfelder:**

- Informatiksysteme
- Informatik, Mensch und Gesellschaft

### **Inhaltliche Schwerpunkte:**

- Geschichte der automatischen Datenverarbeitung
- Digitalisierung
- Internet
- Einsatz von Informatiksystemen

### **Zu entwickelnde Kompetenzen:**

*Leitkompetenzen:* Argumentieren, Darstellen/Interpretieren, Kommunizieren/Kooperieren  
Die Schülerinnen und Schüler

- erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (A).
- nutzen das Internet zu Recherche, Datenaustausch und Kommunikation (K).
- bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A).
- stellen ganze Zahlen und Zeichen in Binärcodes dar (D),
- interpretieren Binärcodes als Zahlen und Zeichen (D).
- beschreiben und erläutern den strukturellen Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der „Von-Neumann-Architektur“ (A)

### **Mögliche konkrete Realisierung und Sequenzierung des Unterrichtsvorhabens:**

1. Informatik in Vergangenheit, Gegenwart und Zukunft: Internetrecherche und Diskussion über Chancen und Risiken der Digitalisierung
2. Binärcodierung
3. Verzweigungen als Neuerung der von-Neumann-Architektur: Anwendung am Know-How-Computer

**Zeitbedarf:** ca. 6 Stunden (= 2 Wochen)

## **Weitere mögliche Unterrichtsvorhaben (EF.II):**

### **1. Graphische Benutzeroberflächen (GUIs)**

Die SchülerInnen lernen Textfelder, Eingabefelder und Knöpfe als wesentliche Bestandteile einer GUI (graphical user interface) kennen. Sie implementieren eine kreativ nach ihren eigenen Vorstellungen funktionierende GUI, z.B. eine, die bei Eingabe eines Vor- und Nachnamens automatisch eine E-Mail-Adresse generiert, oder aber eine, die bei Eingabe eines Filmtitels Informationen zum Film aufzeigt etc.

### **2. Implementation eines Spiels**

Beispiel: Hangman

# Leistungsbemessungskonzept für das Unterrichtsfach Informatik in der Einführungsphase

Das Fach Informatik kann in der Oberstufe schriftlich oder mündlich als Grundkurs gewählt werden.

Bei schriftlicher Belegung in der Einführungsphase schreibt man in jedem Halbjahr genau eine 90minütige Klausur, nämlich im ersten Halbjahr diejenige zum zweiten Klausurtermin und im zweiten Halbjahr diejenige zum ersten Klausurtermin.

Es gilt wie im Zentralabitur der folgende Bewertungsschlüssel:

Punkte	Prozentzahl
15	ab 95%
14	ab 90%
13	ab 85%
12	ab 80%
11	ab 75%
10	ab 70%
9	ab 65%
8	ab 60%
7	ab 55%
6	ab 50%
5	ab 45%
4	ab 40%
3	ab 34%
2	ab 27%
1	ab 20%

In der Einführungsphase werden die Punkte folgendermaßen in Noten umgerechnet:

15-13 Punkte → *sehr gut*; 12-10 Punkte → *gut* usw.

Die **Sonstige Mitarbeitsnote** basiert auf der Qualität, Quantität und Kontinuität der mündlichen Mitarbeit, der Arbeit in Partner- und Gruppenarbeitsphasen sowie auf der Bewertung der anderen im Unterricht angefertigten und vorgestellten (Projekt-) Arbeiten. Letztere werden prozessbegleitend bewertet, nicht allein das Endergebnis ist entscheidend. Weiterhin fließen die Ergebnisse kurzer schriftlicher Lernzielkontrollen in die Note für die Sonstige Mitarbeit mit ein.

Wird Informatik schriftlich gewählt, so werden die Klausurergebnisse und die Sonstige Mitarbeitsnote gleichwertig zu einer Kursnote zusammengefasst; wird Informatik mündlich gewählt, ergibt die Note für die Sonstige Mitarbeit zugleich die Kursnote.