

# LEHRPLAN FÜR DAS ERGÄNZUNGSFACH INFORMATIK

## A. Stundendotation

Klasse	1.	2.	3.	4.
Wochenstunden				4

## B. Didaktische Konzeption

### (1) Beitrag des Faches zur gymnasialen Bildung

Informatik ist aus unserem Alltag nicht wegzudenken: ohne Informatik keine Informationsgesellschaft. Die Lernenden erhalten im Ergänzungsfach Informatik einen Einblick in das Wesen der Informatik und in die Tätigkeit von Informatiker und Informatikerinnen. Dabei lernen sie einerseits theoretische Aspekte der Informatik kennen, andererseits werden konkrete Fähigkeiten der Lösung von Informatikproblemen vermittelt. Das Ergänzungsfach soll dabei insbesondere mit dem Vorurteil aufräumen, Informatik sei lediglich Programmieren, und ein Gefühl für die Vielfalt der Informatik vermitteln.

### (2) Überfachliche Kompetenzen

Das Ergänzungsfach Informatik fördert besonders

#### Reflexive Fähigkeiten

- Probleme analysieren, strukturieren und lösen

#### Sozialkompetenz

- Zu Team- und Projektarbeit bereit sein

#### Sprachkompetenz

- Lösungsstrategien unter Verwendung der korrekten Fachbegriffe mündlich und schriftlich verständlich formulieren

#### Praktische Fähigkeiten und IKT-Kompetenzen

- Das Internet zur gezielten Suche nach geeigneten Informationen zur Unterstützung bei der Realisierung von eigenen Lösungen nutzen

#### Interessen

- Sich mit den Einsatzmöglichkeiten der Informatik und deren Auswirkungen im Alltag auseinandersetzen
- Bereitschaft, Informatiklösungen in verschiedenen Anwendungsgebieten genau zu analysieren und zu hinterfragen

## C. Klassen-Lehrpläne

Die Lerngebiete sind verbindlich. Die Grobinhalte und fachlichen Kompetenzen sind Pflicht-Optionen, aus denen die Lehrpersonen die definitive Auswahl treffen.

### 4. Klasse

#### 1. Lerngebiet: Grundlagen und Konzepte

Grobinhalte	Fachliche Kompetenzen
<ul style="list-style-type: none"><li>• Theoretische Informatik (z.B. endliche Automaten, reguläre Ausdrücke, Graphen, Grenzen der Berechenbarkeit)</li><li>• Informationsrepräsentation (z.B. Informationsgehalt, Komprimierung, Kodierung)</li><li>• Netzwerke (z.B. Netzwerkkarten, Routing, Protokolle der TCP/IP-Suite, synchrone vs. asynchrone Kommunikation)</li><li>• Logik (z.B. Boole'sche Algebra, Aussagenlogik, Prädikatenlogik)</li><li>• Digitaltechnik und Rechnerstruktur (z.B. elementare Schaltalgebra, Aufbau eines Prozessors, Logikschaltungen)</li></ul>	<p>Die Schülerinnen und Schüler können</p> <ul style="list-style-type: none"><li>• aus den behandelten Themen die fundamentalen Konzepte in eigenen Worten erklären.</li><li>• aus den behandelten Themen Probleme lösen.</li></ul>

#### 2. Lerngebiet: Information

Grobinhalte	Fachliche Kompetenzen
<ul style="list-style-type: none"><li>• Datenstrukturen (z.B. elementare Datentypen, strukturierte und dynamische Datentypen, Bäume)</li><li>• Datenbanken (z.B. Entity-Relationship-Modell, Skalierbarkeit, Effizienz)</li><li>• Datensicherheit (z.B. fehlererkennende und fehlerkorrigierende Codes, Prüfsummen)</li></ul>	<p>Die Schülerinnen und Schüler können</p> <ul style="list-style-type: none"><li>• für einfache Problemstellungen angepasste Datenmodelle entwerfen.</li><li>• gegebene komplexe Datenmodelle verstehen, erklären und anwenden.</li><li>• aus den behandelten Themen die fundamentalen Konzepte in eigenen Worten erklären.</li><li>• aus den behandelten Themen Probleme lösen.</li></ul>

### 3. Lerngebiet: Automation

Grobinhalte	Fachliche Kompetenzen
<ul style="list-style-type: none"><li>• Algorithmen aus ausgewählten Bereichen (z.B. Shortest Path, Minimum Spanning Tree, Minimax, MD4/5, RSA, LZW, Scheduling, Paging)</li><li>• Programmieren, Lösen von angemessenen Aufgabenstellungen mit gegebenen Werkzeugen, ev. exemplarische Implementierung von Algorithmen (z.B. Sortieren, Rekursion, Suchbäume, Client-Server-Applikationen)</li></ul>	<p>Die Schülerinnen und Schüler können</p> <ul style="list-style-type: none"><li>• aus den behandelten Themen die fundamentalen Konzepte in eigenen Worten erklären.</li><li>• Probleme strukturieren und sinnvolle Lösungswege formulieren.</li><li>• ein Problem mit einer Programmiersprache lösen.</li><li>• ein gegebenes Programm verstehen, auf Korrektheit prüfen, erweitern oder gegebene Programmteile vervollständigen.</li></ul>