



## EF INFORMATIK

Die nachfolgend beschriebenen Kurse wurden im Laufe der letzten beiden Schuljahre durchgeführt. Sie haben sich bewährt und sind deshalb ins Curriculum aufgenommen worden. Trotzdem muss damit gerechnet werden, dass nicht alle diese Kurse in Zukunft wieder angeboten werden. Es können auch neue Kurse dazukommen. Jeder Kurs dauert ein Semester.

### 1 PROFIL DES FACHES

*Welche Voraussetzungen sind für das Ergänzungsfach Informatik notwendig?*

Die einzige Voraussetzung ist das Interesse an Informationstechnologien. Vorwissen ist für die Kurse des Ergänzungsfaches Informatik nicht notwendig.

*Was bietet das Ergänzungsfach Informatik?*

Das Ergänzungsfach Informatik bietet jede Menge Abwechslung: Selbstständiges Recherchieren, Theorielektionen, Projektarbeit, etc.

In zwei Semestern werden grundlegende Konzepte der Informatik behandelt. Neben Programmieren lernst du einiges über die Funktionsweise des Internets, über Datenbanken, Roboter und vieles mehr. Es geht nicht darum, vertiefte Kenntnisse über die Anwendung von einzelnen Programmen zu erlangen, sondern um die Hintergründe, wie und warum diese funktionieren.

### 2 INTERAKTIVE WEBANWENDUNGEN UND DATENMANAGEMENT

*Ziele*

- Wichtige Konzepte der Webentwicklung verstehen
- Übersicht über Architekturen von modernen Webanwendungen
- Grundlegende Kenntnisse über Datenbanksysteme

*Inhalte*

- Konzeptionelle Unterschiede zwischen statischen und dynamischen Webinhalten
- Verstehen zentraler Konzepte der modernen Webentwicklung (Frontend & Backend)
- Frontend-Entwicklung (Benutzerschnittstelle)
  - Dynamische Inhalte mit JavaScript (DOM-Manipulation, Events, Fetch API)
  - Einführung in ein modernes Frontend-Framework (z. B. React oder Vue.js)
- Backend-Entwicklung (Serverlogik)
  - Unterschiede serverseitiger Sprachen kennen lernen: PHP, Node.js, Python
  - REST-Architektur: Endpoints, Statuscodes, JSON als Austauschformat

- Technische Umsetzung von dynamischen Webinhalten mit einer geeigneten Sprache (PHP, Java, etc.)
- Datenbankmanagement und Modellierung
  - Grundlagen relationaler Datenbanken (z. B. MySQL, PostgreSQL)
  - Datenbankdesign, Normalisierung (1NF–3NF), Entity-Relationship-Modell
  - Zugriff auf Datenbanken im Backend: SQL
- Ein dynamische Webanwendung mit Datenbankanbindung in Teamarbeit planen und umsetzen

### 3 TECHNISCHE GRUNDLAGEN DES INTERNETS, CYBERSECURITY UND KRYPTOGRAPHIE

#### *Ziele*

- Grundlegende Kenntnisse über den technischen Aufbau von Netzwerken
- Übersicht über verschiedenen Bedrohungen des Internets
- Vertiefung der Kenntnisse zu kryptographischen Verfahren

#### *Inhalte*

- TCP/IP - Referenzmodell
- Grundlegende Planung eines Netzwerks sowie Übungen zu den Themen Switching, Routing, Firewalls und DMZ
- Funktionsweisen häufiger Angriffe aus dem Internet, Möglichkeiten der Prävention (Trojaner, Ransomware, Phishing, Man-in-the-Middle-Angriffe, etc.) sowie historischer Rückblick
- Virtuelle Lab-Umgebungen für Simulationen und «Capture the Flag» Übungen zur Schwachstellenanalyse
- Häufig angewendete Compliance-Vorschriften sowie einfache Massnahmen zur Sicherstellung von Daten- und Informationssicherheit
  - Aktuelle Datenschutzgesetze (DSGVO / nDSG CH)
  - Ethik der Cybersecurity (z. B. Hacking, Whistleblowing, „White Hat vs. Black Hat“)
- Vertiefung von Kenntnissen zur symmetrischen und asymmetrischen Kryptographie sowie praktischen Beispielen:
  - Hashfunktionen – z. B. SHA-256, Anwendungen in digitalen Signaturen und Blockchain
  - Digitale Signaturen und Zertifikate – Funktionsweise und Anwendung (z. B. SSL/TLS)
  - Public-Key-Infrastruktur (PKI) – Aufbau und praktische Bedeutung
  - Quantenkryptographie / Post-Quantum-Kryptographie – Zukunftstrends

## 4 MIKROCONTROLLER PROGRAMMIERUNG, AUTOMATENTHEORIE UND DIGITALE LOGIK

### *Ziele*

- Einführung in die Architektur und das Design von Mikrocontrollersystemen
- Übersicht über die Grundlagen der digitalen Logik
- Kurze Einführung in die Automatentheorie, insbesondere endliche Automaten
- Anwendung der erlernten Konzepte in einem Hardwareprojekt

### *Inhalte*

- Unterschied zwischen einem Desktop-PC und einem Mikrocontroller
  - Prozessorarchitekturen und Befehlssätze
  - Vor- und Nachteile der spezifischen Plattformen
- Digitale Logik anhand von Logikgattern wie NOT, AND, OR, NAND, NOR, XOR sowie zugehörige Termumformungen und Vereinfachungen
- Konzept der endlichen Automaten (FSM, Finite State Machines) und Implementierung von einfachen Beispielprogrammen
- Laufendes Hardwareprojekt mit einem Mikrocontrollerboard zur praktischen Veranschaulichung der Inhalte

## 5 FORTGESCHRITTENE PROGRAMMIERKONZEPTE

### *Ziele*

- Übersicht über grundlegende Programmierkonzepte in weiteren Programmiersprachen (C++, Java)
- Erlernen des Konzepts der objektorientierten Programmierung
- Vertiefung der Kenntnisse anhand von sogenannten Design Patterns
- Einführung in die Programmierung von grafischen Benutzeroberflächen

### *Inhalte*

- Prinzipien der objektorientierten Programmierung und Aufzeigen des Unterschieds zum prozeduralen Programmierstil
  - Klassen, Attribute, Datenkapselung
  - Klassenhierarchie und Vererbung
  - Abstraktionskonzept – Unterschied zwischen Objekten und Klassen
- Fluss-, Strukto- und UML-Diagramme zur Darstellung von Abläufen sowie zur Dokumentation von Klassen
- Ausgewählte Design Patterns aus den Bereichen Creational, Structural und Behavioral Design
- Grundlagen der grafischen Benutzeroberflächen mithilfe einer Standardbibliothek wie Qt