

AUTONOME PROVINZ BOZEN-SÜDTIROL

PROVINCIA AUT. DI BOLZANO-ALTO ADIGE

**REALGYMNASIUM  
SPRACHENGYMNASIUM  
TECHNOLOGISCHE FACHOBERSCHULE**



**LICEO SCIENTIFICO  
LICEO LINGUISTICO  
ISTITUTO TECNOLOGICO**

*“J. Ph. Fallmerayer“*

---

39042 Brixen/Bressanone, Dantestraße/Via Dante 39/E

☎ 0472/830893/Fax: 0472/837740  
info@fallmerayer.it

Str. Nr. /Cod. fisc.: 81006290217

## **Fachcurriculum**

### **4. Klasse**

### **TFO**

# **Technologien und Planung von informatischen Systemen und Telekommunikationssystemen**

### **Erziehungs- und Unterrichtsziele (Auszug aus RRL)**

Im Unterricht des Faches Technologien und Planung von informatischen Systemen und Telekommunikationssystemen setzen sich die Schülerinnen und Schüler vertieft mit der Realisierung von Anwendungen für die Netzwerkkommunikation auseinander und lernen die Entwicklung von Software und deren technologische Komponenten zu planen sowie Rechnersysteme und Netzwerke zu installieren, zu konfigurieren und zu verwalten. Die Schülerinnen und Schüler arbeiten an den einzelnen Phasen eines Produktionsprozesses, von der Idee bis zur Realisierung des Projektes mit und können dabei auf die eigene Spezialisierung Bezug nehmen. In den einzelnen Projektphasen benutzen die Schülerinnen und Schüler die nötigen Planungs-, Dokumentations- und Kontrollinstrumente. Dabei wird auf Effizienz, Optimierung und Qualität und auch auf die korrekte Verwendung der spezifischen Fachbegriffe und der Fachsprache geachtet. Die Schülerinnen und Schüler erfahren die Bedeutung eines zielorientierten Arbeitens und die Notwendigkeit, Verantwortung zu übernehmen. Sie beachten die Rechtsvorschriften, welche den spezifischen Bereich regeln und schenken der Arbeitssicherheit und dem Schutz der Person, der Umwelt und des Territoriums besondere Aufmerksamkeit.

### **Kompetenzen am Ende der 5. Klasse (Auszug aus RRL)**

Die Schülerin, der Schüler kann...

- informatische Anwendungen für lokale Netzwerke oder für Remote Services entwickeln
- Werkzeuge und Instrumente auf Grund ihrer funktionalen Merkmale auswählen
- Projekte gemäß den von den betrieblichen Verwaltungssystemen vorgesehenen Abläufen und Standards der Qualität und der Sicherheit durchführen
- Produktionsprozesse gemäß den betrieblichen Erfordernissen realisieren
- Rechnersysteme und Netzwerke installieren, konfigurieren und verwalten
- Dokumentationen zu spezifischen Prozessen erstellen und technische Berichte verfassen

**Fertigkeiten und Kenntnisse (Auszug aus RRL)**

3. Klasse und 4. Klasse (2. Biennium)

Fertigkeiten	Kenntnisse	Abkürzung
die funktionalen Aspekte der wichtigsten Komponenten eines Betriebssystems identifizieren und analysieren	Klassifizierung, Struktur und allgemeine Funktionsweise der Betriebssysteme Struktur und Organisation eines Betriebssystems Prinzipien der Prozessverwaltung	F1, K1
für eine bestimmte Entwicklungsumgebung das passende Betriebssystem auswählen	Klassifizierung und Funktionsweise der Betriebssysteme	F2, K2
Anwendungen entwerfen und entwickeln, die auf Betriebssystemebene agieren	Datendarstellung und Codierung, betriebssystemnahe Prozesse	F3, K3
nebenläufige Anwendungen entwerfen und entwickeln	Techniken und Technologien für die Entwicklung nebenläufiger Prozesse und für die Synchronisation des Zugriffs auf gemeinsam genutzte Ressourcen	F4, K4
den Status eines Projektes in Verbindung mit dem gewählten Management-Modell definieren	Managementmodelle und ihre Projektmanagementphasen Techniken und Werkzeuge für die Spezifikation und Verwaltung eines Projektes	F5, K5
verschiedene Dokumentationsunterlagen entsprechend den standardisierten Entwicklungsphasen erstellen	Typen der Dokumentationsprotokolle und Entwicklungsdokumentationen in den verschiedenen Entwicklungsphasen	F6, K6
für die Darstellung und Dokumentation der Entwurfs- und Implementierungsphase auf die geltenden Standards Bezug nehmen	Standards der Darstellung und Dokumentation	F7, K7

## Stoffverteilungsplan<sup>1</sup>

Themenbereich 1: <b>Nebenläufige Programmierung</b>					
<b>F&amp;F<sup>2</sup></b>	<b>K<sup>3</sup></b>	<b>Inhalte</b>	<b>Methodische-didaktische Hinweise</b>	<b>Indikatoren</b>	<b>Minimalziele</b>
F4	K4	Grundlagen der nebenläufigen Programmierung  Synchronisierung	Theoretische Behandlung von Mutex, Semaphoren, Monitoren  Theoretische Behandlung von den Konzepten Deadlock und Starvation  Erstellung von Programmen, die Prozesse und Threads verwenden, in verschiedenen Programmiersprachen (z. B. C und Java)	der Schüler kann mit Prozessen und Threads umgehen, sie starten und sauber beenden  der Schüler kann von einander abhängige Prozesse und Threads synchronisieren  der Schüler kennt die Gefahren von Deadlocks und Starvation und erkennt	der Schüler kennt die Prinzipien der nebenläufigen Programmierung und weiß, welchen Sinn Synchronisierung hat  der Schüler kennt Probleme, die bei nebenläufiger Programmierung auftreten können
F4	K4	Interprozesskommunikation	Erstellung von Programmen (bevorzugt in der Sprache C) unter Verwendung von Pipes, FIFOs, Message Queue, Signale, Shared Memory	der Schüler kann Programme erstellen, in denen Prozesse miteinander kommunizieren  der Schüler kennt die Möglichkeiten der Interprozesskommunikation und kann sie anwenden	der Schüler kennt die Möglichkeiten der Interprozesskommunikation

<sup>1</sup> Ab hier wird aus Gründen der besseren Lesbarkeit ausschließlich die männliche Form verwendet. Sie bezieht sich auf Personen beiderlei Geschlechts.

<sup>2</sup> Fertigkeiten und Fähigkeiten

<sup>3</sup> Kenntnisse

Themenbereich 2: <b>Einführung in Projektmanagement</b>					
<b>F&amp;F<sup>4</sup></b>	<b>K<sup>5</sup></b>	<b>Inhalte</b>	<b>Methodische-didaktische Hinweise</b>	<b>Indikatoren</b>	<b>Minimalziele</b>
F5 F6	K5 K6	Anforderungsanalyse  Lasten- und Pflichtenheft  Diagramme für die Anforderungsanalyse	anhand konkreter Anwendungsbeispiele werden Anforderungen gesammelt  Erstellung von Lasten- und Pflichtenheften  Erstellung von geeigneten Diagrammen (händisch und mit geeigneter Software)	der Schüler erkennt die Anforderungen an ein Projekt  der Schüler kann Anforderungen sammeln, definieren und formalisieren  der Schüler kann auf korrekte Art und Weise Lasten- und Pflichtenhefte erstellen  der Schüler kennt gängige Diagramme für die Anforderungsanalyse und kann diese korrekt verwenden	der Schüler erkennt die grundlegenden Anforderungen an Projekte  der Schüler versteht ein Lastenheft und kann ein basales Pflichtenheft erstellen
F5	K5	Modelle der Projektentwicklung	Besprechung und Analyse verschiedener Vorgehensmodelle bei Projekten insbesondere bei Softwareprojekten  Erstellung von kleinen Projekten in Teamarbeit	der Schüler kann für ein Projekt ein passendes Modell der Projektentwicklung auswählen und korrekt anwenden  der Schüler kann im Team in verschiedenen Positionen arbeiten	der Schüler kennt unterschiedliche Modelle der Projektentwicklung  der Schüler kann sich in einem Team einbringen

<sup>4</sup> Fertigkeiten und Fähigkeiten

<sup>5</sup> Kenntnisse

Themenbereich 3: <b>Software Engineering</b>					
<b>F&amp;F<sup>6</sup></b>	<b>K<sup>7</sup></b>	<b>Inhalte</b>	<b>Methodische-didaktische Hinweise</b>	<b>Indikatoren</b>	<b>Minimalziele</b>
F5 F6 F7	K5 K6 K7	Einführung in Software Engineering  Lebenszyklus von Software  Qualitätsmerkmale von Software	theoretische Behandlung der verschiedenen Aspekte des Software Engineerings  anhand eines Projekts werden die Methoden des Software Engineerings konkret angewendet  Teamarbeit	Der Schüler kennt die einzelnen Stufen der Entwicklung von Software  Der Schüler kann Software anhand von geltenden Standards entwickeln  Der Schüler kennt die Wichtigkeit der Wartung von Software	der Schüler hat einen Überblick über die einzelnen Entwicklungsphasen von Software und kann ein Softwareprojekt in einem Team mitgestalten
F6 F7	K6 K7	UML  Softwaretests  Dokumentation von Software	theoretische Behandlung der verschiedenen Diagrammtypen, die UML anbietet  praktische Anwendung innerhalb konkreter Projekte  innerhalb von Softwareprojekten werden Tests eingebaut und verwendet (z. B. JUnit-Tests)  innerhalb von Softwareprojekten wird der Code mit passenden Methoden dokumentiert (z. B. Javadoc)	Der Schüler hat einen guten Überblick über die in UML angebotenen Diagrammtypen und kann diese korrekt anwenden  der Schüler kann Softwaretests korrekt anwenden  der Schüler kennt Tools und Normen der Softwaredocumentation und kann Software sauber und vollständig dokumentieren	der Schüler kennt die wichtigsten Diagrammtypen und kann sie anwenden  der Schüler weiß, warum Tests wichtig sind und wie sie verwendet werden  der Schüler kann Software dokumentieren

<sup>6</sup> Fertigkeiten und Fähigkeiten

<sup>7</sup> Kenntnisse