| Semester | Inhaltsbezogene Kompetenzen/Standards | Prozess-bezogene Kompetenzen | geeignete Inhalte zum Kompetenzerwerb | Anmerkungen |
|---|---|---|---|---|
| 1.Semester Erstmalig Informatik | **Rechner und Netze** | | Grundlagen Informatik, Tannenbaum, Informatik-Duden, | Bezug auf reale ‚Schichtenmodelle'<br><br>Praxisdemonstrationen z.B Netzwerkproblemortung |
| | Schichtenarchitektur | Kenntnis und Einordnung in Modelle | | |
| | VON-NEUMANN-Architektur | | | |
| | Client-Server-Struktur | | | |
| | Protokolle | | | |
| | Einblicke in die Geschichte der Informatik | Recherchieren / Auswählen / Auswerten und Präsentieren | | |
| | **Datenbanken und Datenschutz** | | Einstieg in SQL (Throll/Bartosch)<br><br>LPI-Unterlagen, Gesellschaft für Datenschutz/sicherheit | Bewältigung von Informationsbergen,<br><br>Moorsche Gesetz |
| | Überblick zu Datenbanksystemen (DBS) | Wissen | | |
| | Benutzung eines einfachen relationalen DBS | Anwenden | | |
| | Datenschutz und Datensicherheit | Wissen | | |
| | Datenschutzgesetz mit Fallbeispielen | Bewerten | | |
| 2.Semester Erstmalig Informatik | **Grundlagen der Programmentwicklung** | | | |
| | Grundlagen von OOA, OOD, OOP (Klasse, Attribut, Methode, Instanz) | Wissen und Anwendung an exemplarischen Beispielen (vereinfacht) | Freepascal mit Entwicklungsumgebung Lazarus | Beschränkung auf eine Sprache und deren Elementen |
| | Einfache dokumentationsunterstützende Techniken (Klassendiagramm, Struktogramm, Pseudocode) | | | |
| | Elementare Daten- und Steuerstrukturen (Algorithmik im Kleinen), Integration vorhandener Software-Bausteine in eigene Programme | | | |
| | Analyse eines gegebenen Projektes, Einblicke in die Grundlagen systematischer Softwareentwicklung (Software-Life-Cycle), Ergonomie | | | |
| 3.Semester Erstmalig Informatik | **Datenbanken und Softwareentwicklung I** | | | |
| | • Datenmodellierung<br>• relationales Datenbankschema<br>• praktische Umsetzung in ein Datenbank-Managementsystem<br>• Abfragen (Projektion, Selektion, Join)<br>• Datenschutz und Datensicherheit<br>• Algorithmen und Datenstrukturen | Vergleich mit anderen Ordnungssystemen, einfache Anwendungen bedienen können<br><br>· Strukturelle Aspekte bei der | Fortführung Einstieg SQL, Datenbankmodellierung mit S-Design | Fortführung DB und Programmierkurs aus 1.und 2.Semester |

| | | | | |
|---|---|---|---|---|
| | • objektorientierte Modellierung (z. B. UML-Klassendiagramme)<br>• objektorientierte Programmierung<br>• Syntax und Semantik (Syntaxdiagramme) | Konstruktion von Programm-systemen stehen im Vordergrund.<br>· Analyse und Modellierung folgen dem objektorientierten Ansatz.<br>· Projektorientierte Vorgehens-weise mit einem oder mehreren Kurzprojekten<br>· Dokumentation der Entwurfs-schritte mit Hilfe des Rechners<br>· Verwendung von Formatvor-lagen für Textverarbeitung<br>· beispielhafte Vektorgrafiken für Modulhierarchie, Funktionale Spezifikation bzw. Klassenbeschreibungen (z. B. mit Hilfe von UML), Anforderungsdefinition, Testumgebungen, Quelltexte<br>· Der Einsatz eines einfachen CASE-Werkzeuges wird empfohlen. | Beispielprogramme analysieren,<br>Planung von ‚Machbaren'<br>Versionsverwaltung<br>Office-Online<br>Lazarus<br>S-Designer, SQL-Teacher | Beispiele von Softwareprojekten aus Schülerarbeiten<br>Ordnung von Versionen, auch einfache Dateiverwaltung und gemeinsamen Zugriff organisieren<br>Beschränkung auf eine DIE<br>Entscheidungskriteri en des Programms verdeutlichen |
| 4.Semester Erstmalig Informatik | **Datenbanken und Softwareentwicklung II** | | | |
| | Fortsetzung und Beendigung des Themenbereichs Softwareentwicklung aus dem vorigen Kurshalbjahr | Siehe 3.Semester - Fortführung | Kleines Softwareprojekt – enge Zielsetzung, Vorgabe Freepascal und Lazarus | Weniger ist mehr – Beschränkung und Wiederverwendung von Codefragmenten |

| | | | | |
|---|---|---|---|---|
| 1.Semester Fortgeführt Informatik | Wie 1.Semester erstmalig + Vertiefungsgebiet aus<br>• V1 Applikative Programmierung (funktional oder logisch)<br>• V2 Kryptologie und Datensicherheit<br>• V3 Computergrafik<br>• V4 Computer-Netze<br>• V5 Künstliche Intelligenz<br>• V6 Technische Informatik<br>• V7 Maschinennahe Programmierung<br>• V8 Informatik und Gesellschaft<br>• V9 Theoretische Informatik | Verschlüsselung Aufwand, Nutzen und Handhabung<br><br>Grundlagen heutiger Netze<br><br>Technische Grundlagen der IT-Revolution | Huffmanncode, synchrone und asynchrone Verschlüsselung, PGP<br>Zusammenspiel Hardware / Software | Vertiefung aus Wahlpflicht Informatik nutzen |
| 2.Semester Fortgeführt Informatik | **Grundlagen der Programmentwicklung mit Objektpascal und Java** | | | |
| | Grundlagen von OOA, OOD, OOP (Klasse, Attribut, Methode, Instanz) | Wissen und Anwendung an exemplarischen Beispielen (vertiefend) | FreePascal mit Lazarus und Java mit Greenfoot/BlueJ vertiefend | Einführung von Java als Alternative zu FreePascal |
| | Einfache dokumentationsunterstützende Techniken (Klassendiagramm, Struktogramm, Pseudocode) | | | |
| | Elementare Daten- und Steuerstrukturen (Algorithmik im Kleinen), Integration vorhandener Software-Bausteine in eigene Programme | | | |
| | Analyse eines gegebenen Projektes, Einblicke in die Grundlagen systematischer Softwareentwicklung (Software-Life-Cycle), Ergonomie | | | |
| 3.Semester Fortgeführt Informatik | **Datenbanken und Softwareentwicklung I** | | | |
| | • Datenmodellierung<br>• relationales Datenbankschema<br>• praktische Umsetzung in ein Datenbank-Managementsystem<br>• Abfragen (Projektion, Selektion, Join)<br>• Datenschutz und Datensicherheit<br>• Algorithmen und Datenstrukturen | Vergleich mit anderen Ordnungssystemen, einfache Anwendungen bedienen können<br><br>· Strukturelle Aspekte bei der Konstruktion von Programm- | GIS in der Welt<br><br>Java mit Greenfoot und BlueJ | GoogleMaps & Co.<br><br>Klassendefinitionen analysieren und erstellen |

| | | | | |
|---|---|---|---|---|
| | • objektorientierte Modellierung (z. B. UML-Klassendiagramme)<br>• objektorientierte Programmierung<br>• Syntax und Semantik (Syntaxdiagramme) | systemen stehen im Vordergrund.<br>· Analyse und Modellierung folgen dem objektorientierten Ansatz.<br>· Projektorientierte Vorgehens-weise mit einem oder mehreren Kurzprojekten<br>· Dokumentation der Entwurfs-schritte mit Hilfe des Rechners<br>· Verwendung von Formatvor-lagen für Textverarbeitung<br>· beispielhafte Vektorgrafiken für Modulhierarchie, Funktionale Spezifikation bzw. Klassenbeschreibungen (z. B. mit Hilfe von UML), Anforderungsdefinition, Testumgebungen, Quelltexte<br>· Der Einsatz eines einfachen CASE-Werkzeuges wird empfohlen. | Herausstellung anderer Herangehensweisen<br><br>Fester Rahmen für Umfang und Inhalt | Objektcharakter herausarbeiten<br><br>Einzelprojekte – Schwerpunkt aufs Programmieren<br><br>Darstellung der Projekte<br><br>Javadoc nutzen |
| *4.Semester Fortgeführt Informatik* | **Datenbanken und Softwareentwicklung II** | | | |
| | Wie 4.Semester erstmalig mit folgenden Kriterien zur Wahl des Projektthemas:<br>Das Problem muss mindestens so komplex sein, dass die durch den Software-Life-Cycle (SLC) implizierten Methoden als notwendig erlebt werden; andererseits ist der Umfang so einzuschränken, dass mindestens ein Prototyp erstellt werden kann.<br>· Die Projektarbeit darf sich nicht nur auf eine Phase der Softwareentwicklung erstrecken<br>· Die Projektarbeit darf sich nicht auf reine Programmiertätigkeiten beschränken.<br>· Das Thema sollte aus dem Erfahrungskreis der Schülerinnen und Schüler kommen, | Kriterien zur Wahl des Projektthemas<br>· Das Problem muss mindestens so komplex sein, dass die durch den Software-Life-Cycle (SLC) implizierten Methoden als notwendig erlebt werden; andererseits ist der Umfang | Auswahl Freepascal oder Java, formale und inhaltliche Anforderungen, Zeitmanagement | Nach Können ‚FreePascal oder Java'<br><br>Anforderungen an Projekt formal und inhaltlich definiert |

| | | | | |
|---|---|---|---|---|
| | um<br>zeitintensive Sachanalysen bei völlig fremden Themen zu vermeiden oder auf Themen aufbauen, die aus einem Vertiefungsgebiet bereits bekannt sind.<br>・ Das Projekt ist von den Schülerinnen und Schülern selbstständig lösbar.<br>・ In mindestens einer der Projektphasen muss die Anwendersicht behandelt werden. | so einzuschränken, dass mindestens ein Prototyp erstellt werden kann.<br>・ Die Projektarbeit darf sich nicht nur auf eine Phase der Softwareentwicklung erstrecken<br>・ Die Projektarbeit darf sich nicht auf reine Programmiertätigkeiten beschränken.<br>・ Das Thema sollte aus dem Erfahrungskreis der Schülerinnen und Schüler kommen, um zeitintensive Sachanalysen bei völlig fremden Themen zu vermeiden oder auf Themen aufbauen, die aus einem Vertiefungsgebiet bereits bekannt sind.<br>・ Das Projekt ist von den Schülerinnen und Schülern selbstständig lösbar.<br>・ In mindestens einer der Projektphasen muss die Anwendersicht behandelt werden. | Ideenbesprechung, Einschätzung der Realisierbarkeit, Machbarkeitsstudien<br><br>Vorgabe Spiel oder Tool oder Lernsoftware | Nachweis der Arbeitsschritte und des Zeitaufwandes<br><br>Software sollte einsetzbar sein<br><br>Literatur- und Info-beschaffungshinweise, Fehlerkorrektur ist Bestandteil des Softwareprojektes |