

Fassung: 10. Februar 2022

Schulinternes Curriculum Informatik Leistungskurs

(kursiv: s. Vorgaben des Rahmenplans)

5 Wochenstunden = ca. 60 Std. pro Semester

Unterrichtssprache ist C# (kann ggf. wechseln)

Std	Stoffliche Inhalte	Bemerkungen
1. Semester (IN-1) Datenbanken und Softwareentwicklung I		
60	<p>Algorithmen und Datenstrukturen</p> <ul style="list-style-type: none"> • Einführung in C#, Syntax der Sprache, Arbeit mit der Dokumentation, Vergleich der Sprachen • einfache Datentypen, explizites und implizites Casting • höhere Datentypen (Arrays, Listen), Dictionaries • Algorithmenbegriff, Darstellung von Algorithmen (Fluss-, Sequenzdiagramme, PAP, Pseudocode) • Rekursion versus Iteration, am Beispiel der Such- und Sortierverfahren, Türme von Hanoi, Springerproblem, 8-Damenproblem u.a. • Graphenalgorithmen (Breiten- und Tiefensuche, MST, Zusammenhangskomponenten, Dijkstra u.a.) • unterschiedlich effiziente Lösungen für dieselbe Aufgabe; Teile-und-Herrsche-Paradigma, dynamisches Programmieren (z.B. für MaxSubArray- und Dominanzzahlproblem) • Klassifizierung von Algorithmen bez. ihres Aufwandes, \mathcal{O}-Notation • Vergleich schneller Sortierverfahren (HeapSort, QuickSort, MergeSort, BucketSort); • Rucksackproblem, Greedy und dynamische Lösungen • Suchbäume (natürliche S., AVL-Bäume) als Dictionaries; Aufwand der Grundoperationen, die Implementierung erfolgt vorerst in Haskell (große Wiederholung) • strukturelle Induktion auf Bäumen (auch andere Strukturen, z.B. Listen) • Projekt Huffman-Bäume 	<ul style="list-style-type: none"> • die verwendete Hochsprache kann je nach Absprache wechseln (z.B. Typescript), eigene Sprachen sind möglich (Java, C++) • Einsatz von Aufgaben mit „mathematischem Charakter“ • eine möglichst genaue formale Behandlung (z.B. Ermittlung von Komplexitäten, Induktionsbeweise) wird angestrebt • Die Teilnahme an der 1. Runde BwInf ist Pflicht.

Std	Stoffliche Inhalte	Bemerkungen
2. Semester (IN-2): Datenbanken und Softwareentwicklung II		
30	<ul style="list-style-type: none"> • objektorientierte Modellierung (UML-Klassendiagramme) und Programmierung • Konzepte der OOP(Datenkapselung, Vererbung, Polymorphismus, Generalisierung und Spezialisierung) • Aspekte von Softwarequalität • UML: Assoziationstypen; Aggregation und Komposition; Vererbung • Strukturelle Aspekte bei der Konstruktion von Programmsystemen stehen im Vordergrund. • Analyse und Modellierung folgen dem objektorientierten Ansatz (OOM) • Projektorientierte Vorgehensweise mit einem oder mehreren Kurzprojekten (Wasserfallmethode, agiles Projektmanagement) • Dokumentation der Entwurfsschritte mit Hilfe des Rechners: beispielhafte Vektorgrafiken für Modulhierarchie, • Funktionale Spezifikation bzw. Klassenbeschreibungen mit Hilfe von UML, Anforderungsdefinition, Testumgebungen, Quelltexte 	<ul style="list-style-type: none"> • Einführung und Verwendung von \LaTeX
20	<p>Datenbanken</p> <ul style="list-style-type: none"> • Datenmodellierung (Wiederholung ERM) • Erweiterung des ERM durch min-max-Notation, Generalisierung, Rollen, schwache Entitäten • relationales Datenbankschema (Relationenmodell) • praktische Umsetzung in ein Datenbank-Managementsystem • Abfragen (Projektion, Selektion, Join, u.a.) in SQL und relationaler Algebra • Normalisierung (1. bis 3. Normalform) • Hashspeicherverfahren, Kollisionsbehandlung • Projekt: Datenbankerstellung • Zugriff auf Datenbanken aus Programmiersprachen (Interface für DB-Clients) • Datenschutz und Datensicherheit 	<ul style="list-style-type: none"> • Anknüpfung an die Inhalte von WPU 10 (siehe ebda.) • Datenbanken im Internet • Gesellschaftliche Bezüge (Recht auf informationelle Selbstbestimmung; aktuelle Bezüge)



10	<p>Netzwerktechniken:</p> <ul style="list-style-type: none">• Topologie von Netzwerken, Protokolle als Vereinbarungen zwischen Kommunikationspartnern, kabelgebundene und Funknetze, Mobiltelefonnetze, Knoten- bzw. Zellenstruktur und Paketdienst• Protokolle in Schichten, Adressierungen in Netzen, TCP/IP, OSI-7-Schichten-Modell• Client-Server-Struktur, Peer-to-Peer-Netze	
----	---	--

Std	Stoffliche Inhalte	Bemerkungen
3. Semester (IN-3) Grundlagen der Informatik und Vertiefungsgebiet		
60	<ul style="list-style-type: none"> • Stellenwertsysteme und deren Algebra, Gleitkommazahldarstellung • Rechenwerke und deren technische Realisierung • Schaltnetzsynthese und Minimierung, insbesondere das Quine und McCluskey-Verfahren • Schaltwerksanalyse und -synthese • VON-NEUMANN-Architektur • Assembler-Programmierung (Simulationen MOPS, LC3 und Atari800) • zustandsorientierte Modellierung • endliche Automaten, Zustandsübergangsdiagramme • Grammatiken und formale Sprachen, Chomsky-Hierarchie • Pumping-Lemma und reguläre Ausdrücke • TURING-Maschine (nur Einbandmaschinen) • Vergleich natürlicher und formaler Sprachen • Berechenbarkeit, das Halteproblem 	<ul style="list-style-type: none"> • Anknüpfung an den Mathematikunterricht (Aussagenlogik, Funktionsbegriff) • V7 Maschinennahe Programmierung
4. Semester (IN-4): Softwareprojekt		
40	<p>Fortführung eines Projektthemas aus IN-3 (Vertiefungsgebiete V6 Technische Informatik und V7 Maschinennahe Programmierung)</p> <p>große Vortragsreihe zu einem selbstgewählten Thema</p> <ul style="list-style-type: none"> • 	<p>Vorbereitung Präsentationsprüfung</p> <ul style="list-style-type: none"> • Art, Inhalt und Dauer des Vortrages entsprechen den Vorgaben des Abiturs