Ministerium für Bildung und Kultur
SAARLAND

Lehrplan

Informatik

Gymnasiale Oberstufe

Grundkurs

Hauptphase

– Erprobungsphase –

2019

Inhalt

Vorwort

Seit der Entdeckung des Konzeptes der universellen Rechenmaschine und der Formalisierung des Algorithmenbegriffs in den 1930er Jahren hat die Informatik in rasanten Schritten den Alltag durchdrungen und gleichzeitig einen enormen Einfluss auf andere Wissenschaftsbereiche ausgeübt. Ein Schulfach, das dieser schnellen Entwicklung gerecht werden will, wird zwangsläufig Wert darauf legen müssen, Schülerinnen und Schülern eine Auswahl an Basiskonzepten zu vermitteln, die ihnen – in Anlehnung an den Begriff der *fundamentalen Idee* – eine breite, wenn auch exemplarische, Grundlage liefert.

Der vorliegende Lehrplan umfasst dementsprechend eine Auswahl an zentralen Themengebieten, die innerhalb der Informatik vielfältig vernetzt sind und auf diese Weise ihren Beitrag zur Beantwortung der wesentlichen Frage, was sich hinter dem Begriff *Informatik* letztendlich verbirgt, leisten. Ein Großteil der Themen umreißt fachliche Gebiete, die in ihrer konzeptionellen Kombination aus Schlichtheit und gleichzeitiger Mächtigkeit den vielfach zitierten „Test der Zeit“ überdauert haben und somit fundiert sicherstellen, dass der Informatikunterricht sich nicht auf kurzlebige Trends stützt.

Neben den zu vermittelnden inhaltlichen Kompetenzen trägt der Lehrplan auch den prozessbezogenen Kompetenzen Rechnung, die in den *Einheitlichen Prüfungsanforderungen* der Kultusministerkonferenz als *Erwerb und Strukturierung informatischer Kenntnisse, Kennen und Anwenden informatischer Methoden, Kommunizieren und Kooperieren sowie Anwenden informatischer Kenntnisse, Bewerten von Sachverhalten und Reflexion von Zusammenhängen* benannt werden und sich somit eng an die *Bildungsstandards Informatik für die Sekundarstufe II* der *Gesellschaft für Informatik (GI)* sowie an die Richtlinienempfehlungen *Computer Science Curricula* der *Association for Computing Machinery (ACM)* anlehnen.

Die in diesem Lehrplan angeführten fachlichen Kompetenzen dienen einer detaillierten Spezifizierung der angegebenen Fachinhalte; die hier verwendeten Operatoren stellen eine unmittelbare Verbindung zu den erforderlichen Prozesskompetenzen her.

Bei der angegebenen Reihenfolge der fachlichen Inhalte sowie ihrer Anteile an den zur Verfügung stehenden Unterrichtsstunden handelt es sich um grobe Richtlinien: Bei Planung und Durchführung des Unterrichts ist auf eine enge und sinnvolle Verzahnung der Themen zu achten, die ihrerseits den Schülerinnen und Schülern ein klares Bild von der Vernetzung und Relevanz der Themen innerhalb der Fachwissenschaft Informatik vermittelt.

Zugelassene Programmiersprachen sind DELPHI, JAVA und PYTHON.

Zum Umgang mit dem Lehrplan

Der Lehrplan ist nach Themenfeldern gegliedert. Zu jedem Themenfeld werden in einem didaktischen Vorwort die Bedeutung der Thematik für die Schülerinnen und Schüler, die didaktische Konzeption und Besonderheiten wie zum Beispiel notwendige didaktische Reduktionen beschrieben.

In zwei Spalten werden die Inhalte des Themenfeldes und die daraus resultierenden verbindliche Kompetenzerwartungen bzw. Aktivitäten von Schülerinnen und Schülern, die zum Kompetenzerwerb beitragen, formuliert.

Die Kompetenzerwartungen bzw. Aktivitäten von Schülerinnen und Schülern sind bewusst detailliert beschrieben. Dies geschieht mit dem Ziel, die Intensität der Bearbeitung möglichst präzise festzulegen. So kann vermieden werden, dass Themenfelder entweder zu intensiv oder zu oberflächlich behandelt werden. Die detaillierte Beschreibung darf hierbei nicht als Stofffülle missverstanden werden. Der Lehrplan beschränkt sich vielmehr auf wesentliche Inhalte und Themen, die auch Bezugspunkte für schulische und schulübergreifende Leistungsüberprüfungen sind. Darüber hinaus lässt der Lehrplan Zeit für Vertiefungen, individuelle Schwertpunktsetzungen, fachübergreifende Bezüge und die Behandlung aktueller Themen.

Für die verbindlichen Themenfelder sind als Richtwerte jeweils Prozentanteile der insgesamt in der Hauptphase der gymnasialen Oberstufe zu Verfügung stehenden Unterrichtszeit angegeben.

Die zeitliche Abfolge der Inhalte innerhalb des jeweiligen Schulhalbjahres kann den Unterrichtsgegebenheiten angepasst werden.

Die Hinweise geben Anregungen inhaltlicher und methodischer Art.

Themenfelder Hauptphase der gymnasialen Oberstufe

Die angegebenen Anteile beziehen sich auf die in der Hauptphase insgesamt zur Verfügung stehende Unterrichtszeit und sind nicht zuletzt wegen der engen Verzahnung der Themen als ungefähre Richtwerte zu verstehen.

Themenfelder 1. Halbjahr der Hauptphase	**Informatik GK**
Objektorientierte Modellierung / Programmierung	25 %
Rekursion	10 %

Themenfelder 2. Halbjahr der Hauptphase	**Informatik GK**
Suchen und Sortieren	15 %
Dynamische Datenstrukturen: lineare Listen und binäre Bäume	15 %

Themenfelder 3. und 4. Halbjahr der Hauptphase	**Informatik GK**
Kryptologie	20 %
Formale Sprachen	15 %

Objektorientierte Modellierung/Programmierung	**Informatik Hauptphase GK**

Der objektorientierte Ansatz gehört heute in der Informatik zu den am weitesten verbreiteten Programmierparadigmen. Dabei wird in aller Regel dem sogenannten *Message-Passing-Modell* gefolgt, dessen Grundsatz darin besteht, dass der Ablauf eines Programms einer Kommunikation zwischen den erzeugten Objekten entspricht. Damit sind wesentliche Komponenten gegeben, die ein „objektorientierter" Ansatz vorweisen sollte: Objekte mit internem Zustand, der Zusammenfassung dieser Objekten in Klassen mit klassenweise spezialisiertem Verhalten sowie der Möglichkeit einer Vererbung zwischen Klassen, die ein polymorphes Verhalten der Objekte erlaubt.

Der objektorientierte Ansatz fördert in hohem Maße Modellierungs- und Modularisierungskompetenzen und verpflichtet Schülerinnen und Schüler durch die Notwendigkeit der Festlegung auf präzise definierte Schnittstellen zu intensiver Kommunikation.

Inhalte	**Kompetenzerwartungen**
• Objektorientierung und das *Message-Passing-Modell* • Klassen, Objekte, Attribute • Konstruktoren und Methoden	Die Schülerinnen und Schüler • entwerfen *Message-Passing-Modelle* zu vorgegebenen Sachzusammenhängen, • implementieren Klassen mit Attributen primitiver und höherer Typen, • implementieren Konstruktoren und erzeugen Objekte mittels Konstruktoraufruf, • implementieren Methoden mit und ohne Rückgabewert und unterscheiden dabei bzgl. der Kommunikation zwischen *Auftrag* und *Anfrage*, • stellen einzelne Klassen in einer üblichen UML-Notation dar.

Hinweise

Dem *Message-Passing-Modell* liegt im Wesentlichen die Idee zugrunde, dass die erzeugten Objekte miteinander kommunizieren und dies bei der Ausführung des Programms die einzige Möglichkeit des Informationsflusses ist. Die oft mit dem sogenannten „Geheimnisprinzip" in Zusammenhang gebrachte Datenkapselung innerhalb der Objekte bringt insofern mit sich, dass die durch das Objekt gekapselten Daten ausschließlich indirekt (über entsprechende Kommunikation) zugänglich sind. Ein unmittelbarer Zugriff widerspricht demnach der Kernidee des Message-Passing-Modells und verhindert darüber hinaus die Reorganisation der Interna der Klasse.

Einzelne Klassen sind in einer üblichen UML-Notation darzustellen.

Rekursion	**Informatik Hauptphase GK**

Basierend auf der Idee der *rekursiven Funktionen*, die in den früher 1930er Jahren insbesondere durch die Arbeiten von Kurt Gödel maßgeblich dazu beitrug, den Algorithmenbegriff zu formalisieren, erhielt das Konzept der rekursiven Programmierung früh Einzug in das Standardrepertoire der Informatik, und ist seitdem aus dem Gebiet der Algorithmik nicht mehr wegzudenken.

Ziel des Informatikunterrichts ist es, dieses Konzept vorzustellen, und es unmittelbar als mächtiges und gleichzeitig elegantes Werkzeug zu etablieren.

Inhalte	**Kompetenzerwartungen**
• Rekursion als Problemlösungsstrategie • lineare/baumförmige Rekursion und ihre Aufrufschemata • Vergleich äquivalenter rekursiver und iterativer Algorithmenvarianten • rekursive Definition der Fakultät und der Fibonacci-Zahlen	Die Schülerinnen und Schüler • benennen die Merkmale rekursiver Methoden und Definitionen, • skizzieren lineare und baumförmige Aufrufschemata, • wandeln iterative/rekursive Methoden ineinander um, • benennen und implementieren exemplarisch rekursive Definitionen der Fakultät und der Fibonacci-Zahlen, • beurteilen Vor- und Nachteile rekursiver Problemlösungen.

Hinweise

Ggf. durchgeführte Laufzeituntersuchungen rekursiv formulierter Algorithmen können auf formal-mathematische Betrachtungen verzichten; es genügt eine intuitive Vorstellung der Aufrufschemata.

Suchen und Sortieren	**Informatik Hauptphase GK**

Such- und Sortierverfahren auf linearen Reihungen gehören seit langem zu den Klassikern der Algorithmik. Sie haben einen engen Bezug zu vielen alltäglichen, praktischen Situationen und eignen sich in besonderer Weise, eigene Verfahrensideen zu entwerfen, diese zu formalisieren, zu implementieren und schließlich vergleichend zu analysieren. Bzgl. der Analyse bilden sie ein leicht zugängliches Beispiel für eine formale Laufzeitanalyse.

Inhalte	**Kompetenzerwartungen**
• sequentielle Suche • binäre Suche • einfache Sortierverfahren: *bubblesort, selectionsort, insertionsort*	Die Schülerinnen und Schüler • implementieren sequentielle und binäre Suche auf eindimensionalen Feldern, • vollziehen die Sortierverfahren *bubblesort, selectionsort, insertionsort* auf eindimensionalen Feldern an Beispielen nach, • analysieren das Laufzeitverhalten der einfachen Sortierverfahren.

Hinweise

Behandelt werden in diesem Themenbereich ausschließlich Verfahren auf linearen Datenstrukturen mit direktem, indiziertem Zugriff. Ein Ausweiten der genannten Verfahren auf Listenstrukturen bietet sich zwar an, ist jedoch nicht obligatorisch.

Der Schwerpunkt dieser Sequenz liegt auf den algorithmischen Kernideen. Ein Implementieren der Sortieralgorithmen ist nicht obligatorisch.

Die Laufzeitanalysen können in diesem Themenblock intuitiv, d.h. ohne eine formale Präzisierung, durchgeführt werden.

Dynamische Datenstrukturen: lineare Listen und binäre Bäume	**Informatik Hauptphase GK**

Binäre Bäume spielen als Datenstruktur eine zentrale Rolle bei der praktischen Verwaltung und Bearbeitung von Daten. Sie treten in vielfach spezialisierten Varianten auf und nehmen seit jeher eine ganz wesentliche und fundamentale Stellung in traditionsreichen Programmiersprachen wie *Lisp* oder *Prolog* ein. Im Unterricht des Grundkurses sollen die Schülerinnen und Schüler mit den Grundvarianten dieser Strukturen vertraut gemacht werden. Dazu gehört die grundlegende Definition ebenso wie die Einführung in eine einfache Algorithmik auf eben diesen Strukturen.

Inhalte	**Kompetenzerwartungen**
• Definition *binärer Bäume* sowie der Begriffe *Wurzel*, *linker/rechter Teilbaum*, *Knoten*, *Tiefe* • Algorithmik auf binären Bäumen: Suchen, Bestimmen der Tiefe und der Knotenanzahl • Suchbäume • Algorithmik auf Suchbäumen: Suchen, Einfügen, Löschen • Baumtraversierungen: *preorder-*/*inorder-*/*postorder*-Durchläufe	Die Schülerinnen und Schüler • definieren binäre Bäume und Suchbäume, • erläutern die zugehörigen Grundbegriffe, • stellen Bäume graphisch dar, • erläutern die angegebenen Algorithmen auf den Datenstrukturen und führen diese exemplarisch aus, • verwenden rekursive Algorithmen auf Bäumen, • demonstrieren die unterschiedlichen Arten des Baumdurchlaufs.

Hinweise

In der Praxis existiert eine Vielzahl verschiedener Definitionen für die binären Bäume. Bewusst wird an dieser Stelle von einer Vereinheitlichung abgesehen. Nicht zuletzt um eine elegante rekursive Algorithmik zu ermöglichen soll der induktive Charakter der Baumstrukturen im Vordergrund stehen.

Kryptologie	**Informatik Hauptphase GK**

Mit der Möglichkeit der Kommunikation geht unmittelbar auch das Bedürfnis nach Vertraulichkeit einher. Vertrauliche Nachrichten sind nur für die vom Versender vorgesehenen Kommunikationspartner bestimmt, andere Personen sollen dabei keine Kenntnis der Nachrichten erhalten.

In der Geschichte finden sich zahleiche Beispiele von vertraulicher Kommunikation: Schon die Spartaner benutzten vor über 2500 Jahren Verfahren zur Geheimhaltung von Nachrichten. Über die römischen Feldherren bis hin zur ENIGMA-Maschine im 2. Weltkrieg sind historische Verfahren zur Geheimhaltung von Information bekannt. Eines haben alle relevanten historischen Verfahren gemeinsam: Die Motivation zur Geheimhaltung ging dabei vielfach von militärischen Interessen aus.

Erst seit der Digitalisierung der Kommunikation Ende des 20. Jahrhunderts rücken Verfahren zur Geheimhaltung stärker in den Fokus der Allgemeinheit. Große Datenmengen mit zum Teil sensiblen Komponenten werden von Privatpersonen, Behörden und Firmen digital versendet. Erstmals in der Geschichte können diese ungeschützten Daten – in den falschen Händen – gespeichert, verarbeitet und zusammengefügt werden, sofern sie nicht entsprechend geschützt werden.

Die Unterrichtsreihe zur Kryptologie greift die historischen Verfahren auf und klärt an diesen die benötigten Grundbegriffe. Es werden moderne Verfahren erarbeitet, die Geheimhaltung, Authentizität und Integrität gewährleisten. Die mathematische Grundlage bildet nahezu durchweg die modulare Arithmetik.

Inhalte	**Kompetenzerwartungen**
• kryptographische Grundbegriffe: *Klartext*, *Geheimtext*, *Schlüssel*, *Prinzip von Kerckhoffs* • Sicherheit: *Vertraulichkeit*, *Authentizität*, *Integrität* • Chiffrieralgorithmen: *Substitutions*- und *Transpositionsalgorithmen* • klassische symmetrische Verschlüsselungsverfahren: *Skytale*, *Cäsar*, *Vigenère* • Angriffsarten: *Brute-Force*, *Known-Plaintext*, *Häufigkeitsanalysen* • Schlüsselaustauschverfahren: *Diffie-Hellman* • asymmetrische Verschlüsselung: Das *RSA-Verfahren*	Die Schülerinnen und Schüler • nennen und erläutern die angegebenen kryptographischen Grundbegriffe, • unterscheiden und erkennen *Substitutions*- und *Transpositionsalgorithmen,* • ver- und entschlüsseln vorgegebene Texte mit den Verfahren *Skytale*, *Cäsar*, *Vigenère,* • benennen die angegebenen Angriffsarten und führen sie für die Cäsar- und Vigenère-Verschlüsselung durch, • führen das Diffie-Hellman-Verfahren exemplarisch an kleinen Werten durch und erläutern seine Grundidee, • erläutern die Grundidee der asymmetrischen Verschlüsselung anhand des RSA-Verfahrens, • führen das RSA-Verfahren (ohne die Berechnung des privaten Schlüssels) exemplarisch an kleinen Werten durch.

Hinweise

Das Schlüsselaustauschverfahren von Diffie und Hellman gibt nicht nur einen ersten Einblick in die moderne Kryptologie, sondern eignet sich insbesondere auch zur Überleitung in die asymmetrischen Verfahren: Hebt man die Symmetrie des Verfahrens auf und geht zu einer Veröffentlichung der zuerst berechneten Schlüsselkomponente über, so gelangt man automatisch zur Kernidee der asymmetrischen Verschlüsselung und vollzieht gleichzeitig die historische, gleich durch zwei Turing-Awards gekrönte Entwicklung nach.

Bei der Thematisierung der modernen Verfahren darf auf mathematische, den zugrunde liegenden Restklassenring betreffende Details und dementsprechend auf einen Nachweis der Korrektheit verzichtet werden. Nichtsdestotrotz können und sollen die Schülerinnen und Schüler aber etwa eine Anwendung der Potenzgesetze beim Diffie-Hellman-Verfahren erkennen und nachvollziehen können.

Formale Sprachen	**Informatik Hauptphase GK**
Einen einführenden Zugang zu den theoretischen Grundlagen der Informatik ermöglicht die Untersuchung endlicher Automaten und einfacher Klassen formaler Sprachen. Endliche Automaten werden einerseits als abstrakte Modelle zur Beschreibung von Abläufen und der Funktionsweise von Maschinen eingeführt, andererseits ermöglichen sie eine einfache und zugleich präzise Beschreibung regulärer Sprachen. Dieser Zusammenhang zwischen Zeichenmengen und sie erkennender Automaten führt zur Definition des Begriffs der formalen Sprache. Am Beispiel der regulären Sprachen werden die praktische Bedeutung formaler Sprachen bei der Kommunikation zwischen Mensch und Maschine sowie ihre Beschreibungsmöglichkeiten erläutert.	
Inhalte	**Kompetenzerwartungen**
Formale Sprachen • Definition des Begriffs der *formalen Sprache* • Beschreibungen durch Aufzählung bzw. durch charakterisierende Eigenschaften der Worte Endliche Automaten ohne Ausgabe • formale Definition deterministischer und nichtdeterministischer endlicher Automaten als 5-Tupel (Akzeptor) • Beschreibung der Funktionsweise durch Übergangstabellen sowie durch Übergangsgraphen • Konstruktion endlicher Automaten zu vorgegebenen Problemstellungen • Teilmengenkonstruktion Grammatiken • formale Definition einer Grammatik als 4-Tupel • Wortprüfung durch Ableitung • Umwandlung regulärer Grammatiken und zugehöriger akzeptierender Automaten ineinander.	Die Schülerinnen und Schüler • definieren den Begriff der formalen Sprache und beschreiben formale Sprachen durch Aufzählung bzw. durch die Angabe charakterisierender Eigenschaften, • definieren akzeptierende endliche Automaten als 5-Tupel. • stellen endliche Automaten durch Übergangsgraphen sowie durch Übergangstabellen dar, • konstruieren endliche Automaten zu vorgegebenen Problemstellungen, • erstellen deterministische endliche Automaten mit Hilfe der Teilmengenkonstruktion, • definieren Grammatiken als 4-Tupel, • bestimmen die Ableitung vorgegebener ableitbarer Worte, • wandeln reguläre Grammatiken und entsprechende akzeptierende endliche Automaten ineinander um.

Formale Sprachen	Informatik Hauptphase GK

Hinweise

Endliche Automaten zur Modellierung von Abläufen und zur Spezifikation des Verhaltens einfacher Maschinen können an Beispielen aus der Praxis (Getränkeautomat, Parkscheinautomat) erklärt werden. Die Konstruktion endlicher Akzeptoren zu Problemstellungen wie der Suche vorgegebener Teilstrings in Zeichenketten (*pattern-matching*) führt zum Phänomen nichtdeterministischer Automaten. Endliche Automaten, die Folgen von Zeichen über einem Eingabealphabet akzeptieren, führen zum Begriff der formalen Sprache.

Im Gegensatz zur Menge der Worte einer natürlichen Sprache können viele formale Sprachen über die Beschreibung kennzeichnender Eigenschaften der Wörter oder über Bildungsregeln festgelegt werden. Für viele formale Sprachen ist eine vollständige Beschreibung durch charakterisierende Eigenschaften der Wörter nicht möglich. Grammatiken sind Regelwerke, welche die Herleitung aller Wörter einer Sprache ermöglichen.

Der Zusammenhang zwischen Grammatiken und Sprachen wird an einfachen Beispielen erarbeitet. Von wesentlicher Bedeutung ist hierbei die Klärung der (im Allgemeinen unentscheidbaren) Frage, ob eine vorgegebene Folge von Terminalsymbolen nach den Produktionsregeln einer vorgegebenen Grammatik aus der Startvariablen abgeleitet werden kann.