

FACHCURRICULUM INFORMATIK am RG

3. bis 5. Klasse, Realgymnasium mit Schwerpunkt Angewandte Naturwissenschaften

Ziele

Der Informatikunterricht trägt dazu bei, digitale Hilfsmittel sachgerecht und zielgerichtet, verantwortungs- und sinnvoll einzusetzen. Sie sind im Informatikunterricht stets Medium, Werkzeug und Inhalt des Lernens zugleich.

Schülerinnen und Schüler lernen die wichtigsten theoretischen Grundlagen der Informationswissenschaften und erwerben Fertigkeiten in der Nutzung von Instrumenten der Informatik. Dabei sollen ihnen die Prinzipien klar werden, auf denen die verschiedenen aktuellen Varianten eines Informatiksystems gründen. Schülerinnen und Schüler lernen neben der Bedienung entsprechender Systeme vor allem die darauf ausgerichteten Arbeitsmethoden erkennen und beherrschen, damit sie auch künftige technische Entwicklungen besser beurteilen können und ihre Vorkenntnisse und bestehenden Fähigkeiten selbstständig weiterentwickeln.

Im Informatikunterricht sind die inhaltlichen Bereiche Information und Daten, Algorithmen, Sprachen und Automaten, Informatiksysteme, Informatik, Mensch und Gesellschaft immer eng verzahnt mit den Prozessbereichen des Modellierens und Implementierens, Begründens und Bewertens, Strukturierens und Vernetzens, Kommunizierens und Kooperierens, Darstellens und Interpretierens.

Die Schülerinnen und Schüler erwerben die notwendigen Kompetenzen, um die durch Informatiksysteme veränderte Lebenswelt zumindest zu einem wesentlichen Teil zu durchschauen, verstehen, beurteilen und mitgestalten zu können. Im Informatikunterricht werden die Gemeinsamkeiten der für alle Disziplinen gültigen informatischen Strukturen und Methoden einschließlich fächerübergreifend und fächerverbindend, hauptsächlich zur Mathematik, Physik und zu den Naturwissenschaften gestaltet.

Weiters ist es wichtig, Synergien mit dem Umfeld zu finden, indem die Zusammenarbeit mit Universitäten, Forschungseinrichtungen, Wissenschaftsmuseen und der Arbeitswelt ermöglicht wird.

Kompetenzen am Ende der 5. Klasse

Die Schülerin, der Schüler kann:

- mit den gängigsten Software-Programmen umgehen und sie für das eigene Lernen und die Kommunikation im Netz zielgerichtet einsetzen
- unter Nutzung innovativer Methoden und Techniken Daten suchen, analysieren, interpretieren, organisieren, verarbeiten, darstellen und präsentieren
- zu gegebenen Sachverhalten informatische Modelle erstellen, diese mit geeigneten Werkzeugen implementieren und die Implementierung der Modelle reflektieren
- Möglichkeiten und Grenzen der Informatik innerhalb des kulturellen und sozialen Kontextes, in dem sie angewandt wird, bewusst wahrnehmen und reflektieren
- sich selbstständig neue Anwendungen und Informatiksysteme erschließen

2. Biennium Angewandte Naturwissenschaften

| Fertigkeiten | Kenntnisse | Lerninhalte 3. Kl. |
|--|--|--|
| die Charakteristiken der Computerarchitekturen, die Konzepte von Hard- und Software beschreiben | Computerarchitektur, Konzepte von Hard- und Software, binäre Codierung | Zahlensysteme Rechnen mit binären Zahlen, Addition, Subtraktion, Zweierkomplement Die wichtigsten Größeneinheiten (bit, Byte, kByte usw) und Umrechnen zwischen diesen Schematischer Aufbau von Rechnern Speicher und Speichergeräte Eingabe- und Ausgabegeräte |
| das Von Neumann-Prinzip und grundlegende Konzepte von ASCII -Code und Unicode verstehen | Von Neumann-Maschine, ASCII -Code, Unicode | Von Neumann-Maschine Kodierungen Historische Entwicklung von Kodierungen, Unterschiede in der Codepage |
| das Betriebssystem mit seinen grundlegenden Funktionen und Eigenschaften zielgerichtet nutzen | Betriebssysteme, Hilfsprogramme | 4. Klasse |
| den Begriff des Prozesses als ein sich in Ausführung befindliches Programm verstehen und den grundlegenden Mechanismus der Speicherverwaltung und die wichtigsten Funktionen der Dateisysteme erklären | Prozess, Speicherverwaltung, Dateisysteme | 4. Klasse |

| | | |
|--|---|---|
| <p>die Elemente eines elektronischen Dokumentes erkennen und zielgerichtet einsetzen</p> | <p>Textverarbeitung, Tabellenkalkulation, Präsentationssoftware</p> | <p>Tabellenkalkulation: Zellen, Formeln, Bezüge in Formeln, ausgewählte mathematische Funktionen</p> <p>Datenvisualisierung mit Diagrammen</p> <p>Kurze Einführung oder Wiederholung Textverarbeitung und Präsentationssoftware</p> |
| <p>das Internet und seine Dienste nutzen und erklären</p> | <p>Struktur des Internet und Internetdienste</p> | <p>4. Klasse</p> |
| <p>einen Algorithmus in Pseudocode und in einer bestimmten Programmiersprache entwickeln</p> | <p>algorithmische Grundbausteine, Syntax einer Programmiersprache</p> | <p>Begriff Algorithmus</p> <p>Darstellungsformen: Diagramme, Pseudocode</p> <p>Konzept der Variablen und Wertzuweisung</p> <p>Gegebene Algorithmen lesen</p> <p>Syntax und Semantik einer Sprache</p> <p>Programmieren: Variablen, Zuweisung, Kommentare, mathematische Operatoren und Präzedenz derselben Inkrement und Dekrement Wahrheitswerte, Verknüpfungen mit and und or, Bedingungen, Schleifen Listen (Arrays), Funktionen, Verwendung ausgewählter Bibliotheken</p> |
| <p>die Prinzipien, die den Programmiersprachen zugrunde liegen, verstehen und nutzen</p> | <p>Programmiersprachen</p> | <p>Programmiersprache und Abstraktion</p> <p>Programmierparadigmen: Beispiele</p> |

| | | |
|--|--|---|
| | | Kompilierte / interpretierte Sprachen, Compiler, Interpreter Übersicht über Programmiersprachen Symbolische Sprachen |
|--|--|---|

| Fertigkeiten | Kenntnisse | Lerninhalte 4. Kl. |
|--|--|---|
| die Charakteristiken der Computerarchitekturen, die Konzepte von Hard- und Software beschreiben | Computerarchitektur, Konzepte von Hard- und Software, binäre Codierung | 3. Klasse |
| das Von Neumann-Prinzip und grundlegende Konzepte von ASCII -Code und Unicode verstehen | Von Neumann-Maschine, ASCII -Code, Unicode | 3. Klasse |
| das Betriebssystem mit seinen grundlegenden Funktionen und Eigenschaften zielgerichtet nutzen | Betriebssysteme, Hilfsprogramme | Betriebssystem als Schnittstelle zwischen Mensch / Maschine: Aufgaben, Klassifizierung, Architektur Hilfsprogramme: Taskmanager, Console, Dateimanager/GUI |
| den Begriff des Prozesses als ein sich in Ausführung befindliches Programm verstehen und den grundlegenden Mechanismus der Speicherverwaltung und die wichtigsten Funktionen der Dateisysteme erklären | Prozess, Speicherverwaltung, Dateisysteme | Multitasking-Betriebssysteme Prozesse, Status von Prozessen, Beenden und Abbrechen von Prozessen, Kommunikation, evtl. Deadlock Speicherverwaltung: - Speicherhierarchie - Virtualisierung: Swapping / Paging |

| | | |
|--|---|--|
| | | <p>Dateiverwaltung, Benutzerverwaltung, Zugriffsrechte</p> <p>Die gängigsten Datenformate</p> <p>Verwendung von (un)komprimierten Archiven</p> |
| <p>die Elemente eines elektronischen Dokumentes erkennen und zielgerichtet einsetzen</p> | <p>Textverarbeitung, Tabellenkalkulation, Präsentationssoftware</p> | <p>3. Klasse</p> |
| <p>das Internet und seine Dienste nutzen und erklären</p> | <p>Struktur des Internet und Internetdienste</p> | <p>IP-Adressen, Domain Name System,</p> <p>Dienste: http, https, email</p> <p>Webservices: Definition, Datenaustauschformate: XML, Json, csv</p> <p>Externe Datenquellen verwenden z.B. Opendata der Provinz Bozen</p> |
| <p>einen Algorithmus in Pseudocode und in einer bestimmten Programmiersprache entwickeln</p> | <p>algorithmische Grundbausteine, Syntax einer Programmiersprache</p> | <p>Wiederholung aus der 3. Klasse</p> <p>Vertiefung</p> |
| <p>die Prinzipien, die den Programmiersprachen zugrunde liegen, verstehen und nutzen</p> | <p>Programmiersprachen</p> | <p>3. Klasse</p> |

5. Klasse Angewandte Naturwissenschaften

| Fertigkeiten | Kenntnisse | Lerninhalte 5. Kl. |
|--------------|------------|--------------------|
|--------------|------------|--------------------|

| | | |
|--|---|--|
| <p>gegebene Algorithmen lesen und interpretieren und Algorithmen zum Lösen von Aufgaben und Problemen aus verschiedenen Anwendungsgebieten entwerfen und realisieren</p> | <p>die wichtigsten Algorithmen der Numerik</p> | <p>Numerische Integration, Nullstellensuche (Newton, regula falsi, Bisektionsverfahren)</p> |
| <p>formale Sprachen zur Interaktion mit Informatiksystemen und zum Problemlösen nutzen</p> | <p>formale Sprachen</p> | <p>Formale Sprachen und Grammatik, Endliche Automaten</p> <p>Alphabet, Zustandsdiagramme, Transitionstabelle</p> <p>Evtl. reguläre Ausdrücke</p> |
| <p>für einfache Sachverhalte objektorientierte Modelle entwickeln und diese darstellen, die Vorgehensweisen bei der Modellierung informatischer Sachverhalte begründen</p> | <p>Klassendiagramme, Syntax einer objektorientierten Programmiersprache</p> | <p>Objekt, Instanz, Klasse, Eigenschaften, Eigenschaftswerte, Konzept der Kapselung und Information hiding</p> <p>Klassendiagramme</p> |
| <p>einfache Simulationen zum Problemlösen und zur Unterstützung von wissenschaftlichen Untersuchungen nutzen</p> | <p>Simulationsprogramme</p> | <p>Monte-Carlo Simulation zu verschiedenen Problemstellungen</p> <p>Zufallsvariablen</p> |
| <p>die Verwaltung und Speicherung großer Datenmengen modellieren</p> | <p>Datenmodelle</p> | <p>Konzeptionelles, Logisches, Physisches Datenmodell</p> <p>Entity Relationship-Modell, Relationales Datenmodell</p> <p>Überführung von (einfachen) ER-Modellen in das Relationale Modell</p> |

| | | |
|---|--|--|
| <p>einfache Datenmodelle in relationale Modelle umsetzen und diese mit einem Datenbanksystem realisieren</p> | <p>Datenbanken</p> | <p>Relationale Datenbanksysteme Tabellen, Beziehungen, Abfragen</p> |
| <p>Situationen, in denen persönliche Daten weitergegeben werden, bewerten und die Unsicherheit einfacher Verschlüsselungsverfahren erkennen</p> | <p>Datensicherheit, Privacy, Verschlüsselungsverfahren</p> | <p>Rechtliche Grundlagen Verschlüsselungsverfahren: Symmetrisch / Asymmetrisch (Beispiele: Caesar, Substitutionsverfahren, RSA)</p> |