



Fachcurriculum

3. Klasse

TFO

Informatik

Erziehungs- und Unterrichtsziele (Auszug aus RRL)

Im Unterricht des Faches Informatik setzen sich die Schülerinnen und Schüler vertieft mit der Verwendung von Modellen zum Erforschen von Phänomenen und zum Interpretieren experimenteller auseinander. Sie lernen mit technologischen Werkzeugen umzugehen und Vorgangsweisen und Techniken zum Finden von innovativen und verbessernden Lösungen einzusetzen und erhalten die Möglichkeit, sich in den verschiedenen Phasen und Ebenen des Erstellungsprozesses einer Software, von der Ideenfindung bis zur Realisierung, aktiv einzubringen und die geeigneten Entwurfs-, Dokumentations- und Kontrollwerkzeuge anzuwenden. Dabei erfahren die Schülerinnen und Schüler die Bedeutung ergebnisorientierten Handelns, zielorientierten Arbeitens und erkennen die Notwendigkeit, ethische und dem Berufsstand entsprechende Verantwortung zu übernehmen. Sie verwenden die spezifischen Fachbegriffe aus dem Englischen und die Fachsprache korrekt, beachten die Rechtsvorschriften, welche den spezifischen Bereich regeln und schenken der Arbeitssicherheit und dem Schutz der Person, der Umwelt und des Territoriums besonderer Aufmerksamkeit

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

Die Schülerin, der Schüler kann

- die Strategien des rationalen Denkens in ihren dialektischen und algorithmischen Aspekten einsetzen, um bei Problemstellungen geeignete Lösungen auszuarbeiten
- informatische Anwendungen für lokale Netze oder Online-Dienste entwickeln
- Geräte und Softwareinstrumente aufgrund ihrer funktionalen Merkmale auswählen
- Projekte laut Verfahren und Standards, welche das betriebliche Verwaltungssystem bezüglich Qualität und Sicherheit vorgibt, managen
- technische Berichte verfassen und Arbeitsprozesse dokumentieren

Fertigkeiten und Kenntnisse (Auszug aus RRL)

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

Fertigkeiten	Kenntnisse	Abkürzung
Algorithmen unter Verwendung unterschiedlicher Datenstrukturen entwerfen und implementieren	grundlegende Zusammenhänge zwischen Maschinen, Problemen, Informationen und Sprachen Sprachen und Maschinen auf unterschiedlichen Abstraktionsebenen Programmierparadigmen Softwareentwicklungswerkzeuge und Werkzeuge zum Testen der Robustheit von Programmen	F1, K1
in einer gegebenen Situation die geeignetste Art der Datenorganisation für die Verwaltung von Informationen wählen	wichtige Datenstrukturen und deren Implementierung	F2, K2
Textdateien verwalten	Textdateien	F3, K3
unterschiedliche Algorithmen zur Lösung desselben Problems analysieren und vergleichen	Komplexitätstheorie für Algorithmen iterative und rekursive Logik	F4, K4
objektorientierte Programme entwickeln und implementieren	objektorientierte Programmierung	F5, K5
Benutzerschnittstellen entwerfen und realisieren	ereignisorientierte Programmierung und grafische Benutzeroberflächen	F6, K6
statische Web-Seiten mit lokaler Interaktion entwerfen, realisieren und warten	Programmiersprachen zur Definition von Web-Seiten clientseitige Programmiersprachen zur lokalen Verarbeitung von Ereignissen in Web-Seiten	F7, K7

Stoffverteilungsplan¹

Themenbereich 1: Programmierung					
F&F ²	K ³	Inhalte	Methodische-didaktische Hinweise	Indikatoren	Minimalziele
	K1	Einführung in die Informatik <ul style="list-style-type: none"> • Informatik und Verarbeitungssysteme • Informationen und Daten • Unterschied Hard- und Software • Programmiersprachen • Quell- und Maschinencode • Compiler und Interpreter • Debugger • Programmierparadigmen • Auswirkungen der Informatik auf die Gesellschaft • Berufsbilder der Informatik 	Lehrervortrag Diskussion in der Klasse Internetrecherchen	Der Schüler kennt die wichtigsten Begriffe der Informatik und kann sie erklären. Der Schüler kennt grundlegende Zusammenhänge zwischen Maschinen, Problemen, Informationen und Sprachen Der Schüler kennt Sprachen und Maschinen auf unterschiedlichen Abstraktionsebenen Der Schüler kennt die wichtigsten Programmierparadigmen	Der Schüler kennt die wichtigsten Begriffe der Informatik und kann sie erklären. Der Schüler kennt grundlegende Zusammenhänge zwischen Maschinen, Problemen, Informationen und Sprachen
F1	K1, K4	Einführung in die Programmierung <ul style="list-style-type: none"> • Von der Problemstellung zum Algorithmus (Ablaufdiagramm, Struktogramm, Pseudocode) • Ein- Ausgabe • Standards 	Lehrervortrag Diskussion in der Klasse Programmierübungen Internetrecherchen	Der Schüler kann die Strategien des rationalen Denkens in ihren dialektischen und algorithmischen Aspekten einsetzen, um bei Problemstellungen geeignete Lösungen auszuarbeiten Der Schüler kann textuelle Benutzerschnittstellen entwerfen und realisieren	Der Schüler kann die Strategien des rationalen Denkens in ihren dialektischen und algorithmischen Aspekten einsetzen, um bei einfachen Problemstellungen geeignete Lösungen auszuarbeiten Der Schüler kann einfache textuelle Benutzerschnittstellen entwerfen und realisieren

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

² Fertigkeiten und Fähigkeiten

³ Kenntnisse

	<ul style="list-style-type: none"> • Variablen und Konstanten • Datentypen • Operatoren • Errorhandling • Verzweigungen • Schleifen • Funktionen • Casten • Pointer • Arrays • Strings • Headerdateien • Zufallszahlen • IDE (Dev-C++, CLion, Atom) <p>Programmierung grundlegender Algorithmen</p>	<p>Gruppenarbeit</p> <p>Projekt</p>	<p>Der Schüler kennt die iterative und rekursive Logik und kann sie anwenden.</p> <p>Der Schüler kennt einige Softwareentwicklungswerkzeuge und Werkzeuge zum Testen der Robustheit von Programmen und kann diese anwenden.</p> <p>Der Schüler kann technische Berichte verfassen und Arbeitsprozesse dokumentieren</p>	<p>Der Schüler kennt die iterative und rekursive Logik und kann sie bei einfachen Algorithmen anwenden.</p> <p>Der Schüler kennt einige Softwareentwicklungswerkzeuge und Werkzeuge zum Testen der Robustheit von Programmen und kann diese anwenden.</p> <p>Der Schüler kann einfache technische Berichte verfassen und Arbeitsprozesse dokumentieren</p>
--	---	-------------------------------------	---	--

Themenbereich 2: Datenstrukturen					
F&F⁴	K⁵	Inhalte	Methodische-didaktische Hinweise	Indikatoren	Minimalziele
F1, F2,	K1, K2, F4	<p>Zusammengesetzte Datentypen</p> <p>Dynamische Datenstrukturen</p> <ul style="list-style-type: none"> • Listen • Bäume • FIFO (queue) • FILO (stack) 	<p>Lehrervortrag</p> <p>Diskussion in der Klasse</p> <p>Programmierübungen</p> <p>Internetrecherchen</p>	<p>Der Schüler kann die Strategien des rationalen Denkens in ihren dialektischen und algorithmischen Aspekten einsetzen, um bei Problemstellungen geeignete Lösungen auszuarbeiten</p> <p>Der Schüler kann textuelle Benutzerschnittstellen entwerfen und realisieren</p>	<p>Der Schüler kann die Strategien des rationalen Denkens in ihren dialektischen und algorithmischen Aspekten einsetzen, um bei einfachen Problemstellungen geeignete Lösungen auszuarbeiten</p> <p>Der Schüler kann einfache textuelle Benutzerschnittstellen entwerfen und realisieren</p>

⁴ Fertigkeiten und Fähigkeiten

⁵ Kenntnisse

			<p>Gruppenarbeit</p> <p>Projekt</p>	<p>Der Schüler kennt wichtige Datenstrukturen und deren Implementierung</p> <p>Der Schüler kann in einer gegebenen Situation die geeignetste Art der Datenorganisation für die Verwaltung von Informationen wählen</p> <p>Der Schüler kennt die iterative und rekursive Logik und kann sie anwenden.</p> <p>Der Schüler kennt die Komplexitätstheorie für Algorithmen und kann die Algorithmen dementsprechend einordnen.</p> <p>Der Schüler kennt einige Softwareentwicklungswerkzeuge und Werkzeuge zum Testen der Robustheit von Programmen und kann diese anwenden.</p> <p>Der Schüler kann technische Berichte verfassen und Arbeitsprozesse dokumentieren</p>	<p>Der Schüler kennt wichtige Datenstrukturen und deren Implementierung</p> <p>Der Schüler kann in einer gegebenen Situation eine geeignete Art der Datenorganisation für die Verwaltung von Informationen wählen</p> <p>Der Schüler kennt die iterative und rekursive Logik und kann sie bei einfachen Algorithmen anwenden.</p> <p>Der Schüler kennt die Grundlagen der Komplexitätstheorie für Algorithmen und kann einfache Algorithmen dementsprechend einordnen.</p> <p>Der Schüler kennt einige Softwareentwicklungswerkzeuge und Werkzeuge zum Testen der Robustheit von Programmen und kann diese anwenden.</p> <p>Der Schüler kann einfache technische Berichte verfassen und Arbeitsprozesse dokumentieren</p>
--	--	--	-------------------------------------	---	---

Themenbereich 3: Dateien verwalten					
F&F⁶	K⁷	Inhalte	Methodische-didaktische Hinweise	Indikatoren	Minimalziele
F1, F3	K1, K3	<p>Aus Dateien lesen</p> <p>In Dateien schreiben</p> <p>In Dateien suchen</p> <p>Filedeskriptoren</p> <p>Dateiformate (Text, CSV, XML,JSON)</p>	<p>Lehrervortrag</p> <p>Diskussion in der Klasse</p> <p>Programmierübungen</p> <p>Internetrecherchen</p>	<p>Der Schüler kann die Strategien des rationalen Denkens in ihren dialektischen und algorithmischen Aspekten einsetzen, um bei Problemstellungen geeignete Lösungen auszuarbeiten</p> <p>Der Schüler kann textuelle Benutzerschnittstellen entwerfen und realisieren</p> <p>Der Schüler kennt die iterative und rekursive Logik und kann sie anwenden.</p> <p>Der Schüler kennt die Möglichkeit Daten in Dateien zu speichern und die grundlegenden Dateiformate</p> <p>Der Schüler kann Dateien verwalten</p> <p>Der Schüler kennt einige Softwareentwicklungswerkzeuge und Werkzeuge zum Testen der Robustheit von Programmen und kann diese anwenden.</p> <p>Der Schüler kann technische Berichte verfassen und Arbeitsprozesse dokumentieren</p>	<p>Der Schüler kann die Strategien des rationalen Denkens in ihren dialektischen und algorithmischen Aspekten einsetzen, um bei einfachen Problemstellungen geeignete Lösungen auszuarbeiten</p> <p>Der Schüler kann einfache textuelle Benutzerschnittstellen entwerfen und realisieren</p> <p>Der Schüler kennt die Möglichkeit Daten in Textdateien zu speichern und die grundlegenden Dateiformate</p> <p>Der Schüler kann Textdateien verwalten</p> <p>Der Schüler kennt einige Softwareentwicklungswerkzeuge und Werkzeuge zum Testen der Robustheit von Programmen und kann diese anwenden.</p> <p>Der Schüler kann einfache technische Berichte verfassen und Arbeitsprozesse dokumentieren</p>

⁶ Fertigkeiten und Fähigkeiten

⁷ Kenntnisse

Themenbereich 4: Sortierverfahren					
F&F ⁸	K ⁹	Inhalte	Methodische-didaktische Hinweise	Indikatoren	Minimalziele
F1, F4	K1, K4	<p>Sortierverfahren</p> <ul style="list-style-type: none"> • Stupidsort • Bubblesort • Insertionsort • Selectionsort • Mergesort • Quicksort • evtl. Heapsort <p>Komplexitätstheorie</p> <p>O-Notation</p>	<p>Lehrervortrag</p> <p>Diskussion in der Klasse</p> <p>Programmierübungen</p> <p>Internetrecherchen</p> <p>Referate</p> <p>Gruppenarbeit</p> <p>Projekt</p>	<p>Der Schüler kann die Strategien des rationalen Denkens in ihren dialektischen und algorithmischen Aspekten einsetzen, um bei Problemstellungen geeignete Lösungen auszuarbeiten.</p> <p>Der Schüler kann textuelle Benutzerschnittstellen entwerfen und realisieren.</p> <p>Der Schüler kennt die iterative und rekursive Logik und kann sie anwenden.</p> <p>Der Schüler kennt die Komplexitätstheorie für Algorithmen und kann die Algorithmen dementsprechend einordnen.</p> <p>Der Schüler kann unterschiedliche Algorithmen zur Lösung desselben Problems analysieren und vergleichen.</p> <p>Der Schüler kennt einige Softwareentwicklungswerkzeuge und Werkzeuge zum Testen der Robustheit von Programmen und kann diese anwenden.</p> <p>Der Schüler kann technische Berichte verfassen und Arbeitsprozesse dokumentieren.</p>	<p>Der Schüler kann die Strategien des rationalen Denkens in ihren dialektischen und algorithmischen Aspekten einsetzen, um bei einfachen Problemstellungen geeignete Lösungen auszuarbeiten.</p> <p>Der Schüler kann einfache textuelle Benutzerschnittstellen entwerfen und realisieren.</p> <p>Der Schüler kennt die iterative und rekursive Logik und kann sie bei einfachen Algorithmen anwenden.</p> <p>Der Schüler kennt die Grundlagen der Komplexitätstheorie für Algorithmen und kann einfache Algorithmen dementsprechend einordnen.</p> <p>Der Schüler kann einfache unterschiedliche Algorithmen zur Lösung desselben Problems analysieren und vergleichen.</p> <p>Der Schüler kennt einige Softwareentwicklungswerkzeuge und Werkzeuge zum Testen der Robustheit von Programmen und kann diese anwenden.</p> <p>Der Schüler kann einfache technische Berichte verfassen und Arbeitsprozesse dokumentieren.</p>

⁸ Fertigkeiten und Fähigkeiten

⁹ Kenntnisse

Optionaler Themenbereich 5: Grafische Benutzerschnittstellen und Objektorientung					
F&F ¹⁰	K ¹¹	Inhalte	Methodische-didaktische Hinweise	Indikatoren	Minimalziele
F1, F5, F6	K1, K5, K6	<p>Grafische Oberfläche mit C</p> <ul style="list-style-type: none"> • conio.h • curses.h • windows.h • GTK+ <p>C++</p> <ul style="list-style-type: none"> • Grundlagen der Objektorientierung <p>Processing</p> <ul style="list-style-type: none"> • Koordinatensystem • Zeichenfunktionen • Farbdarstellung • trigonometrische Funktionen 	<p>Lehrervortrag</p> <p>Diskussion in der Klasse</p> <p>Programmierübungen</p> <p>Internetrecherchen</p> <p>Gruppenarbeit</p> <p>Projekt</p>	<p>Der Schüler kann die Strategien des rationalen Denkens in ihren dialektischen und algorithmischen Aspekten einsetzen, um bei Problemstellungen geeignete Lösungen auszuarbeiten</p> <p>Der Schüler kennt die grundlegenden Konzepte der objektorientierten Programmierung</p> <p>Der Schüler kann einfache objektorientierte Programme entwickeln und implementieren</p> <p>Der Schüler kann einfache grafische Benutzerschnittstellen entwerfen und realisieren</p> <p>Der Schüler kennt die wichtigsten Konzepte der ereignisorientierte Programmierung und der grafischen Benutzeroberflächen</p> <p>Der Schüler kennt die iterative und rekursive Logik und kann sie anwenden.</p> <p>Der Schüler kennt einige Softwareentwicklungswerkzeuge und Werkzeuge zum Testen der Robustheit von Programmen und kann diese anwenden.</p> <p>Der Schüler kann technische Berichte verfassen und Arbeitsprozesse dokumentieren</p>	

¹⁰ Fertigkeiten und Fähigkeiten

¹¹ Kenntnisse

