



“J. Ph. Fallmerayer“

39042 Brixen/Bressanone, Dantestraße/Via Dante 39/E

☎ 0472/830893/Fax: 0472/837740
info@fallmerayer.it

Str. Nr. /Cod. fisc.: 81006290217

Fachcurriculum

4. 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: Objektorientierung					
F&F²	K³	Inhalte	Methodische-didaktische Hinweise	Indikatoren	Minimalziele
F1, F3, F5	K1, K3, F5	<p>Konzepte der Objektorientierung</p> <ul style="list-style-type: none"> • Klassen/Objekte • Abstraktion/Datenkapselung • Vererbung • Polymorphie • Interfaces • Abstrakte and Anonyme Klassen • Lambdas • Exceptionhandling/Ausnahmebehandlung • Serialisierung/Persistenz • Generizität 	<p>Lehrervortrag</p> <p>Diskussion in der Klasse</p> <p>Programmierübungen</p> <p>Internetrecherchen</p> <p>Gruppenarbeit</p> <p>Projekt</p>	<p>Der Schüler kennt die wichtigsten Begriffe der Objektorientierung und kann sie erklären.</p> <p>Der Schüler kennt grundlegende Zusammenhänge zwischen Maschinen, Problemen, Informationen und Sprachen</p> <p>Der Schüler kennt Sprachen und Maschinen auf unterschiedlichen Abstraktionsebenen</p> <p>Der Schüler kennt die wichtigsten Programmierparadigmen</p> <p>Der Schüler kann die Strategien des rationalen Denkens in ihren dialektischen und algorithmischen Aspekten einsetzen, um bei Problemstellungen geeignete objektorientierte 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 wichtigsten Begriffe der Objektorientierung und kann sie erklären.</p> <p>Der Schüler kennt grundlegende Zusammenhänge zwischen Maschinen, Problemen, Informationen und Sprachen</p> <p>Der Schüler kann die Strategien des rationalen Denkens in ihren dialektischen und algorithmischen Aspekten einsetzen, um bei einfachen Problemstellungen geeignete objektorientierte 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 grundlegenden Exceptiontypen und kann einfache Methoden für das Exceptionshandling anwenden.</p>

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

				<p>Der Schüler kennt verschiedene Exceptiontypen und kann geeignete Methoden für das Exceptionshandling anwenden.</p> <p>Der Schüler kennt die Möglichkeit der Serialisierung und kann Daten, die aus einem objektorientierten Programm hervorgehen abspeichern und einlesen</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 Möglichkeit der Serialisierung und kann Daten, die aus einem objektorientierten Programm hervorgehen abspeichern und einlesen</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: Grafische Benutzerschnittstellen					
F&F ⁴	K ⁵	Inhalte	Methodische-didaktische Hinweise	Indikatoren	Minimalziele
F1, F5, F6	K1, K5, K6	<p>Ereignisorientierte Programmierung und grafische Benutzeroberflächen</p> <ul style="list-style-type: none"> • Grafische Klassen/Objekte • Layouts • CSS • Event Handling • Animationen • MVC-Pattern 	<p>Lehrervortrag</p> <p>Diskussion in der Klasse</p> <p>Programmierübungen</p> <p>Internetrecherchen</p> <p>Gruppenarbeit</p>	<p>Der Schüler kann die Strategien des rationalen Denkens in ihren dialektischen und algorithmischen Aspekten einsetzen, um bei Problemstellungen geeignete objektorientierte Lösungen auszuarbeiten</p> <p>Der Schüler kennt die Möglichkeiten der ereignisorientierten Programmierung und der grafischen Benutzeroberflächen.</p>	<p>Der Schüler kann die Strategien des rationalen Denkens in ihren dialektischen und algorithmischen Aspekten einsetzen, um bei einfachen Problemstellungen geeignete objektorientierte Lösungen auszuarbeiten</p> <p>Der Schüler kennt die Möglichkeiten der ereignisorientierten Programmierung und der grafischen Benutzeroberflächen.</p>

⁴ Fertigkeiten und Fähigkeiten

⁵ Kenntnisse

		<ul style="list-style-type: none"> Observer-Pattern 	Projekt	<p>Der Schüler kann grafische Benutzerschnittstellen in einer objektorientierten Programmiersprache entwerfen und realisieren</p> <p>Der Schüler kennt die iterative und rekursive Logik und kann sie anwenden.</p> <p>Der Schüler kennt verschiedene Exceptiontypen und kann geeignete Methoden für das Exceptionshandling 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 kann einfache grafische Benutzerschnittstellen in einer objektorientierten Programmiersprache 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 grundlegenden Exceptiontypen und kann einfache Methoden für das Exceptionshandling 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 3: Datenstrukturen und Graphentheorie					
F&F⁶	K⁷	Inhalte	Methodische-didaktische Hinweise	Indikatoren	Minimalziele
F1, F2, F3, F5,	K1, K2, K3, K5	Grundlegende Datenstrukturen/Collections <ul style="list-style-type: none"> Arrays Listen 	Lehrervortrag Diskussion in der Klasse	Der Schüler kann die Strategien des rationalen Denkens in ihren dialektischen und algorithmischen Aspekten einsetzen, um bei Problemstellungen geeignete objektorientierte Lösungen auszuarbeiten	Der Schüler kann die Strategien des rationalen Denkens in ihren dialektischen und algorithmischen

⁶ Fertigkeiten und Fähigkeiten

⁷ Kenntnisse

		<ul style="list-style-type: none"> • Hashtabellen • Bäume • Map/Set <p>Graphen und Graphenalgorithmen</p> <ul style="list-style-type: none"> • Gerichtete und ungerichtete Graphen • Kanten und Knoten • Eulerscher Kreis • Euler Weg • Hamiltonscher Kreis • Hamiltonscher Weg • Adjazenzmatrix • Tiefen- und Breitensuche • Dijkstra-Algorithmus • Minimum Spanning Tree • MST-Heuristik • Nearest-Neighbor-Heuristik • Travel Salesman Problem. 	<p>Programmierübungen</p> <p>Internetrecherchen</p> <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 Komplexitätstheorie für Algorithmen und kann die Algorithmen dementsprechend einordnen.</p> <p>Der Schüler kennt die iterative und rekursive Logik und kann sie anwenden.</p> <p>Der Schüler kennt verschiedene Exceptiontypen und kann geeignete Methoden für das Exceptionshandling 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>Aspekten einsetzen, um bei einfachen Problemstellungen geeignete objektorientierte Lösungen auszuarbeiten</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 Grundlagen der Komplexitätstheorie für Algorithmen und kann einfache Algorithmen dementsprechend einordnen.</p> <p>Der Schüler kennt die iterative und rekursive Logik und kann sie bei einfachen Algorithmen anwenden.</p> <p>Der Schüler kennt die grundlegenden Exceptiontypen und kann einfache Methoden für das Exceptionshandling 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 4: App- und Webseitenprogrammierung					
F&F ⁸	K ⁹	Inhalte	Methodische-didaktische Hinweise	Indikatoren	Minimalziele
F1, F5, F6, F7	K1, K5, K6, K7	<p>Webseitenprogrammierung</p> <ul style="list-style-type: none"> • HTML • CSS • JavaScript • jQuery <p>App-Programmierung</p> <ul style="list-style-type: none"> • Betriebssystem Android • App-Programmierung mit Android Studio 	<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 objektorientierte Lösungen auszuarbeiten</p> <p>Der Schüler kennt die Möglichkeiten der ereignisorientierten Programmierung und der grafischen Benutzeroberflächen.</p> <p>Der Schüler kann grafische Benutzerschnittstellen in einer objektorientierten Programmiersprache entwerfen und realisieren.</p> <p>Der Schüler kennt Programmiersprachen zur Definition von Webseiten.</p> <p>Der Schüler kennt clientseitige Programmiersprachen zur lokalen Verarbeitung von Ereignissen in Webseiten.</p> <p>Der Schüler kann statische Webseiten mit lokaler Interaktion entwerfen, realisieren und warten</p> <p>Der Schüler kennt die iterative und rekursive Logik und kann sie anwenden.</p> <p>Der Schüler kennt verschiedene Exceptiontypen und kann geeignete Methoden für das Exceptionshandling anwenden.</p>	<p>Der Schüler kennt die Möglichkeiten der ereignisorientierten Programmierung und der grafischen Benutzeroberflächen.</p> <p>Der Schüler kann einfache grafische Benutzerschnittstellen in einer objektorientierten Programmiersprache 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 objektorientierte Lösungen auszuarbeiten</p> <p>Der Schüler kennt Programmiersprachen zur Definition von Webseiten.</p> <p>Der Schüler kennt clientseitige Programmiersprachen zur lokalen Verarbeitung von Ereignissen in Webseiten.</p> <p>Der Schüler kann einfache, statische Webseiten mit lokaler Interaktion entwerfen, realisieren und warten.</p> <p>Der Schüler kennt die iterative und rekursive Logik und kann sie bei einfachen Algorithmen anwenden.</p>

⁸ Fertigkeiten und Fähigkeiten

⁹ Kenntnisse

			<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 grundlegenden Exceptiontypen und kann einfache Methoden für das Exceptionshandling 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>
--	--	--	---	--

Optimaler Themenbereich 5: **Alternative objektorientierte Programmiersprachen**

F&F ¹⁰	K ¹¹	Inhalte	Methodische-didaktische Hinweise	Indikatoren	Minimalziele
F1, F5, F6	K1, K5, K6	<p>Optionale Inhalte:</p> <p>Gemeinsamkeiten und Unterschiede zwischen objektorientierten Programmiersprachen</p> <ul style="list-style-type: none"> • Java, C++, C# <p>Processing</p> <ul style="list-style-type: none"> • Koordinatensystem • Zeichenfunktionen • Farbdarstellung 	<p>Lehrervortrag</p> <p>Diskussion in der Klasse</p> <p>Programmierübungen</p> <p>Internetrecherchen</p>	<p>Der Schüler kennt die wichtigsten Begriffe der Objektorientierung und kann sie erklären.</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>	

¹⁰ Fertigkeiten und Fähigkeiten

¹¹ Kenntnisse

Fachcurriculum 4. Klasse Informatik – TFO Brixen

		<ul style="list-style-type: none"> • trigonometrische Funktionen <p>Unity</p> <ul style="list-style-type: none"> • Objektorientierte Laufzeit- und Entwicklungsumgebung für Spiele (Spiel-Engine) 		<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 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>	
--	--	---	--	---	--