

Curriculum Einführungsphase – Gymnasium Nepomucenum – Fachschaft Informatik

Die folgenden Kompetenzen aus dem Bereich Kommunizieren und Kooperieren werden in allen Unterrichtsvorhaben der Einführungsphase vertieft und sollen aus Gründen der Lesbarkeit nicht in jedem Unterrichtsvorhaben separat aufgeführt werden:

Die Schülerinnen und Schüler

- verwenden Fachausdrücke bei der Kommunikation über informatische Sachverhalte (K),
- präsentieren Arbeitsabläufe und -ergebnisse (K),
- kommunizieren und kooperieren in Gruppen und in Partnerarbeit (K),
- nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung und gemeinsamen Verwendung von Daten unter Berücksichtigung der Rechteverwaltung (K).
- nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation. (K).

Unterrichtsvorhaben EF-I

Thema: Einführung in die Nutzung von Informatiksystemen und in grundlegende Begrifflichkeiten

Zeitbedarf: ca. 6 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
Prinzipieller Aufbau und Arbeitsweise eines Rechners (a) EVA Prinzip (b) von-Neumann-Architektur Speichern von Daten mit informatischen Systemen am Beispiel der Schulrechner → Netzwerk	Die Schülerinnen und Schüler <ul style="list-style-type: none"> • beschreiben und erläutern den Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der „Von-Neumann-Architektur“ (A), • nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D) 	Arbeit mit dem und Analyse des KnowHow-Computer Bestandteile eines Rechners identifizieren und kategorisieren Einführung in Nutzung des Schulnetzwerkes

Unterrichtsvorhaben EF-II

Thema: Grundlagen der objektorientierten Analyse, Modellierung und Implementierung anhand von statischen Grafikszenen

Zeitbedarf: ca. 12 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Identifikation von Objekten</p> <p>(a) Am Beispiel eines lebensweltnahen Beispiels werden Objekte im Sinne der objektorientierten Modellierung eingeführt.</p> <p>(b) Manche Objekte sind prinzipiell typgleich und werden so zu einer Objektsorte bzw. Objektklasse zusammengefasst.</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none">• ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M),• modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M),• implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I),• stellen den Zustand eines Objekts dar (D).	<p>Material: z.B. Werkzeugkoffer</p> <p>Schülerinnen und Schüler betrachten einen Werkzeugkoffer als Menge gleichartiger Objekte, die in einer Klasse mit Attributen und Methoden zusammengefasst werden können.</p>
<p>2. Analyse von Klassen didaktischer Lernumgebungen</p> <p>(a) Objektorientierte Programmierung als modularisiertes Vorgehen</p> <p>(b) Teilanalyse der Klassen der didaktischen Lernumgebungen SaS</p>		<p>Materialien:</p> <p>Dokumentation der didaktischen Bibliothek SaS</p>
<p>3. Implementierung statischer Szenen</p> <p>(a) Grundaufbau einer Java-Klasse</p> <p>(b) Konzeption einer Szene mit View-Objekt und sichtbaren Objekten</p> <p>(c) Deklaration und Initialisierung von Objekten</p> <p>(d) Methodenaufrufe mit Parameterübergabe zur Manipulation von Objekteigenschaften (z.B. Farbe, Position, Drehung)</p> <p>(e) Einführung des Debuggers</p>		<p>Projekt: z.B. Flaggen verschiedener Länder</p> <p>Schülerinnen und Schüler erstellen ein Programm, das mit Hilfe von geometrischen Objekten der SaS-Umgebung Flaggen verschiedener Länder mit unterschiedlicher Komplexität auf den Bildschirm bringt.</p> <p>Projekt: Wettbewerb: Wer gestaltet das schönste/interessanteste Bild</p>

Unterrichtsvorhaben EF-III

Thema: Grundlagen der objektorientierten Programmierung und algorithmischer Grundstrukturen in Java anhand von einfachen Animationen

Zeitbedarf: ca. 30 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Diskrete Veränderung von statischen Situationen durch Aufruf von Methoden.	Die Schülerinnen und Schüler	Projekt: z.B. Licht an-/ausschalten bei Haus. Ampelphasen Schalten
2. Bewegungsanimationen am Beispiel einfacher grafischer Objekte <ul style="list-style-type: none"> (a) Kontinuierliche Verschiebung eines Objekts mit Hilfe einer Schleife (While-Schleife) (b) Tastaturabfrage zur Realisierung einer Schleifenbedingung für eine Animationsschleife (c) Mehrstufige Animationen mit mehreren sequenziellen Schleifen (d) Tastaturabfragen zur Steuerung von Objekten (e) Prüfung von Kollision zweier Objekte mit Hilfe von Methodenaufrufen und Verzweigungen (IF-Anweisungen) (f) Zählen von Rundenabläufen und Punkten mit Variablen (g) Visualisierung der algorithmischen Abläufe im Debugger (h) Darstellung von Algorithmen in Struktogrammen 	<ul style="list-style-type: none"> • analysieren und erläutern einfache Algorithmen und Programme (A), • entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M), • ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), • modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M), • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M), • modifizieren einfache Algorithmen und Programme (I), 	Projekt: z.B. rollende oder fallende Kugeln Die Schülerinnen und Schüler bewegen Objekte mit Hilfe von Schleifen über den Bildschirm und Prüfen ob Ziele getroffen werden oder nicht. Weiter möglich Projekte: z.B. Simulation der Planetenbewegung im Sonnensystem, Simulation einer Uhr mit Zeigern, Simulation eines Billardtisches, Pong-Spiel, Pfeilwurf
3. Erstellen und Verwalten größerer Mengen einfacher grafischer Objekte	<ul style="list-style-type: none"> • implementieren Klassen in einer Programmiersprache auch unter 	Projekt: z.B. Lauflicht Die Schülerinnen und Schüler simulieren

<p>(a) Erzeugung von Objekten mit Hilfe von Zählschleifen (FOR-Schleife)</p> <p>(b) Verwaltung von Objekten in eindimensionalen Feldern (Arrays)</p> <p>(c) Animation von Objekten, die in eindimensionalen Feldern (Arrays) verwaltet werden</p> <p>(d) Binärzahlen</p>	<p>Nutzung dokumentierter Klassenbibliotheken (I),</p> <ul style="list-style-type: none"> • implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I), • implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I), • testen Programme schrittweise anhand von Beispielen (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I). 	<p>verschiedene Leuchtmuster einer Reihe von Kreisobjekten.</p> <p>Weitere mögliche Projekte: z.B. Hubschrauberlandeplatz, Flugplatz, Allee mit Bäumen</p> <p>Projekt: Binärzahlanzeige (in Anlehnung an die Lampenleiste aus der Sendung mit der Maus)</p>
<p>4. Modellierung und Animation komplexerer grafisch repräsentierbarer Objekte in eigenen Klassen</p> <p>(a) Modellierung eines Simulationsprogramms mit eigenen Klassen, die sich selbst mit Hilfe von einfachen graphischen Objekten zeigen mit Hilfe eines Implementationsdiagramms</p> <p>(b) Implementierung eigener Methoden mit und ohne Parameterübergabe</p> <p>(c) Realisierung von Zustandsvariablen</p> <p>(d) Animation mit Hilfe des Aufrufs von selbstimplementierten Methoden</p>	<p>Nutzung dokumentierter Klassenbibliotheken (I),</p> <ul style="list-style-type: none"> • implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I), • implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I), • testen Programme schrittweise anhand von Beispielen (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I). 	<p>Projekt: z.B. Ampeln</p> <p>Die Schülerinnen und Schüler erstellen eine Ampelkreuzung mit mehreren Ampelobjekten.</p> <p>Weitere Mögliche Projekte: mehrere Uhren, die verschiedene lokale Zeiten anzeigen, City-Szene mit vielen verschiedenen Objektklassen mit eigenen Animationen</p>

Unterrichtsvorhaben EF-IV

Thema: Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand von grafischen Spielen und Simulationen

Zeitbedarf: ca. 30 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Vertiefung des Referenzbegriffs und Einführung des Prinzips der dynamischen Referenzierung</p> <p>(a) Einführung von Assoziationen, die über Referenzübergaben hergestellt werden (Typ „kennt“)</p> <p>(b) Einführung des Geheimnisprinzips und der Sichtbarkeitsmodifikatoren.</p> <p>(c) Einführung von Methoden mit Rückgabe (get-Methoden)</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • analysieren und erläutern eine objektorientierte Modellierung (A), • stellen die Kommunikation zwischen Objekten grafisch dar (M), • ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), 	<p>Projekt: z.B. Windpark</p> <p>Die Schülerinnen und Schüler entwickeln die Simulation eines Windparks aus mehreren Windrädern mit Einführung der unabhängigen Klasse Wind, die von den Windrädern referenziert wird. Kann erweitert werden um eine Klasse Uhrzeit</p>
<p>2. Entwicklung eines Spiels mit der Notwendigkeit von Kollisionskontrollen zwischen zwei oder mehr grafischen Objekten</p> <p>(a) Modellierung des Spiels ohne Berücksichtigung der Kollision mit Hilfe eines Implementationsdiagramms</p> <p>(b) Dokumentation der Klassen des Projekts</p> <p>(c) Implementierung eines Prototypen ohne Kollision</p> <p>(d) Ergänzung einer Kollisionsabfrage durch zusätzliche Assoziationsbeziehungen in Diagramm, Dokumentation und Quellcode</p> <p>(e) Verallgemeinerung der neuen Verwendung von Objektreferenzen</p>	<ul style="list-style-type: none"> • modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M), • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M), • ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M), • modellieren Klassen unter Verwendung von Vererbung (M), 	<p>Projekt: z.B. Ufospiel</p> <p>Die Schülerinnen und Schüler entwickeln die Simulation eines Ufos, das Asteroiden ausweichen soll mit denen eine Kollision möglich ist.</p> <p>Weitere mögliche Projekte: z.B. SaSHuhn (angelehnt an Moorhuhn),</p>

<p>3. Erarbeitung einer Simulation mit grafischen Objekten, die sich durch unterschiedliche Ergänzungen voneinander unterscheiden (Vererbung durch Spezialisierung ohne Überschreiben von Methoden)</p> <p>(a) Analyse und Erläuterung einer Basisversion der grafischen Klasse</p> <p>(b) Realisierung von grafischen Erweiterungen zur Basisklasse mit und ohne Vererbung</p> <p>(c) Verallgemeinerung und Reflexion des Prinzips der Vererbung am Beispiel der Spezialisierung</p>	<ul style="list-style-type: none"> • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I), • testen Programme schrittweise anhand von Beispielen (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I), • modifizieren einfache Algorithmen und Programme (I), • stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D), • dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D). 	<p>Projekt: z.B.: Erweiterung des Ufospiels um spezielle Unterklassen bei den Asteroiden.</p> <p>Weitere mögliche Projekte: z.B.</p> <p>Turnverein: Turner führen verschiedene Übungsausführungen durch</p> <p>Schneemann: Die Schülerinnen und Schüler erstellen eine Simulation von Schneemännern, die unterschiedliche Kopfbedeckungen tragen.</p>
<p>4. Entwicklung einer komplexeren Simulation mit grafischen Elementen, die unterschiedliche Animationen durchführen (Vererbung mit Überschreiben von Methoden)</p> <p>(a) Analyse und Erläuterung einer einfachen grafischen Animationsklasse</p> <p>(b) Spezialisierung der Klasse zu Unterklassen mit verschiedenen Animationen durch Überschreiben der entsprechenden Animationsmethode</p> <p>(c) Reflexion des Prinzips der späten Bindung</p>		<p>Projekt: z.B. Flummibälle</p> <p>Die Schülerinnen und Schüler entwickeln eine Simulation von Flummibällen, bei der unterschiedliche Bälle unterschiedliche Bewegungen durchführen.</p>

Unterrichtsvorhaben EF-V

Thema: Such- und Sortialgorithmen anhand kontextbezogener Beispiele

Zeitbedarf: ca. 12 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>Binäre Suche auf sortierten Daten</p> <ul style="list-style-type: none"> (a) Suchaufgaben im Alltag und im Kontext informatischer Systeme (b) Evtl. Simulationsspiel zum effizienten Suchen mit binärer Suche (c) Implementierung der binären Suche <p>Effizienzbetrachtungen zur binären Suche</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • beurteilen die Effizienz von Algorithmen am Beispiel von Sortierverfahren hinsichtlich Zeit und Speicherplatzbedarf (A), • entwerfen einen weiteren Algorithmus zum Sortieren (M), • analysieren Such- und Sortieralgorithmen und wenden sie auf Beispiele an (D). 	<p>Projekt: z.B. Helferprogramm für das Zahlenratespiel</p> <p>Projekt: z.B. NutzerIDs am Kopierer</p>
<p>Explorative Erarbeitung eines Sortierverfahrens</p> <ul style="list-style-type: none"> (a) Sortierprobleme im Kontext informatischer Systeme und im Alltag (z.B. Dateisortierung, Tabellenkalkulation, Telefonbuch, Bundesligatabelle, usw.) (b) Vergleich zweier Elemente als Grundlage eines Sortieralgorithmus (c) Erarbeitung eines Sortieralgorithmus durch die Schülerinnen und Schüler 		<p>Material: z.B.: Kartenspiele</p>
<p>Systematisierung von Algorithmen und Effizienzbetrachtungen</p> <ul style="list-style-type: none"> (a) Formulierung und Erläuterung von mehreren Algorithmen im Pseudocode (Sortieren durch Einfügen, Sortieren durch Vertauschen, Sortieren durch Auswählen) (b) Bewertung von Algorithmen anhand der Anzahl der nötigen Vergleiche (c) Effizienzbetrachtungen an einem konkreten Beispiel bezüglich der Rechenzeit und des Speicherplatzbedarfs 		<p>Projekt: z.B. Bücherregal</p> <p>Schülerinnen und Schüler visualisieren die Abläufe der Sortieralgorithmen</p>

Unterrichtsvorhaben EF-VI

Thema: Digitalisierung und Grundlagen des Datenschutzes

Zeitbedarf: ca. 6 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>Binärcodierungen</p> <ul style="list-style-type: none"> (a) natürliche Zahlen, (b) Ganzzahlen, (c) Zeichen 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • stellen ganze Zahlen und Zeichen in Binärcodes dar (D), • interpretieren Binärcodes als Zahlen und Zeichen (D), • bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A), 	<p>(Kann jederzeit an passender Stelle in das Unterrichtsgeschehen eingefügt werden)</p>
<p>Betrachtung des Themas Datenschutz</p> <p>Selbstentdeckendes Erkunden:</p> <ul style="list-style-type: none"> (a) Umfang von Datensammlungen, (b) Vorgehen bei Datensammlungen, (c) wirtschaftlichen Interessen bei Datensammlungen (d) Motivation der Akteure <p>Problematisierung und Anknüpfung an die Lebenswelt der Schülerinnen und Schüler</p>		<p>Material: Onlinespiel DataDealer</p> <p>(Kann jederzeit an passender Stelle in das Unterrichtsgeschehen eingefügt werden)</p>