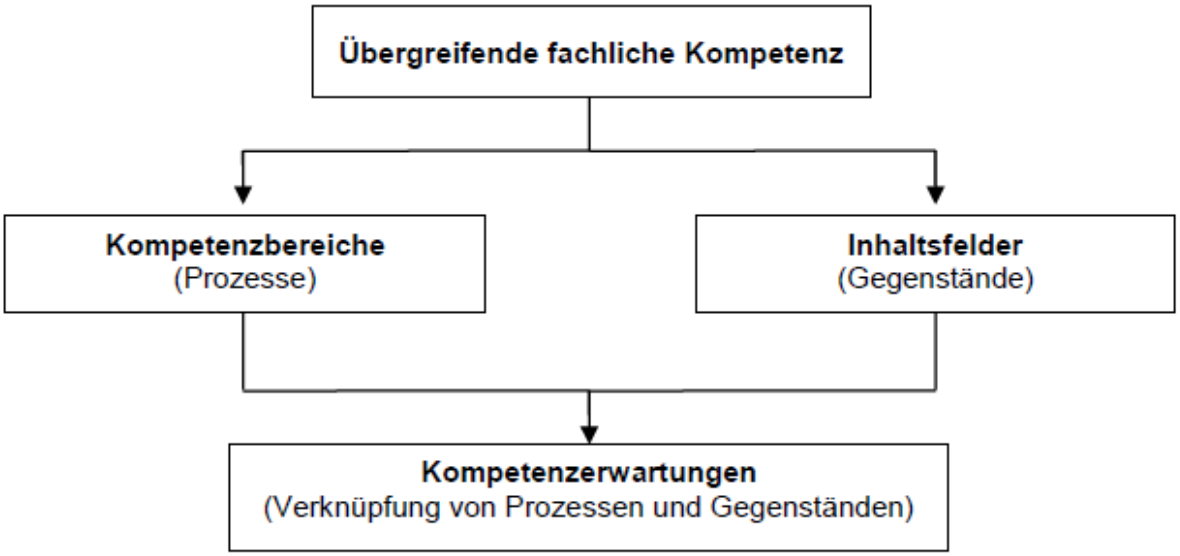
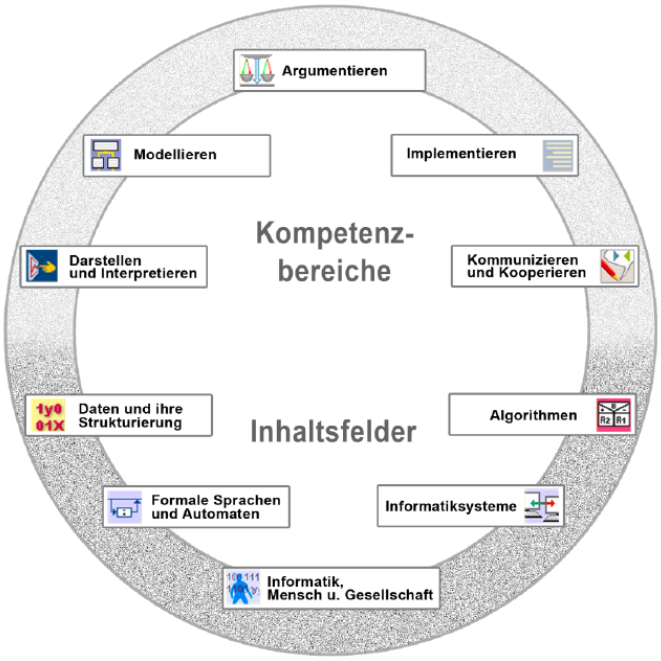
Informatik - Jahrgang EF **Mai 2015**

Inhaltsfelder und Kompetenzbereiche orientieren sich am ***Kernlehrplan für die Sekundarstufe II Gymnasium /Gesamtschule in Nordrhein-Westfalen (2013).***



Kompetenzbereiche repräsentieren die Grunddimensionen des fachlichen Handelns. Sie dienen dazu, die einzelnen Teiloperationen entlang der fachlichen Kerne zu strukturieren und den Zugriff für die am Lehr-Lernprozess Beteiligten zu verdeutlichen.
Inhaltsfelder systematisieren mit ihren jeweiligen inhaltlichen Schwerpunkten die im Unterricht der gymnasialen Oberstufe verbindlichen und unverzichtbaren Gegenstände und liefern Hinweise für die inhaltliche Ausrichtung des Lehrens und Lernens.
Kompetenzerwartungen führen Prozesse und Gegenstände zusammen und beschreiben die fachlichen Anforderungen und intendierten Lernergebnisse, die auf zwei Stufen bis zum Ende der Sekundarstufe II erreicht werden sollen.



Jahrgang EF

Inhaltsfelder	Kompetenzbereiche und konkretisierte Kompetenzerwartung
Daten und ihre Strukturierung	**Die Schülerinnen und Schüler** • 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), • modellieren Klassen unter Verwendung von Vererbung (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), • stellen den Zustand eines Objekts dar (D), • stellen die Kommunikation zwischen Objekten grafisch dar (M), • stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D), • dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D), • analysieren und erläutern eine objektorientierte Modellierung (A), • implementieren Klassen in Java insbesondere auch unter Nutzung dokumentierter Klassenbibliotheken (I).

Algorithmen	**<u>Analyse, Entwurf und Implementierung einfacher Algorithmen</u>:** **Die Schülerinnen und Schüler** • analysieren und erläutern einfache Algorithmen und Programme (A), • modifizieren einfache Algorithmen und Programme (I), • entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M), • implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I), • testen Programme schrittweise anhand von Beispielen (I). **<u>Algorithmen zum Suchen und Sortieren:</u>** **Die Schülerinnen und Schüler** • analysieren Such- und Sortieralgorithmen und wenden sie auf Beispiele an (D), • entwerfen einen weiteren Algorithmus zum Sortieren (M), • beurteilen die Effizienz von Algorithmen am Beispiel von Sortierverfahren hinsichtlich Zeitaufwand und Speicherplatzbedarf (A). **<u>Grundlegende Modellierungstechniken im Rahmen der Algorithmik:</u>** Modellierung von Abläufen mit Algorithmen insbesondere: Algorithmusbegriff, Ablaufstrukturen, einfache und höhere Datenstrukturen, Zerlegen in Teilalgorithmen; Struktogramme; spezielle Verfahren (z. B. Rekursion) u.a.: Modularisierung, Parameter, Rückgabewerte, Schleifen, bedingte Anweisungen, Pseudocode, Struktogramme nach Nassi-Shneiderman, Flussdiagramme, einfache Datentypen
Formale Sprachen und Automaten	**<u>Syntax und Semantik einer Programmiersprache:</u>** Die Schülerinnen und Schüler • implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I).

Informatiksysteme	**<u>Digitalisierung:</u>** Die Schülerinnen und Schüler • stellen ganze Zahlen und Zeichen in Binärcodes dar (D), • interpretieren Binärcodes als Zahlen und Zeichen (D). **<u>Einzelrechner:</u>** Die Schülerinnen und Schüler beschreiben und erläutern den strukturellen Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der „Von- Neumann-Architektur“ (A). **<u>Dateisystem:</u>** Die Schülerinnen und Schüler nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung und gemeinsamen Verwendung von Daten unter Berücksichtigung der Rechteverwaltung (K). **<u>Internet:</u>** Die Schülerinnen und Schüler nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K).
Informatik, Mensch und Gesellschaft	**<u>Einsatz von Informatiksystemen:</u>** Die Schülerinnen und Schüler nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D). **<u>Wirkungen der Automatisierung:</u>** Die Schülerinnen und Schüler bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A). **<u>Geschichte der automatischen Datenverarbeitung:</u>** Die Schülerinnen und Schüler erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (A). **<u>Allgemein:</u>** Die Schülerinnen und Schüler können typische Einsatzbereiche, Möglichkeiten, Grenzen, Chancen und Risiken von Informations- und Kommunikationssystemen untersuchen und einschätzen u.a.: Social Networking (Twitter, Facebook, …), Datenschutz, Uhrheberrechte

Allgemein: **Kommunizieren und Kooperieren**

Die Schülerinnen und Schüler

- können im Team arbeiten
- organisieren und koordinieren die Arbeit in Projektgruppen
- verwenden die Fachsprache angemessen
- veranschaulichen und beschreiben Sachverhalte u. a. mit Hilfe von Texten und Diagrammen
- können den Arbeitsablauf und die Arbeitsergebnisse dokumentieren
- können Lern- und Arbeitsergebnisse adressatengerecht präsentieren

Beurteilungsbereich „Sonstige Leistungen im Unterricht/ Sonstige Mitarbeit“

Im Beurteilungsbereich „Sonstige Leistungen im Unterricht/ Sonstige Mitarbeit“ können – neben den nachfolgend aufgeführten Überprüfungsformen – vielfältige weitere zum Einsatz kommen, für die kein abschließender Katalog festgesetzt wird. Im Rahmen der Leistungsbewertung gelten auch für diese die oben ausgeführten allgemeinen Ansprüche der Lernerfolgsüberprüfung und Leistungsbewertung. Im Verlauf der gymnasialen Oberstufe ist auch in diesem Beurteilungsbereich sicherzustellen, dass Formen, die im Rahmen der Abiturprüfungen – insbesondere in den mündlichen Prüfungen – von Bedeutung sind, frühzeitig vorbereitet und angewendet werden.

Zu den Bestandteilen der „Sonstigen Leistungen im Unterricht/Sonstigen Mitarbeit“ zählen u. a. unterschiedliche Formen der selbstständigen und kooperativen Aufgabenerfüllung, Beiträge zum Unterricht, von der Lehrkraft abgerufene Leistungsnachweise wie z.B. die schriftliche Übung, von der Schülerin oder dem Schüler vorbereitete, in abgeschlossener Form eingebrachte Elemente zur Unterrichtsarbeit, die z.B. in Form von Präsentationen, Protokollen, Referaten und Portfolios möglich werden. Schülerinnen und Schüler bekommen durch die Verwendung einer Vielzahl von unterschiedlichen Überprüfungsformen vielfältige Möglichkeiten, ihre eigene Kompetenzentwicklung darzustellen und zu dokumentieren.

Der Bewertungsbereich „Sonstige Leistungen im Unterricht/Sonstige Mit-arbeit“ erfasst die im Unterrichtsgeschehen durch mündliche, schriftliche und ggf. praktische Beiträge sichtbare Kompetenzentwicklung der Schülerinnen und Schüler. Der Stand der Kompetenzentwicklung in der „Sonstigen Mitarbeit“ wird sowohl durch Beobachtung während des Schuljahres (Prozess der Kompetenzentwicklung) als auch durch punktuelle Überprüfungen (Stand der Kompetenzentwicklung) festgestellt.

Überprüfungsformen

Die Kompetenzerwartungen des Kernlehrplans ermöglichen eine Vielzahl von Überprüfungsformen. Im Verlauf der gesamten gymnasialen Oberstufe soll – auch mit Blick auf die individuelle Förderung – ein möglichst breites Spektrum der genannten Formen in schriftlichen, mündlichen oder praktischen Kontexten zum Einsatz gebracht werden. Darüber hinaus können weitere Überprüfungsformen nach Entscheidung der Lehrkraft eingesetzt werden. Wichtig für die Nutzung der Überprüfungsformen im Rahmen der Leistungsbewertung ist es, dass sich die

Schülerinnen und Schüler zuvor im Rahmen von Anwendungssituationen hinreichend mit diesen vertraut machen konnten. Weitere über die Auflistung hinaus gehende Überprüfungsformen sind möglich.

Überprüfungsform I	Analyse und Eingrenzung einer kontextbezogenen Problemstellung und Entwicklung eines Modells oder Teilmodells mit erläuternden Begründungen der Entwurfsentscheidungen
Überprüfungsform II	Analyse, Erläuterung und Modifikation eines vorgegebenen informatischen Modells sowie die vergleichende Beurteilung unterschiedlicher Entwürfe
Überprüfungsform III	Vollständige oder teilweise Implementation einer bereits modellierten Problemstellung
Überprüfungsform IV	Entwurf und formale Darstellung von Algorithmen zu einer vorgegebenen informatischen Problemstellung
Überprüfungsform V	Analyse und Erläuterung von vorgegebenen Algorithmen oder Programmausschnitten
Überprüfungsform VI	Interpretation gegebener textueller, grafischer oder formaler Darstellungen informatischer Zusammenhänge und deren Überführung in eine andere Darstellungsform
Überprüfungsform VII	Darstellung, Erläuterung und sachgerechte Anwendung von informatischen Begriffen, Verfahren und Lösungsstrategien
Überprüfungsform VIII	Analyse und Beurteilung einer Problemlösung oder eines Informatiksystems nach vorgegebenen oder eigenen Kriterien
Überprüfungsform IX	Analyse und Bewertung des Einsatzes eines Informatiksystems in Bezug auf ethische, rechtliche oder gesellschaftliche Fragestellungen

Inhaltsfelder und Werkzeuge im Detail der Unterrichtsplanungen

Objektorientierte Modellierungen im dreidimensionalen Raum

Die Java-Klassenbibliothek **GLOOP** (Graphics Library for object oriented programming) wurde entwickelt für die Verdeutlichung objektorientierter Zusammenhänge beim Einstieg in die Objektorientierung im Unterricht der gymnasialen Einführungsphase und bietet die Möglichkeit eines didaktisch vereinfachten und intuitiven objektorientierten Zugangs zur dreidimensionalen Grafikprogrammierung mit OpenGL.

In einem virtuellen dreidimensionalen Raum können Objekte positioniert und bewegt werden. Hierfür stehen unter anderem die Klassen "GLKugel", "GLQuader", "GLZylinder", "GLWuerfel", "GLKegel", "GLKegelstumpf", "GLTorus" und "GLPrismoid" zur Verfügung. Sie sind Unterklassen der Klasse "GLObjekt". Durch Setzen einer Textur können die Oberflächen von Objekten gestaltet werden. Um die Objekte sehen zu können, bedarf es allerdings noch einer Kamera und einer Lichtquelle die die Szene ausleuchtet (Objekte der ebenfalls zur Verfügung stehenden Klassen "GLKamera" und "GLLicht"). Für die Programmsteuerung, Benutzereingaben, Ausgaben und Rechnungen stehen die Hilfsklassen "GLMaus", "GLTastatur", "Sys" und "GLVektor" zur Verfügung.

Im Sinne einer Vergleichbarkeit der Kurse sind folgende Beispielprojekte ohne Angabe einer Reihenfolge mit GLOOP zu empfehlen:

- Erde- Mond- System
- Aufbau einer Häuserzeile mit jeweils mehreren Etagen sowie Türen und Fenstern
- Skulpturenpark

Alle benötigten Dateien und Beispiele sind im System vorinstalliert bzw. befinden sich im Fachschaftslaufwerk.

Oberflächenprogrammierung mit Swing und AWT

Bei **Swing** handelt es sich um eine Programmierschnittstelle (API) und Grafikbibliothek zum Programmieren von grafischen Benutzeroberflächen. Das Abstract Window Toolkit (**AWT**) ist Bestandteil der Java Foundation Classes (JFC) und stellt eine Standard-API zur Erzeugung und Darstellung einer plattformunabhängigen grafischen Benutzerschnittstelle (GUI) für Java-Programme dar.

Im Sinne einer Vergleichbarkeit der Kurse sind folgende Beispielprojekte ohne Angabe einer Reihenfolge mit Swing/ AWTzu empfehlen:

- vereinfachter Taschenrechner
- Klickspiele wie Trisentis, TIC- TAC- TOE, Mindsweeper, Schiebepuzzle
- Warenkorb mit Kundendaten

Alle benötigten Dateien und Beispiele sind im System vorinstalliert bzw. befinden sich im Fachschaftslaufwerk.

Sequenzierung der Unterrichtsvorhaben
Die Konkretisierungen der Unterrichtsvorhaben und der Aufbau der Unterrichtssequenzen orientieren sich am schulinternen Lehrplan der Fachgruppe Informatik des Konrad- Zuse- Gymnasiums Paderborn und Veröffentlichungen der Gesellschaft für Informatik. Die detaillierten Konkretisierungen, verschiedene Beispiele für die Unterrichtspraxis und die Verankerung der Sequenzen im Verlauf der gesamten Oberstufe werden von der Fachgruppe Informatik des Kopernikus Gymnasium Rheine spezifiziert.

Unterrichtsvorhaben EF- I (6 Stunden)
Das erste Unterrichtsvorhaben stellt eine allgemeine Einführung in das Fach Informatik dar. Dabei ist zu berücksichtigen, dass für manche Schülerinnen und Schüler in der Einführungsphase der erste Kontakt mit dem Unterrichtsfach Informatik stattfindet, so dass zu Beginn Grundlagen des Fachs behandelt werden müssen.

Unterrichtssequenz	Kompetenzerwartung	Beispiel
Informatik als Wissenschaft der Verarbeitung von Informationen Darstellung von Informationen in Schrift, Bild und Ton Speichern von Daten mit informatischen Systemen am Beispiel der Schulrechner Vereinbarung von Richtlinien zur Datenspeicherung auf den Schulrechnern (z.B. Ordnerstruktur, Dateibezeichner usw.) Informatische Kommunikation in Rechnernetzen am Beispiel des Schulnetzwerks (z.B. Benutzeranmeldung, Netzwerkordner, Zugriffsrechte, Client-Server) Richtlinien zum verantwortungsvollen Umgang mit dem Internet Identifikation typischer Komponenten informatischer Systeme und anschließende Beschränkung auf das Wesentliche, Herleitung der „Von-Neumann-Architektur“	Die Schülerinnen und Schüler • 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), • nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K).	Textkodierung, Bildkodierung Demontage eines Demonstrationsrechners

Unterrichtsvorhaben EF- II (40 Stunden)

Ein zentraler Bestandteil des Informatikunterrichts der Einführungsphase ist die Objektorientierte Programmierung. Dieses Unterrichtsvorhaben führt in die Grundlagen der Analyse, Modellierung und Implementierung in diesem Kontext ein. Des Weiteren werden einfache algorithmische Grundstrukturen der Programmiersprache Java sowie verschiedene Datentypen und Datenstrukturen eingeführt.

Dazu werden zunächst konkrete Gegenstandsbereiche aus der Lebenswelt der Schülerinnen und Schüler analysiert und im Sinne des Objektorientierten Paradigmas strukturiert. Dabei werden die grundlegenden Begriffe der Objektorientierung und Modellierungswerkzeuge wie Objektkarten, Klassenkarten oder Beziehungsdiagramme eingeführt.

Für die Realisierung erster Projekte wird direkt mit der didaktischen Programmierumgebung GLOOP begonnen. Die von der Bibliothek vorgegebenen Klassen werden von Schülerinnen und Schülern in Teilen analysiert und entsprechende Objekte anhand einfacher Problemstellungen *(beispielsweise Skulpturengarten, Spielfiguren beim Schach, Straßenschilder)* erprobt. Dazu muss der grundlegende Aufbau einer Java-Klasse thematisiert und zwischen Deklaration, Initialisierung und Methodenaufrufen unterschieden werden.

Eine Besonderheit dieses Unterrichtsvorhabens liegt auf der Entwicklung mehrerer Projekte, die durch Eingaben des Benutzers gesteuerte Animationen aufweisen. Zunächst wird ein Projekt bearbeitet, das in Anlehnung an das vorangegangene Unterrichtsvorhaben eine Szene darstellt, die lediglich aus Objekten besteht, zu denen das didaktische System Klassen vorgibt. Einzelne Objekte der Szene werden animiert, um ein einfaches Spiel zu realisieren (*beispielsweise Dart, Mensch-Ärgere-Dich-Nicht)* oder die Szene (*beispielsweise Autofahren an einer Ampelkreuzung*) optisch aufzuwerten. Für die Umsetzung dieses Projekts werden Kontrollstrukturen in Form von Schleifen und Verzweigungen benötigt und eingeführt.

Sind an einem solchen Beispiel im Schwerpunkt Schleifen und Verzweigungen eingeführt worden, sollen diese Konzepte an weiteren Beispielprojekten eingeübt werden. Dabei muss es sich nicht zwangsläufig um solche handeln, bei denen Kontrollstrukturen lediglich zur Animation verwendet werden. Auch die Erzeugung größerer Mengen grafischer Objekte und deren Verwaltung in einem Feld (*Etagenbau bei Häusern*) kann ein Anlass zur Verwendung und Einführung von Kontrollstrukturen sein.

Das Unterrichtsvorhaben schließt mit einem Projekt, das komplexere grafische Elemente beinhaltet, so dass die Schülerinnen und

Schüler mehr als nur die Klasse erstellen müssen, welche die Szene als Ganzes darstellt. Elemente der Szene müssen zu sinnhaften eigenen Klassen zusammengefasst werden, die dann ihre eigenen Attribute und Dienste besitzen. Auch dieses Projekt soll eine Animation, ggf. im Sinne einer Simulation, sein, bei der Attributwerte von Objekten eigener Klassen verändert werden und diese Veränderungen optisch sichtbar gemacht werden.

Komplexere Assoziationsbeziehungen zwischen Klassen werden in diesem Unterrichtsvorhaben erst in den späteren Projekten behandelt. Sie stellen den Schwerpunkt des Unterrichtsvorhabens EF- III dar.

Unterrichtssequenz	Kompetenzerwartung	Beispiel
Identifikation von Objekten Am Beispiel eines lebensweltnahen Beispiels werden Objekte im Sinne der Objektorientierten Modellierung eingeführt. Objekte werden mit Objektkarten visualisiert und mit sinnvollen Attributen und „Fähigkeiten“, d.h. Methoden versehen. Manche Objekte sind prinzipiell typgleich und werden so zu einer Objektsorte bzw. Objektklasse zusammengefasst. **Analyse von Klassen didaktischer Lernumgebungen** Objektorientierte Programmierung als modularisiertes Vorgehen (Entwicklung von Problemlösungen auf Grundlage vorhandener Klassen) Teilanalyse der Klassen der didaktischen Lernumgebungen GLOOP **Implementierung dreidimensionaler, statischer Szenen** Grundaufbau einer Java-Klasse Konzeption einer Szene mit Kamera, Licht und sichtbaren Objekten Deklaration und Initialisierung von Objekten Methodenaufrufe mit Parameterübergabe zur Manipulation von	Die Schülerinnen und Schüler • 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), • stellen die Kommunikation zwischen Objekten grafisch dar (M), • implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I), • stellen den Zustand eines Objekts dar (D). • 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	Beispiel: Skulpturengarten Beispiel: Sonnensystem Beispiel:Olympische Ringe Beispiel: Schachbrett Beispiel: Spielbretter Beispiel: Kerzensimulation Beispiel: Uhren Beispiel: Ampeln Beispiel: Ampelkreuzung Beispiel: Wurfspiel Beispiel: Ufospiel Beispiel: Dart

Objekteigenschaften (z.B. Farbe, Position, Drehung) **Bewegungsanimationen am Beispiel einfacher grafischer Objekte (GLObjekte)** Kontinuierliche Verschiebung eines GLObjekts mit Hilfe einer Schleife (While-Schleife) Tastaturabfrage zur Realisierung einer Schleifenbedingung für eine Animationsschleife Mehrstufige Animationen mit mehreren sequenziellen Schleifen Meldungen zur Kollision zweier GLObjekte mit Hilfe von Abstandsberechnungen und Verzweigungen (IF-Anweisungen) **Erstellen und Verwalten größerer Mengen einfacher grafischer Objekte (GLObjekte)** Erzeugung von Objekten mit Hilfe von Zählschleifen (FOR-Schleife) Verwaltung von Objekten in eindimensionalen Feldern (Arrays) Animation von Objekten, die in eindimensionalen Feldern (Arrays) verwaltet werden **Modellierung und Animation komplexerer grafisch repräsentierbarer Objekte** Modellierung eines Simulationsprogramms mit eigenen Klassen, die sich selbst mit Hilfe von einfachen GLObjekten zeigen mit Hilfe eines Implementationsdiagramms Implementierung eigener Methoden mit und ohne Parameterübergabe Realisierung von Zustandsvariablen Thematisierung des Geheimnisprinzips und des Autonomitätsprinzips von Objekten	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), • ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M), • modifizieren einfache Algorithmen und Programme (I), • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I), • 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).	Beispiel: Billardkugeln Beispiel: Autospiel Beispiel: Schneemann Diese Beispiele sind nur eine Empfehlung. Alle benötigten Dateien und Beispiele sind im System vorinstalliert bzw. befinden sich im Fachschaftslaufwerk.

Unterrichtsvorhaben EF- III (24 Stunden)

Dieses Unterrichtsvorhaben beschäftigt sich im Schwerpunkt mit dem Aufbau komplexerer Objektbeziehungen und größeren Projektvorhaben. Während in den meisten vorangegangenen Unterrichtsvorhaben Objekte nur jeweils solchen Objekten Nachrichten schicken konnten, die sie selbst erstellt haben, soll spätestens in diesem Unterrichtsvorhaben an einzelnen Beispielen diese hierarchische Struktur aufgebrochen werden.

Dazu bedarf es zunächst einer präzisen Unterscheidung zwischen Objektreferenzen und Objekten, so dass klar wird, dass Dienste eines Objektes von unterschiedlichen Objekten über unterschiedliche Referenzen in Anspruch genommen werden können. Auch der Aufbau solcher Objektbeziehungen muss thematisiert werden. Des Weiteren wird das Prinzip der Vererbung im objektorientierten Sinne angesprochen. Zunächst wird die Vererbung als Spezialisierung im Sinne einer einfachen Erweiterung einer Oberklasse vorgestellt. Darauf folgt ein Projekt, welches das Verständnis von Vererbung um den Aspekt der späten Bindung erweitert, indem Dienste einer Oberklasse überschrieben werden. Modellierungen sollen in Form von Implementationsdiagrammen erstellt werden.

Dieses Unterrichtsvorhaben kann (sollte) losgelöst von der GLOOP- Umgebung behandelt werden und – kann falls graphisch gewünscht – auch in Java- Swing (oder AWT) umgesetzt werden.

Unterrichtssequenz	**Kompetenzerwartung**	**Beispiel**
Entwicklung eines Spiels mit der Notwendigkeit von Kollisionskontrollen zwischen zwei oder mehr grafischen Objekten (Siehe auch EF- II) **Erarbeitung einer Simulation mit grafischen Objekten, die sich durch unterschiedliche Ergänzungen voneinander unterscheiden (Vererbung durch Spezialisierung ohne Überschreiben von Methoden) (Siehe auch EF- II)** Analyse und Erläuterung einer Basisversion einer Klasse Realisierung von Erweiterungen zur Basisklasse mit und ohne Vererbung (Implementationsdiagramm und Quellcode) Verallgemeinerung und Reflexion des Prinzips der Vererbung am Beispiel der Spezialisierung **Entwicklung einer komplexeren Simulation mit grafischen Elementen,**	Die Schülerinnen und Schüler • 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), • 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	Beispiele: Ausarbeitung der Beispiele aus EF- II Beispiel: Warenkorb mit Kundendaten Beispiel: Bibliothek

die unterschiedliche Animationen durchführen (Vererbung mit Überschreiben von Methoden) Analyse und Erläuterung einer einfachen grafischen Animationsklasse Spezialisierung der Klasse zu Unterklassen mit verschiedenen Erweiterungen durch Überschreiben der entsprechenden Methode Reflexion des Prinzips der späten Bindung	Vererbung (M), • 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)	

Unterrichtsvorhaben EF – Folgende Themen werden in die bestehenden Unterrichtsvorhaben EF- II und EF- III eingebettet

- Unterrichtsvorhaben EF- Zusatz I beschäftigt sich mit der Erarbeitung von Such- und Sortieralgorithmen. Der Schwerpunkt des Vorhabens liegt dabei auf den Algorithmen und einer möglichen Nutzung selbst und nicht auf deren Implementierung in einer Programmiersprache. Im Wesentlichen nutzen die Schülerinnen und Schüler mögliche Einsatzszenarien für Such- und Sortieralgorithmen, um sich der Bedeutung einer effizienten Lösung dieser Probleme bewusst zu werden. Des Weiteren soll das Prinzip der *binären Suche* behandelt und nach Effizienzgesichtspunkten untersucht werden.

- Das Unterrichtsvorhaben EF- Zusatz II behandelt Fragen aus dem Kontext der Geschichte der Datenverarbeitung und insbesondere den daraus sich ergebenen Fragen des Datenschutzes. Eingebettet werden kann dieser Themenblock beispielsweise bei den Projekten *Warenkorb* und *Verwaltung einer Bibliothek* in EF- III. Anschließend wird verstärkt auf den Aspekt des Datenschutzes eingegangen. Dazu wird das Bundesdatenschutzgesetz in Auszügen behandelt und auf schülernahe Beispielsituationen zur Anwendung gebracht. Dabei steht keine formale juristische Bewertung der Beispielsituationen im Vordergrund, die im Rahmen eines Informatikunterrichts auch nicht geleistet werden kann, sondern vielmehr eine persönliche Einschätzung von Fällen im Geiste des Datenschutzgesetzes.