

# Objektorientierte Modellierung und Programmierung

Zuordnung	Algorithmen
Klassenstufe	Kurstufe BF und LF
Bildungsplanbezug	BF, LF
Werkzeug	Java-Editor, Greenfoot
Autoren	T. Schaller, U.Lautebach

## Inhalt

Schülerinnen und Schüler lernen anhand eines Softwareprojekts zur Erstellung einer Geometrie-Software die Grundlagen der objektorientierten Programmierung und Modellierung kennen. Anschließend modellieren sie ein einfaches Spiel.

## Themen

- Klassen und Objekte
- Attribute und Methoden
- primitive Variablen, Referenzvariablen
- Deklaration einer Klasse
- Lebenszyklus von Objekten / Garbage Collector
- abstrakte Klassen
- TypCasts
- Polymorphie
- Klassenbeziehungen (Vererbung, Aggregation bzw. Komposition, Assoziation)
- von UML zum Quelltext
- Graphical User Interface (GUI)
- Modellierung von Klassen
- Software-Projekte
- Sequenzdiagramme
- UML-Klassendiagramme
- for-each-Schleifen

# Bildungsplan-Bezüge

## 3.1.3 Algorithmen - Strukturierte und objektorientierte Programmierung - Basisfach und Leistungsfach

### Variablen und Datentypen

- (3) implizite und explizite Typumwandlungen (type casts) zur Konvertierung von Werten unterschiedlicher Datentypen ineinander verwenden und dabei auftretende Probleme beschreiben

### Objektorientierung

- (5) eigene Klassendefinitionen mit Attributen und Methoden implementieren
- (6) Referenzvariablen/Referenztypen mit primitiven Variablen / primitiven Datentypen vergleichen
- (7) mögliche Ursachen und Auswirkungen von Nullpointerfehlern erläutern [nur LF]
- (9) die Funktion von Zugriffsmodifikatoren (public, private, protected) erläutern und diese verwenden
- (12) den Lebenszyklus von Objekten (unter anderem Instanziierung, Verwendung, Zerstörung) beschreiben [nur LF]
- (13) die Rolle von Konstruktoren beim Erzeugen von Instanzen einer Klasse (Objekten) erläutern und diese implementieren
- (14) generische Datentypen bei der Instanziierung vorgegebener Klassen verwenden [nur LF]
- (15) sprachliche Mittel zur Sammlung gleichartiger Objekte (Collection) und zum Durchlaufen aller Elemente der Sammlung (zum Beispiel Iteration, foreach, Lambdaausdrücke etc.) nutzen [nur LF]
- (16) das Konzept der Vererbung erläutern und in Modellierungen und Implementierungen anwenden
- (17) das Prinzip der Polymorphie erläutern und anwenden und dabei auch sichere class casts [nur LF] verwenden
- (18) abstrakte Klassen und abstrakte Methoden nutzen [nur LF]

### Modellierung und Dokumentation

- (20) UML-Klassendiagramme mit Vererbung und Assoziation (auch rekursiv) entwerfen und zur Modellierung nutzen