

Strukturierte und objektorientierte Programmierung (3.x.2.1)

Dieser Stoffverteilungsplan stellt einen möglichen Unterrichtsgang für die Themen „generische Datentypen“, „Sammlungen (Collections)“ und „Durchlaufen einer Sammlung (Iteration, foreach, Lambdaausdrücke etc)“ tabellarisch dar und kennzeichnet welche Inhalte auf jeden Fall behandelt werden sollten (x) oder optional zur Verfügung stehen (o). Für das Beifach (BF) fordert der Bildungsplan eigentlich nichts zu den Themen. Es bietet es sich allerdings an, zumindest eine Unterrichtsstunde für eine Demonstration der Notwendigkeit generischer Datentypen zu verwenden (s. generische Implementierung einfach verketteter Listen (3.2.1.2)).

Es wird hierbei vorausgesetzt, dass die Grundlagen der Objektorientierung (Klasse, Attribute, Methoden mit Parametern und Rückgabewerten, Datentypen und Typumwandlung, Methodensignatur, Zugriffsmodifikation insbesondere Setter/Getter, Kapselung, Konstruktoren und Instanziierung) bereits bekannt und eingeübt sind.

Es gibt keine vorgefertigten Arbeitsblätter. Programmieraufträge finden sich im Projekt „Filmsammlung“, eine ausführlichere Beschreibung mit möglichen *Aufgaben* zu den **Inhalten** im Unterrichtsverlauf.

Std.	Inhaltsbezogene Kompetenzen	Inhalt Programmieraufträge Aufgaben	BF	LF
0-1	3.x.1.2 (3) implizite und explizite Typumwandlungen (type casts) [...] verwenden und dabei auftretende Probleme beschreiben	1. Wiederholung: Kennenlernen des Projekts, Begriffsklärungen zu Typisierung/Typumwandlung einschränkende und erweiternde Typumwandlung primitiver Datentypen und Referenztypen. Typinferenz (optional Fachbegriff: Zieltyp) optional (für Lambdaausdrücke Std.): Begriffsklärung formale (Wert)Parameter, konkrete Argumente bei Methoden TODO: Bounds Check TODO: Korrekte Filmanzahl TODO: Suches TODO: Getter	o	x
2-3	3.3.1.2 (14) generische Datentypen bei der Instanziierung vorgegebener Klassen verwenden	2. Generische Datentypen – Notwendigkeit, Deklaration Aufgabe.a1 <i>Aufgabe 1-6</i> <i>Aufgabe 7 (o): generischen Datentyp selbst deklarieren</i>	o	x
		3. Generische Datentypen – Nutzung, Bounds Fachbegriffe: generischer Typ, formaler Typparameter, parametrisierter Typ, konkreter Typparameter, Originaltyp, konkretes Typargument Diamant-Operator, Instanziierung (Nutzung) <i>Aufgabe 1-4</i> <i>Aufgabe.a2</i> <i>TODO: Typsicherheit garantieren</i> Bounds <i>Aufgabe 6 oder 7</i> <i>Aufgabe.a3</i>		x



4-5	<p>3.x.1.2 (22) Dokumentationen zu gegebenem Quellcode und Bibliotheken nutzen</p> <p>3.3.1.2 (15) sprachliche Mittel [...] zum Beispiel Iteration, [...] nutzen.</p>	<p>4. Interfaces, Iterable – Arbeit mit der Dokumentation</p> <p>Interface Arbeiten mit der offiziellen Java API Dokumentation: <code>Iterable<T></code>, <code>Iterator<E></code> <i>Aufgabe 1 und 2</i></p>		X
	<p>3.x.1.2 (22) Dokumentationen zu gegebenem Quellcode und Bibliotheken nutzen</p> <p>3.3.1.2 (15) sprachliche Mittel zur Sammlung gleichartiger Objekte (Collection) und zum Durchlaufen [...] (zum Beispiel Iteration, foreach, [...]) nutzen</p>	<p>5. Das Collections Framework und Iteration</p> <p>Arbeiten mit der offiziellen Java API Dokumentation: <code>Collections-Framework</code>, <code>Iterable<T></code>, <code>Iterator<E></code> <i>Aufgabenauswahl 1-4</i> Iteration auf Basis von <code>Iterable<T></code>: while-Schleife, for-Schleife, erweiterte for-Schleife (foreach) <i>Aufgabe.a4 Aufgabe 5</i> <i>Aufgabe.a5</i> Vererbung bei generischen Typen</p>		X
6-7	<p>3.x.1.2 (22) Dokumentationen zu gegebenem Quellcode und Bibliotheken nutzen</p> <p>3.3.1.2 (15) sprachliche Mittel [...] zum Beispiel [...] Lambdaausdrücke [...] nutzen.</p>	<p>6. Lambdaausdrücke, Funktionales Interface, Iteration</p> <p>funktionales Interface Notation: Lambdaausdruck <i>Aufgaben 1-5 zur Inspiration für eigene Aufgaben</i></p>		O
	<p>Ohne Bildungsplanbezug – ausschließlich für interessierte, schnelle Schülerinnen und Schüler</p>	<p>7. Streams, Filter-Map-Reduce-Muster</p> <p>Streams, Filter-Map-Reduce-Muster <i>Aufgaben 1-2 zur Inspiration für eigene Aufgaben</i></p> <p>8. generische Methoden</p> <p>Notwendigkeit generischer Methoden <code>aufgabe.a4_generic</code> <code>aufgabe.a6</code> <i>Aufgaben 1-4</i></p>		O