**Kompetenzraster für die Bildungsplaneinheit 5 "Objektorientierte Softwareentwicklung (OOSE)“**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **1 Klassen und Objekte** | **2 Methoden und Kontrollstrukturen** | **3 Vererbung und Assoziation** | **4 Grafische Benutzeroberflächen (GUI)** |
| Kompe-tenzen | * Ich kenne die Begriffe Klasse, Objekt, Attribut, Attributwert und Methode und kann sie voneinander abgrenzen. * Ich kann eigene Objekte anlegen, Attributwerte festlegen, auslesen und Methoden aufrufen. * Ich kann eigene Klassen mit den entsprechenden Attributen und Methoden auf Basis eines UML-Klassendiagramms modellieren und implementieren. Dabei berücksichtige ich die Trennung in Fach- und Präsentationsschicht. * Ich kann eigene Methoden (mit/ohne Parameter, mit/ohne Rückgabewert) und Konstruktoren entwickeln und implementieren. * Ich kenne das Prinzip der Kapselung und Modularisierung und kann dieses anwenden. | * Ich kann Programme unter Nutzung der Kontrollstruktur „Alternative“ entwickeln und implementieren. * Ich kann Programme unter Nutzung der Kontrollstruktur Wiederholung entwickeln und implementieren. * Ich kann Programme mit der Datenstruktur Array/Liste entwickeln und implementieren. * Ich kann Fehler bei der Programmausführung unter Nutzung der Ausnahmebehandlung abfangen. | * Ich kann Klassenhierarchien unter der Verwendung des Konzepts der Vererbung entwickeln und implementieren. * Ich kann abstrakte Klassen entwickeln und implementieren. * Ich kann Beziehungen zwischen Klassen unter der Verwendung des Konzepts der Assoziation entwickeln und implementieren. | * Ich kann Projekte unter Verwendung einer grafischen Benutzeroberfläche entwickeln und implementieren. * Ich kenne den Begriff Zwei-Schichten-Architektur und kann ihn auf Programme anwenden. * Ich kann eine grafische Benutzeroberfläche mit verschiedenen Widgets mit Hilfe eines Generators und von Klassenbibliotheken erstellen und layouten. * Ich kann Ereignisbehandlung in Programmen mit grafischen Benutzeroberflächen praktisch umsetzen. |
| Inhalte | 1.1 Grundbegriffe kennen und unterscheiden   * Klasse und Objekt * Attribut und Attributwert * Methode * UML-Klassendiagramm   1.2 Mit Objekten arbeiten   * Objekt erstellen * Attributwert zuweisen * Methode aufrufen   1.3 Klassen erstellen   * UML-Klassendiagramm modellieren * Konstruktor erstellen * Textausgabe   1.4 Methoden implementieren   * ohne/mit Rückgabewert * ohne/mit Parameter   1.5 Konstruktor implementieren   * Leerkonstruktor * Angabe von Attributwerten erzwingen   1.6 Sichtbarkeit von Attributen   * privates Attribut anlegen * set- und get-Methode implementieren * set- und get-Methode anwenden | 2.1.1 Programme mit einseitiger Verzweigung entwickeln, dokumentieren und implementieren.   * Vergleichsoperatoren   2.1.2 Programme mit zweiseitiger Verzweigung entwickeln, dokumentieren und implementieren.  2.1.3 Programme mit mehrseitiger Verzweigung entwickeln, dokumentieren und implementieren.  2.1.4 Verzweigungen mit verknüpften Bedingungen entwickeln, dokumentieren und implemen­tieren.  2.2 Programme mit wiederholenden Anweisungen entwickeln, dokumentieren und implementieren  2.3 Datenstruktur Array implementieren  2.4 Ausnahmebehandlungen implementieren   * try - except * ValueError * AssertionError * finally | 3.1.1 Klassenhierarchien (Vererbung) entwickeln und modellieren.   * Oberklasse und Unterklasse * Generalisierung und Spezialisierung   3.1.2 Vererbung implementieren.   * Methoden überschreiben * super() * Klassen importieren   3.2 Abstrakte Klassen erklären, modellieren und implementieren.  3.3.1 Beziehungen zwischen Klassen (Assoziationen) entwickeln und modellieren.   * Multiplizitäten * Navigierbarkeit * Rollen   3.3.2 Assoziationen implementieren. | 4.1 Programme auf Basis der Zwei-Schichten-Architektur anwenden:   * Werte ausgeben * Werte einlesen * Ereignisse beim Klick auf eine Schaltfläche verarbeiten   4.2 Programme mit einer GUI entwickeln, die Kontrollkästchen (Checkbox) enthalten. Diese Programme dokumentieren und implementieren.  4.3 Programme mit einer GUI entwickeln, die Kontrollkästchen (RadioButtons) enthalten. Diese dokumentieren und implementieren.  4.4 Programme mit einer GUI entwickeln, die Schieberegler (Slider) enthalten. Diese dokumentieren und implementieren.  4.5 Eigene GUI erstellen. Eigenes Programm mit dieser GUI entwerfen und entwickeln. |