

| | |
|----------------------------------|--|
| Studiengang: | Bachelor Bioinformatik |
| Modulbezeichnung: | Programmierung 2 |
| ggf. Kürzel: | I-B-2 |
| ggf. Untertitel: | - |
| ggf. Lehrveranstaltungen: | Vorlesung: Programmierung 2 Übung: Programmierung 2 |
| Semester: | 2. Semester |
| Angebotsturnus: | jährlich im Sommersemester |
| Modulverantwortliche(r): | Prof. Dr. Sebastian Hack |
| Dozent(in): | Prof. Dr. Sebastian Hack, Prof. Dr. Andreas Zeller |
| Sprache: | Deutsch |
| Zuordnung zum Curriculum: | Wahlpflichtmodulelement der Kategorie „Grundvorlesungen der Informatik“ |
| Lehrform / SWS: | Vorlesung: 2 SWS Übung: 4 SWS Übungsgruppen mit bis zu 20 Studierenden |
| Arbeitsaufwand: | 270 h = 45 h Präsenz- und 225 h Eigenstudium und Bearbeitung der Übungsaufgaben |
| Kreditpunkte: | 9 |
| Voraussetzungen: | Programmierung 1; Mathematik für Informatiker 1 und Mathematikveranstaltungen im Studiensemester oder vergleichbare Kenntnisse aus sonstigen Mathematikveranstaltungen werden empfohlen |
| Lernziele / Kompetenzen: | Die Studierenden lernen die Grundprinzipien der imperativen /objektorientierten Programmierung kennen. Dabei wird primär Java als Programmiersprache verwendet. In dieser Vorlesung lernen sie: <ul style="list-style-type: none"> - mittelgroße objektorientierte Systeme in Java zu implementieren und zu testen - kleinere, wohlstrukturierte Programme in C++ zu schreiben - im Wesentlichen als Umsetzung/Übersetzung der entsprechenden Java-Konzepte - sich in wenigen Tagen eine neue |

| | |
|---|--|
| | imperative/objektorientierte Sprache anzueignen, um sich in ein bestehendes Projekt einzuarbeiten |
| Inhalt: | <ul style="list-style-type: none"> - Objekte und Klassen - Klassendefinitionen - Objektinteraktion - Objektsammlungen - Objekte nutzen und testen - Vererbung - Dynamische Bindung - Fehlerbehandlung - Klassendesign und Modularität - Objekte in C++ - Systemnahe Programmierung <p>sowie spezifische Vorlesungen für die Programmieraufgaben.</p> |
| Studien- Prüfungsleistungen: | <p>Prüfungsleistungen werden in zwei Teilen erbracht, die zu gleichen Teilen in die Endnote eingehen. Um die Gesamtveranstaltung zu bestehen, muss jeder Teil einzeln bestanden werden.</p> <p>Im Praktikumsteil müssen die Studierenden eine Reihe von Programmieraufgaben selbstständig implementieren. Diese Programmieraufgaben ermöglichen das Einüben der Sprachkonzepte und führen außerdem komplexere Algorithmen und Datenstrukturen ein. Automatische Tests prüfen die Qualität der Implementierungen. Die Note des Praktikumsteils wird maßgeblich durch die Testergebnisse bestimmt.</p> <p>Im Vorlesungsteil müssen die Studierenden eine Klausur absolvieren und Übungsaufgaben bearbeiten. Die Aufgaben vertiefen dabei den Stoff der Vorlesung. Die Zulassung zu der Klausur hängt von der erfolgreichen Bearbeitung der Übungsaufgaben ab.</p> <p>Im Praktikumsteil kann eine Nachaufgabe angeboten werden.</p> |
| Literatur: | Bekanntgabe jeweils vor Beginn der Vorlesung auf der vorlesungsseite im Internet |