Modulbeschreibung 39-Inf-PP Prinzipien der Programmierung

Technische Fakultät

Version vom 07.11.2026

Dieses Modulhandbuch gibt den derzeitigen Stand wieder und kann Änderungen unterliegen. Aktuelle Informationen und den jeweils letzten Stand dieses Dokuments finden Sie im Internet über die Seite

https://ekvv.uni-bielefeld.de/sinfo/publ/modul/420520482

Die jeweils aktuellen und gültigen Regelungen im Modulhandbuch sind verbindlich und konkretisieren die im Verkündungsblatt der Universität Bielefeld veröffentlichten Fächerspezifischen Bestimmungen.

39-Inf-PP Prinzipien der Programmierung

Fakultät

Technische Fakultät

Modulverantwortliche*r

Prof. Dr. Markus Nebel

Prof. Dr. Alexander Sczyrba

Turnus (Beginn)

Jedes Wintersemester

Leistungspunkte

10 Leistungspunkte

Kompetenzen

Ziel ist die Vermittlung der grundlegenden Prinzipien verschiedener Programmiersprachen (imperativ, objektorientiert, funktional) und ihr praktischer Einsatz in der Softwareentwicklung. Studierende erlernen die Fähigkeit zur selbständigen Lösung einfacher Algorithmenentwurfs- und Programmieraufgaben.
Insbesondere sollen die Studierenden in der Lage sein, Software aus vorhandenen Bibliotheken wiederzuverwenden und selbst entwickelte Software so zu strukturieren, dass diese einfach erweitert und wiederverwendet werden kann.
Darüber hinaus verstehen die Studierenden am Beispiel von Java und Haskell, wie man neue Programmierkonzepte systematisch erlernt und fundiertes Sprachwissen effizient erwirbt und präzise kommuniziert.

Lehrinhalte

Die Vorlesung "Grundlagen und Prinzipien der Programmierung " führt zunächst in grundlegende Konzepte der Informatik ein (Prinzipien der Programmierung). Sie behandelt Themen wie "Was ist Informatik?", Spezifikation und Algorithmus, Strategien für den Algorithmenentwurf und Maschinenmodelle. Grundlegende Algorithmen auf Listen und Bäumen sowie zur Verarbeitung von Zeichenreihen werden entwickelt.

Parallel zu diesem, mehr theoretisch orientierten, Vorlesungsteil eine Einführung in das Programmieren. Nach einer Einführung in die Kalküle und Terminologie zur systematischen Beschreibung von Programmiersprachen werden die imperative Programmierung und darauf aufbauend schrittweise die objektorientierten Sprachkonstrukte der Programmiersprache Java vorgestellt. Dabei stehen der problembezogene, sinnvolle Einsatz der Sprachkonstrukte und ihr Bezug zu ähnlichen Konzepten in anderen objektorientierten Programmiersprachen im Vordergrund. Lehrinhalte sind u.a. Typen, Werte und Variablen, Objekte und Klassen, Methoden, Kontrollstrukturen, Rekursion, Vererbung, Interfaces, Datenkapselung, Sichtbarkeit und Gültigkeitsbereiche, Generics und Exceptions.

Weiterhin wird die funktionale Programmierung am Beispiel Haskell behandelt. Neben dem Erlernen der jeweils wesentlichen Sprachelemente sind Konzepte wie Rekursion, Funktionen höherer Ordnung, parametrischer Typ-Polymorphismus, Typ-Klassen und die Lazy Evaluation wichtige Lehrinhalte.

In der Übung werden Aufgaben zur Vorlesung behandelt. Das Selbststudium zur Vorlesung wird durch Programmieraufgaben angeleitet, die parallel zum Übungsbetrieb zu bearbeiten sind.

Das „Programmierpraktikum“ führt in die grundlegende Anwendung der jeweiligen Compiler in UNIX-Umgebungen, Programmierumgebungen (IDEs) und Versionskontrollsystemen ein. Anhand einfacher Datenstrukturen mit den zugehörigen Operationen und der Umsetzung kleiner objektorientierter Modellierungsaufgaben wird belastbares Grundlagenwissen vermittelt, das anschließend im Anwendungskontext "Grafische Benutzungsoberflächen" realitätsnah vertieft wird.

Empfohlene Vorkenntnisse

–

Notwendige Voraussetzungen

–

Erläuterung zu den Modulelementen

Modulstruktur: 1 uPr [1]

Veranstaltungen

Titel	Art	Turnus	Workload [5]	LP[2]
Grundlagen und Prinzipien der Programmierung	Übung	WiSe	60 h (30 + 30)	2
Grundlagen und Prinzipien der Programmierung	Vorlesung	WiSe	180 h (120 + 60)	6
Programmierpraktikum	Praktikum	WiSe	60 h (30 + 30)	2

Prüfungen

Zuordnung Prüfende	Art	Gewichtung	Workload	LP[2]

Modulverantwortliche*r prüft oder bestimmt Prüfer*in *Portfolio mit Abschlussprüfung bestehend aus:* *1) Portfolio von Übungen zu Inhalten der Vorlesung* *Übungsaufgaben oder Programmieraufgaben, die veranstaltungsbezogen gestellt werden (Bestehensgrenze 50% der erzielbaren Punkte). Die Kontrolle der Übungsaufgaben umfasst auch direkte Fragen zu den Lösungsansätzen, die von den Studierenden in den Übungen beantwortet werden müssen. Der*die Lehrende kann ein individuelles Erläutern und Vorführen von Aufgaben verlangen sowie einen Teil der Übungsaufgaben durch Präsenzübungen ersetzen. Die Übungsaufgaben im Rahmen des Portfolios werden in der Regel wöchentlich ausgegeben und dienen dem begleitenden Erlernen selbständiger Umsetzungen der in der Vorlesung vorgestellten Lerninhalte.* *2) einer Abschlussprüfung im Rahmen des Programmierprojekts* *Das im Rahmen des Praktikums begonnene Programmierprojekt muss erfolgreich beendet werden und darüber hinaus muss im Projektbericht (ca. 4 Seiten) reflektiert werden, wie methodisch vorgegangen wurde.* *Die Abschlussprüfung zu den Inhalten des Programmierprojekts nimmt Bezug auf die Übungs- oder Programmieraufgaben oder entwickelt sich aus den in den Übungen erlernten Kompetenzen.* *Beide Portfolioelemente werden durch eine*n Prüfer*in geprüft. Es erfolgt eine abschließende Gesamtbewertung.*	Portfolio mit Abschlussprüfung	unbenotet	-	-

Legende

1 Die Modulstruktur beschreibt die zur Erbringung des Moduls notwendigen Prüfungen und Studienleistungen.

2 LP ist die Abkürzung für Leistungspunkte.

3 Die Zahlen in dieser Spalte sind die Fachsemester, in denen der Beginn des Moduls empfohlen wird. Je nach individueller Studienplanung sind gänzlich andere Studienverläufe möglich und sinnvoll.

4 Erläuterungen zur Bindung: "Pflicht" bedeutet: Dieses Modul muss im Laufe des Studiums verpflichtend absolviert werden; "Wahlpflicht" bedeutet: Dieses Modul gehört einer Anzahl von Modulen an, aus denen unter bestimmten Bedingungen ausgewählt werden kann. Genaueres regeln die "Fächerspezifischen Bestimmungen" (siehe Navigation).

5 Workload (Kontaktzeit + Selbststudium)

SoSe Sommersemester

WiSe Wintersemester

SL Studienleistung

Pr Prüfung

bPr Anzahl benotete Modul(teil)prüfungen

uPr Anzahl unbenotete Modul(teil)prüfungen