BOSS-NR. 61800

Modul INF-BSc-213: Funktionale Programmierung[9]					
Englischer Modultitel: Functional Programming					
Studiengänge: Bachelorstudiengang Informatik , Bachelorstudiengang Angewandte Informatik					
Turnus jährlich im Wintersemester		Dauer 1 Semester	Studienabschnitt Ab 5. Semester	Credits 4	Aufwand 120 (45/75)

1 Modulstruktur

Nr.	Element / Lehrveranstaltung	Typ	Credits	SWS
1	Funktionale Programmierung	V	3	2
2	Übung zur Funktionalen Programmierung	Ü	1	1

2 Lehrveranstaltungssprache: deutsch

3 Lehrinhalte

Die Lehrveranstaltung führt ein in Konzepte und Anwendungen funktionaler, musterbasierter und monadischer Programmierung anhand der zur Zeit mächtigsten und am weitesten verbreiteten funktionalen Programmiersprache Haskell. Der Kern eines funktionales Programms ist ein System rekursiver Gleichungen zwischen funktionalen Ausdrücken. Seine Ausführung besteht – wie in der Algebra – in der Auswertung von Ausdrücken durch Anwendung der Gleichungen. Auch die Eingabe- und Ausgabedaten sind funktionale Ausdrücke, wobei hier die Funktionen auf Konstruktoren beschränkt sind, die allein den Aufbau der Daten(muster) beschreiben. Dieses Sprachmodell und der damit einhergehende Programmierstil unterscheiden sich zunächst stark von dem einer objekt-orientierten, imperativen und zustandsorientierten Sprache wie Java. Sie erlauben in der Regel weitaus problemnahere, häufig sehr flexible und generische Lösungen, die leicht an neue Anforderungen, modifizierte Datenstrukturen, etc., anpassbar sind. Darüberhinaus lassen sich diese Charakteristika funktionaler Programme mithilfe von Typklassen, insbesondere den monadischen, auch auf zustandsorientierte Lösungen übertragen, die deterministische, nichtdeterministische oder um differenzierte Ausnahmebehandlungen erweiterte Berechnungen realisieren. Schließlich lassen sich in Haskell auch klassischerweise der logischen oder relationalen Programmierung vorbehaltene Aufgaben lösen wie die Beantwortung prädikatenlogischer Anfragen, speziell die Lösung von Gleichungen. Dies ist eine Konsequenz der meistens lazy evaluation genannten call-by-need-Strategie, der die Ausführung jedes Haskell-Programms folgt. Sie erlaubt u. a. auch das Rechnen mit potentiell unendlichen Datenströmen, Prozessbäumen, etc. Es geht also bei dieser Lehrveranstaltung neben dem Kennenlernen eines weiteren Programmierstils um die Klassifikation aller wichtigen Programm- und Datenstrukturen auf der Basis mathematischer Modelle und deren mehr oder weniger direkter Implementierung in Haskell. Umgekehrt wird Haskell damit auch als kompakte Modellierungs- und Entwurfssprache einsetzbar, in der sich formale Modelle direkt ausführen lassen (rapid prototyping).

4 Kompetenzen

Die Studierenden lernen den Umgang mit Konzepten funktionaler, musterbasierter und monadischer Programmierung und deren Einsatz in verschiedenen Anwendungsbereichen. Sie werden damit u. a. vorbereitet auf Wahlpflicht-Lehrveranstaltungen wie Übersetzerbau und das Fachprojekt Rapid Prototyping. Außerdem lernen sie, wie diese Konzepte nicht nur zur Lösung reiner Implementierungsaufgaben, sondern auch zur präzisen Modellierung verwendet werden können, was sie wiederum befähigt, die Konzepte sowie entsprechende Werkzeuge auch in Arbeitsumgebungen zu nutzen, in denen nichtfunktionale Implementierungssprachen vorherrschen. Das ist vor allem deshalb möglich, weil – wie in der LV gezeigt wird – auch zustandsorientierte und logische Programmierung Spezialfälle der funktionalen sind.

5 Prüfungen

Modulprüfung: Klausur oder mündliche Prüfung BOSS-NR. 61891
Studienleistung: BOSS-NR. 61841

[9] Bis zum Sommersemester war dieses Modul ein Pflichtmodul und wurde unter der Nummer INF-BSc-114 geführt.

	• Aktive Mitarbeit in der Übung und Erreichen der Mindestpunktzahl der Übungsaufgaben oder erfolgreiche Teilnahme an einem midterm-Test nach Ankündigung des Veranstalters Die Studienleistung ist Voraussetzung für die Teilnahme an der Klausur.		
6	**Prüfungsformen und -leistungen** ☒ Modulprüfung ☐ Teilleistungen		
7	**Teilnahmevoraussetzungen** *Erfolgreich abgeschlossen:* –keine– *Vorausgesetzte Kenntnisse:* Modul „ Datenstrukturen Algorithmen und Programmierung 1 (DAP 1)“, „ Datenstrukturen Algorithmen und Programmierung 2 (DAP 2)“, Modul „Mathematik für Informatiker 1“ oder Modul „Höhere Mathematik 1“, Modul „Mathematik für Informatiker 2“ oder Modul „Höhere Mathematik 2“		
8	**Modultyp und Verwendbarkeit des Moduls** Wahlpflicht-Modul im Bachelor-Studiengang Informatik und Angewandte Informatik[10] *Katalog:* Konzepte für Software		
9	**Modulbeauftragte/r** Prof. Dr. J. Rehof	**Zuständiger Fakultät** Informatik	Beschluss Fakultätsrat 22.05.2019

[10] Bis zum Sommersemester 2019 Pflichtmodul im Bachelorstudiengang Informatik. Dieses Modul kann nicht zusammen mit dem außerkraftgesetzten Pflichtmodul „Funktionale Programmierung“ gewählt werden.