

Aktuelle Trends in der Programmierung

Allgemeine Angaben

Kürzel	TPRG
Modulverantwortliche	Prof. Dr. Nüßer
Dozenten	Prof. Dr. Nüßer, Prof. Dr. Stehr, Prof. Dr. Ströder
Lehrsprache	Deutsch
Semester	4
ECTS-Punkte	5
Kontaktstunden	40
Selbststudium	85
Dauer	1 Semester
Art	Pflicht
Häufigkeit	Jedes Studienjahr
Gewichtung	5/180
Prüfungsleistung	KRS90

Stichwörter

- Programmierparadigmen
 - Funktionale Programmierung, z.B. Scala
 - Logische Programmierung
- Trends, wie VM-Sprachen, Generierung etc.
- Quanten-Computing

Zugangsvoraussetzungen

- Programmierung I
- Programmierung II
- Grundlagen der Informatik
- Software Engineering and Continuous Integration
- Software Modeling and Architecture

Verwendbarkeit

- Secure Software
- Projekt-Module

Qualifikations- und Kompetenzziele

Die Studierenden verstehen, dass Trends in der Programmierung stets auch Antworten auf aktuelle Herausforderungen geben. Sie können diese Herausforderungen und die jeweiligen Lösungsansätze einordnen und bewerten. Konkret kennen sie über die klassischen Paradigmen der Programmierung (prozedural und objekt-orientiert) hinaus die wichtigsten weiteren Ansätze. Sie können unterschiedliche Paradigmen und Programmiersprachen einordnen und bewerten. Sie können kleinere Projekte in unterschiedlichen Programmiersprachen bewältigen. Sie sind in der Lage, auch neue Sprachkonstrukte schnell einordnen zu können.

Lehr- und Lernmethoden

Unterschiedliche Lehr-/Lernumgebungen: Präsenzveranstaltungen, Eigenstudium; Wechselnde Lehr-/Lernmethoden: Individuelles und kooperatives Lernen, problemorientiertes und integratives Lernen, forschendes Lernen, synchrones und asynchrones Lernen; Fallstudienarbeit, Expertenvorträge.

Besonderheiten

Übungsaufgaben, Literaturstudium, Referate, Gruppenarbeiten, Studienarbeiten, Beispielausarbeitungen, Projektarbeiten, Literaturquellensuche und -nutzung, Internetrecherche, Transfer und Querverbindungen zu Praxisunternehmen und anderen Modulen

Inhalte

- Überblick Programmierparadigmen
 - Imperative Programmierung
 - Deklarative Programmierung, insb. funktionale und logische Paradigmen
- Elemente funktionaler Programmierung
 - Grundstrukturen
 - Stärken und Schwächen
 - Beispiele
- Elemente logische Programmierung
 - Grundstrukturen
 - Stärken und Schwächen
 - Beispiele
- Weitere Programmier-Trends, z.B.
 - Notebooks
 - VM-Sprachen
- Fallstudien, z.B.
 - Quanten-Computing
 - Sprach-Performance

Grundlegende Literaturhinweise

GUMM, H.P. und M. SOMMER, 2013. *Einführung in die Informatik*. 10. Auflage. München: Oldenbourg.

ZELLE, J. 2016. *Python Programming*. Wilsonville: Franklin, Beedle & Associates.

Ergänzende Literaturempfehlungen

HUNT, A. and THOMAS, D. 1999. *The pragmatic programmer*. Boston: Addison-Wesley

Weitere Spezialliteratur wird in der Vorlesung bekannt gegeben.