# 65010 Moderne Methoden der Software-Entwicklung

Modulverantwortliche/r: Dr. Sebastian Küpper

| Dauer des Moduls | ECTS | Workload | Häufigkeit |
|---|---|---|---|
| ein Semester | 10 | 300 Stunden | in jedem Semester |

Lehrveranstaltung(en): Moderne Methoden der Software-Entwicklung

Detaillierter Zeitaufwand:
Bearbeitung des Lehrveranstaltungstextes: 150 Stunden
Bearbeitung der Übungs- und Einsendeaufgaben: 75 Stunden
Nachbearbeitung und Klausur- bzw. Prüfungsvorbereitung: 75 Stunden

Qualifikationsziele:
Das Modul soll die Fähigkeit vermitteln, formal genau zu definieren, was eine Software erfüllen muss, um als korrekt zu gelten. Hierfür soll insbesondere der Zusammenhang zwischen Vor- und Nachbedingungen verstanden werden und für einfache Methoden die schwächste Vorbedingung von Hand errechnet werden können. Für komplexe Softwaresysteme ist eine exakte Berechnung von schwächsten Vorbedingungen hingegen kein gangbares Vorgehen, um die Korrektheit einer Software nachzuweisen. Für solche Fälle vermittelt das Modul Modellierungs- und Verifikationstechniken, die den Nachweis der Korrektheit mithilfe automatisierter Verfahren erlauben.
Im Kontext der Videospielentwicklung mit einer Spiele-Engine soll das Modul die algorithmischen Grundlagen essenzieller Komponenten einer Spiele-Engine vermitteln – insbesondere die Kollisionserkennung und Rotation. Darauf aufbauend sollen auch praktische Kenntnisse im Umgang mit Spiele-Engines – am Beispiel von Unity – erworben werden, so dass die selbstständige Entwicklung eines (einfachen) Videospiels ermöglicht wird. Schließlich sollen ein reflektierter Umgang mit den Mitteln der Metaprogrammierung erlernt und grundlegende Refactoring-Techniken verstanden werden.

Inhalte:
Trotz eines zunehmenden Bewusstseins für die Notwendigkeit von vorbereitenden Tätigkeiten ist die Programmierung immer noch die Kernaktivität der Softwareentwicklung. Zwar hat die Einführung neuer Programmiersprachen wie Java oder C# durchaus Produktivitätssteigerungen bewirkt, jedoch ist die Fortentwicklung dieser Sprachen viel zu schwerfällig, um mit den ständig wachsenden Ansprüchen an Funktionalität und Umfang schritthalten zu können. Stattdessen haben sich im Kontext der objektorientierten Programmierung eine ganze Reihe von Techniken und Methoden entwickelt, mit deren Hilfe sich - auf der Basis existierender Programmiersprachen - die Softwareentwicklung effizienter gestalten lässt. In dieser Vorlesung konzentrieren wir uns einerseits auf Methoden, die dazu dienen, die Korrektheit von Software zu definieren und formal nachzuweisen, andererseits lernen wir die Spieleengine Unity kennen, als Beispiel einer integrierten Entwicklungsumgebung für einen speziellen Anwendungsbereich. Refactoring und Metaprogrammierung sind schließlich Techniken, die die Wartbarkeit und Weiterentwicklung von Code erleichtern.
Bitte beachten Sie: die ersten drei Lektionen, die sich um die Spezifikation und die Verifikation der Korrektheit von Software drehen, erfordern eine formal-analytische Betrachtungsweise von Software und basieren auf Techniken der Theoretischen Informatik.

Die Themen der Lehrveranstaltung sind:
- Design by contract
- Modellierung zustandsbasierter Systeme
- Verifikation via CTL Model Checking
- Videospielentwicklung mit einer Spieleengine (Unity)
- Refactoring
- Metaprogrammierung

Ergänzende Literatur:

E.Gamma et al. Design Patterns (Addison-Wesley 1995).
M. Fowler: Refactoring: Improving the Design of Existing Code (Addison-Wesley 1999),

Refactoring (deutsche Version) (mitp Professional 31.10.2019)
K. Beck, C. Andres: Extreme Programming Explained. Embrace Change 2. Ausgabe (Addison Wesley 2004).

| | |
|---|---|
| Inhaltliche Voraussetzung | Es werden Programmierkenntnisse in C# oder Java vorausgesetzt, wie sie z. B. durch das Modul 63611 "Einführung in die objektorientierte Programmierung" vermittelt werden. Zusätzlich ist ein durch Praxis erworbenes Gefühl für die objektorientierte Programmierung nützlich, um die Bedeutung der in der Lehrveranstaltung geschilderten Probleme und Lösungen abschätzen zu können.<br><br>Weiterhin wird vorausgesetzt, dass Sie grundlegende Kenntnisse der Theoretischen Informatik, beispielsweise durch das Modul 63912 „Grundlagen der Theoretischen Informatik“ mitbringen. |
| Lehr- und Betreuungsformen | Lehrveranstaltungsmaterial<br>Internetgestütztes Diskussionsforum<br>Einsendeaufgaben mit Korrektur und/oder Musterlösung |
| Anmerkung | Das Modul 65010 "Moderne Methoden der Software-Entwicklung" ersetzt das Modul 63613 "Moderne Programmiertechniken und -methoden". |
| Formale Voraussetzung | keine |

| | |
|---|---|
| Verwendung des Moduls | M.Sc. Data Science<br>M.Sc. Informatik<br>M.Sc. Praktische Informatik<br>M.Sc. Wirtschaftsinformatik |

## Prüfungsformen

| | Art der Prüfungsleistung | Voraussetzung |
|---|---|---|
| Prüfung | benotete zweistündige Prüfungsklausur | Eine Zulassung zur Klausur erfolgt, wenn in 5 von 7 Einsendeaufgaben mindestens ein Punkt erreicht wurde. |
| Stellenwert der Note 1/12 | | |