

Modulnr. PTI822	Modulname Grundlagen der Softwareentwicklung	Dozent(en) Prof. Dr.S.Schwarz, Fak. PTI Prof. Dr. G. Beier, Fak. PTI
Studiengäng(e): Informatik (B. Sc.)	Semester: 2. Semester (SS)	
Studienrichtung(-en)/-schwerpunkt(-e): alle	ECTS-Punkte: 8 Arbeitsaufwand in h: 240	
	Lehr- und Lernformen in h:	
	Vorlesung	60 (4 SWS)
	Praktikum	30 (2 SWS)
	Vor-/ Nachbereitung	90
	Selbststudium	60
Lernziele		
<p>Die Studierenden beherrschen die grundlegenden Methoden der Softwareentwicklung. Sie kennen klassische Algorithmen und abstrakte Datentypen und können diese in der Softwareentwicklung einsetzen.</p> <p>Software Engineering: Die Studierenden können Softwareentwicklungsprojekte mit objektorientierten Methoden planen und modellieren. Sie kennen informelle und formalisierte Verfahren der Anforderungsspezifikation und können diese anwenden. Sie kennen die zentrale Bedeutung des Konfigurationsmanagements und können übliche Versionsmanagementwerkzeuge im Rahmen eines geplanten Konfigurationsmanagementprozesses anwenden.</p> <p>Algorithmen und Datenstrukturen: Die Studierenden können algorithmisch denken, mit Abstraktionskonzepten arbeiten, praktische Probleme algorithmisch lösen und Algorithmen in Programme zu übersetzen. Sie beherrschen den Entwurf von Algorithmen und Datenstrukturen sowie die Analyse von Struktur und Verhalten von Algorithmen. Sie beherrschen Prinzipien für den systematischen Algorithmenentwurf und Konzepte zur Modellierung komplexer Datenstrukturen und kennen wichtige Standardalgorithmen. Sie können die Laufzeit von Verfahren abschätzen. Sie wissen, dass die Effizienz eines Algorithmus von der Wahl der Datenstrukturen abhängt und finden zu gegebenen Algorithmen passende Datenstrukturen.</p>		
Lehrinhalte		
<p>Software Engineering (Vorlesung: 30 h, Praktikum: 15 h, Vor/Nachbereitung: 45 h, Selbststudium: 30h)</p> <ul style="list-style-type: none"> • Einführung in das Fach Software Engineering • Kreativitätstechniken zur Anforderungserhebung • Anforderungsspezifikation mit Use Case Modellen • Klassenmodellierung mit der UML • Anwendung von UML-CASE-Werkzeugen • Konzept und Anwendung von Entwurfsmustern • Umsetzung von UML-Modellen in die Implementierung • Konfigurationsmanagement und Versionsverwaltung • Objektorientiertes Testen, Unit-Tests • Testgetriebene Entwicklung • Einsatz einer modernen integrierten Entwicklungsumgebung (Eclipse) <p>Algorithmen und Datenstrukturen (Vorlesung: 30 h, Praktikum: 15 h, Vor/Nachbereitung: 45 h, Selbststudium: 30 h)</p> <ul style="list-style-type: none"> • Algorithmen (iterativ, rekursiv) und Laufzeitabschätzungen • Sortierverfahren (Bubble-, Selection-, Insertion-, Merge-, Quick-, Heapsort) • Abstrakte Datentypen (Folge, Menge, Abbildung, Priority-Queue) und Implementierungen: <ul style="list-style-type: none"> lineare Datenstrukturen: Listen, Stacks, Queues hierarchische Datenstrukturen: Bäume, binäre Suchbäume, 2-3-Bäume, AVL-Bäume • Graphalgorithmen (Zusammenhangskomponenten, Minimalgerüste, kürzeste Wege) • Pattern Matching Algorithmen (Boyer-Moore-, Knuth-Morris-Pratt-Verfahren) <p>Literatur</p> <p>Software Engineering</p> <ul style="list-style-type: none"> • Fowler, Martin: UML konzentriert, Addison-Wesley, 2003 • Oestereich, Bernd: Objektorientierte Softwareentwicklung, Analyse und Design mit UML 2.1; Oldenbourg, 2006 • Störrle, Harald: UML 2 für Studenten, Pearson Studium, 2005 (PDF-Ausgabe) 		

Algorithmen und Datenstrukturen

- Cormen, Leiserson, Rivest, Stein: Algorithmen - Eine Einführung, Oldenbourg, 2007
- Robert Sedgewick: Algorithmen in Java Grundlagen, Datenstrukturen, Sortieren, Suchen. Teil 1-4, Pearson Studium, 2003
- Ralf Hartmut Güting, Stefan Diekert: Datenstrukturen und Algorithmen, Teubner, 2003

Voraussetzungen/Vorkenntnisse

Inhalte entsprechend Modul

PT1819 - Grundlagen der Programmierung (1. Semester)

PT1892 - Logik

Leistungsnachweise

Art: schriftliche Prüfungsleistung

Zeitdauer: 120 min

Vorleistungen: Testat

Erarbeitet am: 16.01.2009

durch: Prof. S. Schwarz, Prof. Dr. G. Beier

PLS 04. Juni 2010