

Lehrveranstaltungshandbuch PI2

Praktische Informatik 2

Version: 2 | Letzte Änderung: 10.09.2019 15:52 | Entwurf: 0 | Status: vom verantwortlichen Dozent freigegeben

– Allgemeine Informationen

Langname Praktische Informatik 2

**Anerkennende
LModule** [PI2 BaET](#)

Verantwortlich Prof. Dr. Dieter
Rosenthal
Professor Fakultät IME

Gültig ab Sommersemester 2021

Niveau Bachelor

Semester im Jahr Sommersemester

Dauer Semester

**Stunden im
Selbststudium** 60

ECTS 5

Dozenten Prof. Dr. Dieter
Rosenthal
Professor Fakultät IME
Derichs

Voraussetzungen Grundkenntnisse der
Sprache C

Unterrichtssprache deutsch

**separate
Abschlussprüfung** Ja

Literatur

Elektronische Verweise auf ebooks und Online
Tutorials

Abschlussprüfung

Details

Schriftliche Klausur:
Die Studierenden sollen nachweisen, dass sie 1.) mit grundlegenden Begrifflichkeiten sicher umgehen, 2.) programmiersprachliche und abstraktere Konstrukte zur Lösung von Anwendungsproblemen einsetzen und 3.) programmiersprachliche Lösungsvorschläge auf Korrektheit prüfen können. Typische Aufgabenformen zu 1.) sind Multiple-Choice-Fragen, Lückentexte, Bewertung von Aussagen hinsichtlich ihrer Korrektheit, zu 2.) Lösung kleinerer umgangssprachlich formulierter Probleme durch Struktogramme und Programmstücke und zu 3.) das Finden von Fehlern in vorgegebenen Programmstücken.

Mindeststandard

Mindestens 50% der möglichen Gesamtpunktzahl.

Prüfungstyp

Klausur

– Vorlesung / Übungen

Lernziele

Zieltyp	Beschreibung
Kenntnisse	Grundlagen der objektorientierten Programmierung
Kenntnisse	Aufbau von Klassen, Erstellen von Objekten Konstruktoren, Überladen von Methoden, Initialisierungsliste Referenzen vs. Pointer Destruktoren
Kenntnisse	Vererbung
Kenntnisse	Polymorphie abstrakte Methoden und Klassen virtuelle Methoden
Kenntnisse	Zugriffsmechanismen private, protected, public friend
Kenntnisse	Schlüsselwort static Benutzung in Funktionen/Methoden vs. Klassen
Kenntnisse	Templates: Methoden Klassen
Fertigkeiten	Erstellung von Klassen und Objekten
Fertigkeiten	Programmierung von abgeleiteten Klassen und Erstellung der zugehörigen Objekte
Fertigkeiten	Anwenden von Polymorphie bei geerbten Klassen Programmierung abstrakter Klassen Programmierung virtueller Methoden
Fertigkeiten	Programmieren von Attributen und Methoden in verschiedenen Schutzbereichen
Fertigkeiten	Programmierung von templates

Besondere Voraussetzungen

Grundkenntnisse in C

Begleitmaterial

elektronische Vortragsfolien zur Vorlesung, frei verfügbare Softwareentwicklungsumgebungen aus dem Internet, elektronische Sammlung von Beispielprogrammen

Separate Prüfung

Nein

Aufwand Präsenzlehre

Typ**Präsenzzeit (h/Wo.)**

Vorlesung

2

Übungen (ganzer Kurs)

1

Übungen (geteilter
Kurs)

1

Tutorium (freiwillig)

0

– Praktikum

Lernziele

Zieltyp	Beschreibung
Kenntnisse	Erstellung von Klassen und Objekten
Kenntnisse	Programmierung von abgeleiteten Klassen und Erstellung der zugehörigen Objekte
Kenntnisse	Anwenden von Polymorphie bei geerbten Klassen Programmierung abstrakter Klassen Programmierung virtueller Methoden
Kenntnisse	Programmieren von Attributen und Methoden in verschiedenen Schutzbereichen
Fertigkeiten	Anwendung der unter "Kenntnisse" und "Fertigkeiten" genannten Aspekte auf praxisbezogene Szenarien durch selbstständige Arbeit in kleinem Team.

Aufwand Präsenzlehre

Typ	Präsenzzeit (h/Wo.)
Praktikum	1
Tutorium (freiwillig)	0

Besondere Voraussetzungen

keine

Begleitmaterial elektronische Sammlung von Beispielprogrammen

Separate Prüfung Ja

Separate Prüfung

Prüfungstyp praxisnahes Szenario bearbeiten (z.B. im Praktikum)

Details

Die Studierenden schließen sich zu Kleingruppen zusammen. Jede Kleingruppe absolviert mehrere "Praktikumsrunden" mit zugewiesenen Laborterminen. In jeder Runde werden Programmieraufgaben algorithmischer und objektorientierter Art gelöst - erstens durch eine abstraktere Darstellung (z.B. Beschreibung eines Algorithmus durch ein Struktogramm), zweitens durch eine programmtechnische Umsetzung (z.B. C++-Programm). Zur Vorbereitung eines Labortermins muss ein "Vorbereitungsblatt" praktisch gelöst werden. Die dabei erworbenen Kenntnisse werden zu Beginn des Termins geprüft (kurzer schriftlicher Eingangstest, persönliches Gespräch mit dem Betreuer). Wird diese Prüfung nicht bestanden, so muss ein Folgetermin wahrgenommen werden; im Wiederholungsfall führt dies zum Nichtbestehen des Praktikums. Im Erfolgsfall wird ein "Laborarbeitsblatt" mit weiteren Aufgaben unter Aufsicht (und ggf. mit Hilfestellung) bearbeitet.

Mindeststandard

Erfolgreiche Teilnahme an allen Laborterminen, d.h. insbesondere selbstständige (ggf. mit Hilfestellung) Lösung der Programmieraufgaben.

