

Lehrveranstaltung

PI1 - Praktische Informatik 1

Version: 2 | Letzte Änderung: 01.08.2019 16:59 | Entwurf: 0 | Status: vom verantwortlichen Dozent freigegeben

^ Allgemeine Informationen

Langname	Praktische Informatik 1
Anerkennende LModule	<u>PI1 BaTIN</u>
Verantwortlich	Prof. Dr. Cartsten Vogt Professor Fakultät IME
Niveau	Bachelor
Semester im Jahr	Wintersemester
Dauer	Semester
Stunden im Selbststudium	60
ECTS	5
Dozenten	Prof. Dr. Cartsten Vogt Professor Fakultät IME
Voraussetzungen	keine
Unterrichtssprache	deutsch
separate Abschlussprüfung	Ja

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

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.

^ Vorlesung / Übungen

Lernziele

Kenntnisse

Algorithmen
Eigenschaften
Darstellungsmöglichkeiten

Digitalrechner
Bit/Byte-Begriff
Grobaufbau der Soft- und Hardware-Architektur

Grundbegriffe der Programmierung
höhere Programmiersprachen vs. Maschinensprachen
Übersetzung vs. Interpretation
prozedurale vs. objektorientierte Sprachen: C vs. Java

Variablen-Begriff

einfache Datentypen in Java (und C)
Zahlen
Wertebereiche
Konstantendarstellung
Operationen
Zeichen
Codierungen: ASCII, Unicode
Operationen
Zeichenketten
boolesche Werte
Konstantendarstellung
Operationen

Kontrollstrukturen in Java (und C)
abstrakte Darstellungsformen
Struktogramme
Programmablaufpläne

Blöcke
bedingte Anweisungen
if
if-else
Mehrfachverzweigungen: switch-case
Schleifen
kopfgesteuerte Schleifen
for
while
fußgesteuerte Schleifen: do-while

statische Methoden in Java
Definition von Methoden
Kopf mit Parametern und Rückgabotyp
Körper mit Wertrückgabe
Aufruf von Methoden
Parameterübergabe: Wertaufruf vs. Referenzaufruf
Überladung
Speicherklassen

Arrays in Java
Speicherorganisation: Referenz-Begriff
Indizierung und Schleifen
mehrdimensionale Arrays

Objekte und Klassen in Java
Motivation und Grundbegriffe der objektorientierten Programmierung
Kapselung
Objekte mit Attributen und Methoden
Klassen
Konstruktoren
Zugriffsschutz
klassenbezogene Attribute und Methoden

Fertigkeiten

Erstellung von Algorithmen zur Lösung vorgegebener Probleme
Beschreibung in natürlicher Sprache
Beschreibung in grafischer Form (Struktogramme und/oder Programmablaufpläne)

Programmierung elementarer Operationen in einer höheren Sprache
auf Zahlen
auf Zeichen und Zeichenketten
auf booleschen Werten

Programmierung mit Kontrollstrukturen

Programmierung mit Methoden

Programmierung mit strukturierten Datentypen, insbes. Arrays

Aufwand Präsenzlehre

Typ	Präsenzzeit (h/Wo.)
Vorlesung	2
Übungen (ganzer Kurs)	1
Übungen (geteilter Kurs)	1
Tutorium (freiwillig)	0

Separate Prüfung

keine

^ Praktikum

Lernziele

Kenntnisse

Programmierung elementarer Operationen auf einfachen Datentypen

Programmierung mit Kontrollstrukturen (mit vorheriger Erstellung entsprechender Struktogramme und/oder Programmablaufpläne)

Programmierung mit Methoden

Programmierung mit strukturierten Datentypen, insbes. Arrays

Fertigkeiten

Umgang mit einer Softwareentwicklungsumgebung

Fehlersuche und -beseitigung in Programmen

Erstellung von Algorithmen und Umsetzung in Programme

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

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. Java-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.