

<b>Programmiersprachen-Praktikum</b> <i>Programming Languages Practical</i>								Modulnummer:	
Bachelor				Schwerpunkt					
Pflicht <input type="checkbox"/>				Computational Finance <input type="checkbox"/>					
Winf-Schwerpunkt-Pflicht <input type="checkbox"/>				E-Business <input type="checkbox"/>					
Winf-Schwerpunkt-Wahlpflicht <input type="checkbox"/>				IT-Management <input type="checkbox"/>					
Winf-Wahl <input type="checkbox"/>				Logistik <input type="checkbox"/>					
Anzahl der SWS	V	UE	K	S	Prak.	Proj.	$\Sigma$	Kreditpunkte: 4	Turnus alle 4 Semester
	0	0	0	0	2	0	2		
Formale Voraussetzungen: ME-705.01 Programmiersprachen									
Inhaltliche Voraussetzungen: -									
Vorgesehenes Semester: ab 5. Semester									
Sprache: Deutsch									
<p>Ziele: Wissen über Konzepte moderner Programmiersprachen auf die Untersuchung eine konkrete Programmiersprache übertragen können</p> <p>Wissen über Programmierstile (Paradigmen), wie imperatives, objektorientiertes, funktionales und logisches Programmieren auf die Untersuchung eine konkrete Programmiersprache übertragen können</p> <p>Beschreibungen von Programmiersprachen in kleinen Teams verstehen können,</p> <p>Die Qualität einer Programmiersprache erfassen können</p> <p>Einen wissenschaftlichen Text über eigene Untersuchungsergebnisse konzipieren und verfassen können</p> <p>Über Erfahrung in der projektbezogenen Problemlösung in kleinen Teams verfügen.</p>									

Inhalte: Untersuchung einer konkreten Programmiersprache in Hinsicht auf:

Konzepte wie

- Werte (Datenstrukturen und Ausdrücke).
- Speicher (Variablen und Befehle)
- Bindung (Vereinbarungen und Gültigkeitsbereiche).
- Abstraktion (Funktionen, Prozeduren und Parameterübergabe).
- Kapselung (Moduln, abstrakte Datentypen, Klassen, generische Pakete).
- Typsysteme (Überladen, Anpassungen, Polymorphie, Untertypen und Vererbung).
- Ablaufsteuerung (Sprünge, Ausweg, Ausnahmen).
- Nebenläufigkeit und Verteiltheit

die Unterstützung von Paradigmen (Programmierstilen) wie

- Imperatives Programmieren.
- Objekt-orientiertes Programmieren.
- Nebenläufiges Programmieren.
- Funktionales Programmieren.
- Logisches Programmieren.

Beurteilung von Sprachen nach Prinzipien des Sprachentwurfs wie

- Syntax.
- Semantik.
- Pragmatik.

In der Übung Anwendung der in der Vorlesung erworbenen Kenntnisse und Fähigkeiten bei der Untersuchung spezifischer Konzepte und Eigenschaften von spezifischer Programmiersprachen (z. B. Ada, Eiffel, Java, Haskell, Prolog) Der Stoff des Kurses Programmiersprachen (ME 705-01) wird vertieft. Gruppen von bis zu zwei Studierenden wählen eine Beispielsprache aus, die sie anhand der im Kurs behandelten Konzepte und Paradigmen untersuchen.

Bis zum Semesterende erstellen sie einen Bericht von ca. 20 Seiten über die Beispielsprache.

Unterlagen (Skripte, Literatur, Programme usw.): David A. Watt (2004). Programming Languages Design Concepts, Chichester: Wiley and Sons.

Robert W. Sebesta (2002). Concepts of Programming Languages 5/e, Reading, MA: Addison-Wesley. 670 Seiten.

Form der Prüfung:

Semesterarbeit: Bericht über die Analyse einer Programmiersprache

Arbeitsaufwand	Präsenz	28 h
	Übungsbetrieb/Prüfungsvorbereitung	92 h
	Summe	120 h

Lehrende:  
Berthold Hoffmann

Verantwortlich:  
Berthold Hoffmann