

Vorlesung

Objektorientierte Programmierung

Studiengang MBI

Prof. Dr. Uwe Petermann
FB IMN, HTWK Leipzig, 04251 Leipzig (Germany)
uwe@imn.htwk-leipzig.de

Zusammenfassung

Die Vorlesung vermittelt theoretische und praktische Kenntnisse in der objektorientierten Programmierung. An Fallbeispielen wird vermittelt, wie aus einer Problemstellung die für die rechen-technische Lösung wesentlichen Algorithmen und Datenstrukturen abgeleitet werden können.

Eine verbreitete und erfolgreiche Herangehensweise ist dabei, Daten als Objekte zu betrachten, die durch Operationen, auch Methoden genannt, bearbeitet oder untersucht werden können. Dabei kann von den Details der Realisierung dieser Operationen abstrahiert werden.

Als Programmiersprache wird Java benutzt, eine universelle und plattformunabhängige Programmiersprache, die Objektorientiertheit, Parallelarbeit und Ausnahmebehandlung in sich vereint. Für die Praxis ist der große Umfang der zu der Sprache gehörenden Klassenbibliotheken wichtig, die auszugsweise behandelt werden.

Dieses Dokument als .pdf-Datei

Ablauf, Teilnahme, Leistungsnachweis

Ablauf

Vorlesung (1SWS) und Übung/Rechnerübung (2SWS)

Teilnahme

Teilnahme an den zur Vorlesung gehörenden Lehrveranstaltungen ist Pflicht

Leistungsnachweis

Prüfungsrelevante Studienleistung Projekt

Materialien zur Vorlesung

Einführung und Überblick

Einführung:

Grundlegende Konstrukte von Programmiersprachen

Grundlegende Konstrukte von Programmiersprachen:

Grundbegriffe zur Objektorientierung

Grundbegriffe zur Objektorientierung:

Fallstudie 1 - Sortieren von Objekten

Fallstudie 1 - Sortieren von Objekten:

Fallstudie 2 - Dateien und Datenstrukturen

Fallstudie 2 - Dateien und Datenstrukturen:

Fallstudie 3 - Objektorientierung und Rekursion

Fallstudie 3 - Objektorientierung und Rekursion

Übungen zur Vorlesung Objektorientierte Programmierung:

Sortieren einfacher Objekte

Eine Folge von Objekten von Standardklassen wird nach deren eingebauter Ordnung sortiert.

SortSimple.java

Sortieren komplexer Objekte

ComplexSort.java

Eine Folge von Objekten einer selbst definierten Klasse Name

Name.java

wird nach der durch die Implementierung festgelegten Ordnung sortiert.

Arbeit mit der Schnittstelle java.util.Map

Kommentierte und überarbeitete Version des Beispiels FrequencyTable aus dem Tutorium zum Collection Framework (

<http://java.sun.com/docs/books/tutorial/collections/index.html>

)

1. Beispielaufruf:

```
java FrequencyTable1 to be or not to be happy
```

Programm liest von der Standardeingabe und
FrequencyTable1.java

2. Gegenüberstellung der verschiedenen Sichten (views).
FrequencyTable2.java

Weitere interessante Quellen:

1. Einführung in das Java Collections Framework:
2. Einführungen in die Arbeit mit Java:
3. Java Technologie von der Quelle:
4. BlueJ — Eine einfache Java-Entwicklungsumgebung:
5. Emacs — Ein erweiterbarer Editor:
6. JDE — Eine Java-Entwicklungsumgebung als Emacs-Erweiterung:
7. XEmacs — Ein erweiterbarer Editor: