

# Programmiermethodik

## Vorlesung und Praktikum

SS 2001

Prof. Dr. W. Effelsberg, G. Kühne, Ch. Kuhmünch

Universität Mannheim

	Programmiermethodik © Prof. Dr. W. Effelsberg	1. Einführung	1-1
---	--	---------------	-----

### Inhalt

1. Einführung, Vorstellung der Programmieraufgabe
2. Der Software-Entwicklungszyklus
3. Konzepte der objektorientierten Programmierung
4. Verteilte Anwendungen
5. Objektorientierte Analyse
6. Objektorientiertes Design
7. Objektorientiertes Programmieren
8. Verwaltung gemeinsamer Daten mit CVS

	Programmiermethodik © Prof. Dr. W. Effelsberg	1. Einführung	1-2
---	--	---------------	-----

## Literatur

- G. Bannert, M. Weitzel: **Objektorientierter Softwareentwurf mit UML**. Addison-Wesley-Longman, München, 1999

Dies ist das Lehrbuch zur Vorlesung. Es ist auch in der Lehrbuchsammlung der Universitätsbibliothek Mannheim verfügbar.

- M. Schader, L. Schmidt-Thieme. Java: Eine Einführung. 3. Auflage, Springer-Verlag, 2000

*Hinweis:* Die 3. Auflage unterscheidet sich von der 2. Auflage durch die Aufnahme der Swing-Bibliothek.

- G. Booch: Objektorientierte Analyse und Design. Addison-Wesley, Bonn, Paris, Reading, 1994

	Programmiermethodik © Prof. Dr. W. Efratsberg	1. Einführung	1-3
---	--	---------------	-----

## 1. Einführung, Vorstellung der Programmieraufgabe

- **Lernziel:** Softwareentwicklung im Großen

Erlernen von Methoden und Techniken zu

- Entwurf
  - Spezifikation
  - Implementierung
  - Test
  - Wartung
  - Dokumentation
- von großen, komplexen Software-Systemen.

- **Methode**

Der Schwerpunkt liegt auf dem **Praktikum**, begleitet durch eine 2-stündige Vorlesung, in der die wichtigsten Konzepte vermittelt werden.

	Programmiermethodik © Prof. Dr. W. Efratsberg	1. Einführung	1-4
---	--	---------------	-----

## Die Programmieraufgabe


- Eine Internet-fähige Version des Spiels **CLUEDO**

Für das Praktikum im Sommersemester 2001 haben wir uns die Entwicklung eines verteilten Spiels als Aufgabe ausgedacht. Es ist das bekannte Gesellschaftsspiel "Cluedo". Ziel des Praktikums ist es, eine netzwerkfähige Computerversion dieses Spiels in Java zu implementieren.

	Programmiermethodik © Prof. Dr. W. Efratsberg	1. Einführung	1-5
---	--	---------------	-----

## Die drei Hauptkomponenten des Spiels **CLUEDO**

- **CluedoServer**: Ein CluedoServer ist der zentrale Anlaufpunkt für Cluedo-Spieler. Der Server kann mehrere Spiele (auch als *Sessions* bezeichnet) parallel verwalten. Innerhalb eines Spiels übernimmt der Server die Spielvorbereitung und koordiniert die Spielzüge.
- **CluedoClient**: Ein CluedoClient ermöglicht einem einzelnen Spieler die Teilnahme an einem Cluedo-Spiel. Er stellt die Funktionalität zur Verfügung, die der Spieler vom Brettspiel gewohnt ist. Zusätzlich bietet der Client Funktionen für die Anmeldung eines Spielers bei einem Server sowie für die Auswahl oder Erzeugung einer Session. Der Phantasie bezüglich der graphischen Gestaltung der Benutzeroberfläche sind keine Grenzen gesetzt.
- **CluedoRobot**: Ein CluedoRobot ist ein computergesteuerter, automatischer Cluedo-Spieler. Er agiert nach einer "eingebauten" Strategie und führt seine Züge autonom ohne Benutzerinteraktion aus. Zusätzlich bietet er Funktionalität für die Anmeldung an einem Server sowie für die Auswahl oder Erzeugung einer Session.

	Programmiermethodik © Prof. Dr. W. Efratsberg	1. Einführung	1-6
---	--	---------------	-----

## Vier Phasen der Entwicklung

Entwurf und Implementierung der Komponenten erfolgen in vier Phasen, die jeweils durch einen Meilenstein abgeschlossen werden:

1. Pflichtenheft
2. Entwurfsspezifikation in UML
3. Implementierung in Java
4. Integrationstest und Abnahme

	Programmiermethodik © Prof. Dr. W. Effeisberg	1. Einführung	1-7
---	--	---------------	-----

## Organisation des Praktikums

- Wie in der industriellen Softwareentwicklung erfolgt die Arbeit in Gruppen (Teams), die sich selbst organisieren. Jede Gruppe wird von einem Tutor betreut.
- Jede Gruppe hat die Aufgabe, eine der drei Komponenten Server, Client oder Robot zu entwerfen und zu implementieren. Jede Instanz einer Komponente muss mit jeder Instanz einer anderen Komponente interaktionfähig sein.
- Wir stellen eine Spezifikation des Kommunikationsprotokolls zwischen Client und Server zur Verfügung. Es ist zwingend notwendig, dass alle Implementierungen dieses Protokoll unterstützen.

	Programmiermethodik © Prof. Dr. W. Effeisberg	1. Einführung	1-8
---	--	---------------	-----

## Metagruppen

- Mehrere Gruppen werden zu Metagruppen zusammengefasst. In jeder Metagruppe gibt es Gruppen, die Server, Clients und Robots entwickeln. Die Metagruppen haben den Zweck, Tests zwischen Client-, Server- und Robot-Entwicklern zu ermöglichen. Zusätzlich kann Funktionalität identifiziert werden, die in verschiedenen Modulen benötigt wird. So benötigen beispielsweise sowohl Client als auch Robot die gleiche Kommunikationsfunktionalität.
- Eine Metagruppe wird jeweils von genau einem Tutor/einer Tutorin betreut. Die Koordination innerhalb der Metagruppe wird durch den Tutor organisiert.

	Programmiermethodik © Prof. Dr. W. Effeisberg	1. Einführung	1-9
---	--	---------------	-----

## Abgaben und Kolloquien

- **Abgaben**  
Zum Ende jeder Phase findet eine Abgabe statt. Zu diesen Terminen muss jede Gruppe bestimmte Dokumente und Implementierungen angefertigt haben. Die Ergebnisse werden jeweils den Tutoren vorgestellt, dabei besteht Anwesenheitspflicht für alle Gruppenmitglieder.
- **Kolloquium**  
Am Ende des Semesters findet für jede Gruppe ein Kolloquium statt. Hier präsentiert die Gruppe die Ergebnisse des Praktikums vor dem jeweiligen Tutor **und Lehrstuhlmitarbeitern**. Es besteht Anwesenheitspflicht für alle Gruppenmitglieder. Jedes Gruppenmitglied muss bei der Präsentation mitwirken und Fragen beantworten können. Der Schein für die Lehrveranstaltung Programmiermethodik wird bei erfolgreichem Bestehen des Kolloquiums vergeben.

	Programmiermethodik © Prof. Dr. W. Effeisberg	1. Einführung	1-10
---	--	---------------	------

## Anforderungen an ein Softwareprodukt

- Korrektheit
- Robustheit
- Gute Bedienbarkeit
- Effizienz
- Wartbarkeit
- Testbarkeit
- Wiederverwendbarkeit
- Portierbarkeit

	Programmiermethodik © Prof. Dr. W. Effeisberg	1. Einführung	1-11
---	--	---------------	------

## Teamorganisation

- **1 Chefprogrammierer**
  - Organisation des Teams
  - Verteilung der Aufgaben
  - Betreuung
  - Überwachung
- **n Programmierer**
  - Feinentwurf
  - Programmierung der Module
- **1 Dokumentierer**
  - Spezifikationen erstellen und verwalten
  - Programmdokumentation erstellen
  - Benutzerhandbuch erstellen
- **1 Qualitätssicherer**
  - Code Review (Code-Inspektion)
  - Testfälle erstellen
  - Tests durchführen und dokumentieren

	Programmiermethodik © Prof. Dr. W. Effeisberg	1. Einführung	1-12
---	--	---------------	------