# Objektorientierte Programmierung (WI-2024)

- Studienanleitung -

Hochschule für
Wirtschaft und Recht Berlin
Berlin School of Economics and Law

Im Sommersemester 2023 findet die Vorlesung ausschließlich in Präsenz an der HWR Berlin am Campus Lichtenberg statt. Nur für ausgewählte Situation (z.B. Krankheit) kann auf das Big-Blue-Blutton Konferenzsystems (Link siehe moodle) zurückgegriffen werden. Die inhaltliche Orientierung der durchgeführten Termine kann dem Veranstaltungsplans auf Seite 9 entnommen werden!

**Achtung** - Die Abschlussnote wird auf der Grundlage <u>einer Kurzübung</u> (20%), <u>eines Reviews</u> (20%) und der schließlich der <u>Belegarbeit</u> (60%) realisiert!

Prof. Dr.-Ing. habil. Andreas Schmietendorf

HWR Berlin, Fachbereich II Alt-Friedrichsfelde 60, D-10315 Berlin

Web: https://blog.hwr-berlin.de/schmietendorf

Email: andreas.schmietendorf@hwr-berilin.de

# Inhaltsverzeichnis

1	Moti	Motivation		
2	Inhal	te der Vorlesung	3	
	2.1	Objektorientierte Programmierung	3	
	2.1.1		3	
	2.1.2			
	2.1.3	Vererbung & Polymorphie	4	
	2.1.4	Schnittstellen-Konzept (interface)	4	
	2.1.5	Persistente und transiente Objekte	4	
	2.1.6			
	2.1.7	Collections - Datenstrukturen	5	
	2.1.8	Java Generics	5	
	2.2 V	Vorgesehene Exkurse (Optional)	5	
	2.3.1	GUI-Programmierung mit Java-SWING/JavaFX	5	
	2.3.2	Unit-Test und Verwendung von jUnit	5	
	2.3.3	Möglichkeiten des Debuggings unter Eclipse	5	
4	Infor	mationen zur Projektarbeit	6	
	4.1 Ü	Übersicht zur Projektarbeit	6	
	4.2	Organisatorische Bewertungskriterien	6	
	4.3 F	Formale Bewertungskriterien	7	
5	Litera	aturhinweise	7	
P1	an der V	Vorlesungen – SoSe2023	9	
Vi	rtuelles	Klassenzimmer – SoSe2023	10	

### 1 Motivation

Die objektorientierte Softwareentwicklung (kurz OOSE) ist aktuell eine der wichtigsten Vorgehensweisen (Paradigma) zur industriellen Erstellung benötigter Softwarelösungen. Dem entsprechend nimmt die Vermittlung von Wissen in den Informatik-Curricula der Hochschulen und Universitäten einen breiten Raum ein. Die hier vorgestellte Vorlesung wurde für Bachelor-Studenten der "Wirtschaftsinformatik" an der HWR Berlin konzipiert. Aufbauend auf den im 1. Semester vermittelten Grundlagen zur strukturierten Programmierung bzw. Algorithmen und Datenstrukturen, wird jetzt speziell auf die Möglichkeiten der objektorientierten Programmierung Bezug genommen. Für ausgewählte Sachverhalte wird auf UML-Notationselemente zurückgegriffen, der umfängliche Einsatz von UML ist allerdings erst Gegenstand der aufbauenden Vorlesung zum Thema "Objektorientierte Systemanalyse und Entwurf" im 3. Semester. Die Beispiele und Übungen zur Codierung werden mit Hilfe von Java realisiert, eine entsprechende Abbildung auf Microsoft C# oder auch JavaScript kann aufgrund der Sprachnähe leicht realisiert werden. Die teamorientiert herauszuarbeitende Belegarbeit sollte sich an agilen Werten, Prinzipien und Techniken, typisch für Scrum- oder auch XP-Projekte orientieren.

# 2 Inhalte der Vorlesung

# 2.1 Objektorientierte Programmierung

### 2.1.1 Motivation zur objektorientierten Softwareentwicklung

- Einordnung und Abgrenzung des objektorientierten Programmierparadigma
- Historische Entwicklung des objektorientierten Programmierparadigma
- Erwartete Vorteile der objektorientierten Softwareentwicklung
- Kernkonzepte der objektorientierten Programmierung
- Aktuelle Programmiersprachen entsprechend des objektorientierten Paradigmas

### 2.1.2 Klassen, Objekte und Methoden

- Abstraktion in der objektorientierten Softwareentwicklung
- Klassifizierung fachlicher Anwendungsdomänen
- Klassen als Baupläne für Objekte der realen Welt
- Konzept von Klassen
  - Trennung von Konzeption und Umsetzung
  - Definition von Klassen in Java
  - Klassen als Vorlage für eine Menge von Objekten
- Kapselung der Objekteigenschaften über Methoden
  - Objekte als Instanz einer Klasse
  - Destruktor- und Konstruktormethode
  - Zugriffsmethoden (getter und setter)

### 2.1.3 Vererbung & Polymorphie

- Vererbung
  - Vererbung Begriff der abgeleiteten Klassen (Subklassen)
  - Überlagerung (Override) von Attributen und Methoden
  - Elementzugriffe und Einschränkungen
- Polymorphie
  - Polymorphe Verwendung von Objekten
  - Statische und dynamische Bindung (late binding)
  - Abstrakte Klassen und Methoden (u.a. Programmiervorgaben)

### 2.1.4 Schnittstellen-Konzept (interface)

- Grundkonzept (Nachbildung der Mehrfachvererbung)
- Definition (Spezifikation) und Implementierung
- Beispiele für die Anwendung von Interfaces
- Aktuelle Entwicklungstendenzen (Exkurs)
  - Java RMI (Remote Method Invocation Interface)
  - CORBA (Common Object Broker Architecture)
  - WSDL-Konzept (Web Services)
  - Swagger/OpenAPI (REST based Web APIs)
  - GraphQL orientierte Serviceschnittstellen
- Organisation einer arbeitsteiligen Softwareentwicklung

#### 2.1.5 Persistente und transiente Objekte

- Serialisierung von Objekten
  - Schreiben und Lesen eines Objektes unter Verwendung einer Datei
  - Senden und Empfangen eines Objektes über eine Netzwerkverbindung
- Umgang mit Objektreferenzen
  - Potentielle Probleme
  - Eigenschaft des Serialisierungs-API im JDK
- Weiterführende Aspekte
  - Ausschluss der Serialisierung
  - Versionierung von Objekten
- Grundlegende Idee eines Persistence Layers (Exkurs)

#### 2.1.6 Exceptions – Ausnahmebehandlung in Java

- Grundlagen, Begriffe und Ziele
- Behandlung von Exceptions (try & catch)
- Verwendung spezialisierter Fehlerklassen
- Performanceeigenschaften der Ausnahmebehandlung

#### 2.1.7 Collections - Datenstrukturen

- Traditionellen Collections (seit dem JDK 1.0)
  - Vektor
  - Stack
  - Hashtable
- Java Collections Framework (JCF seit dem JDK 1.2)
  - ArrayList (variables Array)
  - LinkedList (vgl. verkettete Liste
  - HashMap (vgl. Key Value Store)
  - HashSet (Vermeidung von Dublikaten)
- Einsatz von foreach-Schleifen

#### 2.1.8 Java Generics

- Probleme eines Typecasts zur Laufzeit (Autoboxing)
- Typisierte Collections (u.a. diamond operator ab Java 7)
- Einsatz generischer Klassen und Methoden
- Typeinschränkungen (bounded type parameter)
- Einsatz von Wildcards

### 2.2 Vorgesehene Exkurse (Optional)

Im Rahmen der Vorlesung werden neben den bereits aufgezeigten Themen entsprechende Exkurse eingeschoben. In Abhängigkeit des zur Verfügung stehenden Zeitbudgets werden dabei weiterführende Themen der Java-Implementierungstechnologie vermittelt.

### 2.3.1 GUI-Programmierung mit Java-SWING/JavaFX

- Klassenbibliotheken und Beispiele
- Verwendung von GUI-Buildern
- Alternative Lösungsansätze

### 2.3.2 Unit-Test und Verwendung von jUnit

- Überblick zum JUnit Framework
- Beispielhafte Verwendung

## 2.3.3 Möglichkeiten des Debuggings unter Eclipse

- Funktionalitäten und Einsatzszenarien
- Beispielhafte Verwendung

# 4 Informationen zur Projektarbeit

# 4.1 Übersicht zur Projektarbeit

Die zu erbringende Leistung im Modul "Objektorientierte Programmierung" besteht in der semesterbegleitenden Durchführung eines Softwareentwicklungsprojekts in einer Gruppe von maximal 4 Teilnehmern. Jede Gruppe erhält eine Aufgabenstellung, die entweder auf der Grundlage eines eigenen Vorschlages (frühestens zum Ende Mai) oder aber durch ein vom Dozenten vorgeschlagenes Thema festgelegt werden kann. In jedem Fall sind die daten- und funktionsbezogenen Anforderungen mit dem Dozenten abzustimmen! Die Aufgabenstellung ist während des Semesters (insbesondere im Kontext des letzten Drittels) in Teilschritten, angepasst an die Inhalte der Vorlesung, zu lösen. Die Bewertung der individuellen Leistung findet dann anhand dreier Teilleistungen statt:

**Projektvortrag** - Jede Gruppe muss während des Semesters den anderen Teilnehmern ihre Aufgabenstellung präsentieren. Dieser Vortrag wird mit 20% der Gesamtleistung bewertet und sollte sich an den folgenden Aspekten orientieren:

- Motivation, Zielstellung und Projektplanung
- Analytische Problembetrachtung (User Stories)
- Architekturüberlegungen und Reflektion der 8 behandelten Themen
- Prototypischer Lösungsansätze (unter Nutzung von Java)
- Qualitätssicherung der Implementierung
- Offene Probleme und kritische Reflektion

**Projektbericht** - Jede Gruppe liefert ca. <u>einen Monat</u> nach dem Ende der Vorlesungen einen Projektbericht über Moodle ab. Die bereits im Zusammenhang mit dem Projektvortrag aufgezeigten Aspekte sind innerhalb des Belegs im Sinne zu bearbeitender Phasen umfänglich zu berücksichtigen! Der Projektbericht wird mit 60% der Gesamtleistung bewertet.

**Kurzübungen** - Jede Gruppe hat während des Semesters mindestens eine Kurzübung zu präsentieren (eigenständige Auswahl)! Die Bewertungen gehen zu 20% in die Gesamtleistung ein. Bitte berücksichtigen Sie die limitierten Möglichkeiten zur Vorstellung der Kurzübungen innerhalb des Verlegungsplans!

Insgesamt erhalten Sie somit 3 Bewertungen (0,2\*Kurzvortrag+0,2\*Vortrag+0,6\*Beleg) aus denen sich die Gesamtbewertung ergibt.

**ACHTUNG** – Sowohl die gehaltenen Vorträge (Kurzübungen und Projekt) als auch die Projektarbeit werden mit Hilfe der folgenden Kriterien bewertet:

Inhalt, Aufgabentreue, Komplexität, Kritische Reflektion, Struktur/Stringenz, Teamorientierung/Präsentationsstil, Zeitnutzung, Fragen.

Die inhaltliche Bewertung der erfolgt auf der Grundlage der berücksichtigten Vorlesungsinhalte! Sofern von den vermittelten Lösungsansätzen abgewichen wird, ist dieses unter Bezugnahme auf damit einhergehenden Vor- und Nachteile <u>exakt</u> zu begründen.

# 4.2 Organisatorische Bewertungskriterien

Es ist zumindest ein grober Projektplan zu erstellen, der deutlich zeigt, welche Phasen des Projekts wann geplant sind und welche Aufgaben von den Gruppenmitgliedern übernommen werden. Außerdem ist auf einer gesonderten Übersicht (ein DIN A4-Blatt) zu notieren, welches

Gruppenmitglied welche Aufgaben übernommen hat. Beide Dokumente sind den Projektdokumenten hinzuzufügen. Aus dem Titelblatt der Projektdokumente müssen deutlich die Matrikelnummern der Gruppenmitglieder hervorgehen.

Die Abgabe der Dokumente erfolgt ausschließlich via Upload in Moodle. Dabei sind sämtliche Dokumente, Skizzen/Modelle, Quellcodes sowie Testergebnisse bzw. Testdaten zu berücksichtigen. Bitte stellen Sie eine eindeutige Team-Zuordnung sicher.

### 4.3 Formale Bewertungskriterien

Die gestellte Aufgabe ist in einer Projektgruppe von maximal 4 Teilnehmern zu bearbeiten. Es ist bei der Projektplanung darauf zu achten, dass jeder Teilnehmer für eine ausreichend große Arbeitsmenge eingeplant wird.

Die folgende Aufzählung verdeutlicht die Bewertungsschwerpunkte, dem entsprechend sollten sich diese auch bei der Verteilung des Arbeitsumfangs im Projekt wiederspiegeln.

- Form (10%):
  - o Rechtschreibung, Layout, Übersicht, allgemeiner Eindruck
- Analytische Problemstellung (10%):
  - o Bearbeitungstiefe, Abgrenzung, User Stories
- Architekturüberlegungen (10%):
  - Übersichtlichkeit, Abstraktion, Geschick
- Codierung (40%):
  - o Funktionsumfang, Verständlichkeit, Wartbarkeit, Wiederverwendbarkeit, Komplexität und Funktionsfähigkeit
- Analytische und konzeptionelle Qualitätssicherung (20%):
  - o Programmierkonventionen, Pattern, Testarten
- Benutzer-/Installationshandbuch (10%):
  - o Verständlichkeit, Übersicht

Bei einer Gruppenstärke von 4 Mitgliedern sollten pro Mitglied mindestens 20% Gesamtanteil, bei 3 Mitgliedern mindestens 30% Gesamtanteil pro Mitglied übernommen werden.

Die Bewertung der Projektarbeit erfolgt bei erkennbar gleichmäßiger Arbeitsverteilung und vergleichbarem Engagement der einzelnen Mitglieder möglichst einheitlich. Bei erkennbaren Abweichungen in der Leistung erfolgt jedoch auch eine differenzierte Bewertung.

### 5 Literaturhinweise

Die im Folgenden kurz vorgestellten Literaturhinweise bieten eine Übersicht zu weiterführenden Information, zu den im Kontext der Vorlesung vermittelten Themengebieten. Die dabei gewählte Reihenfolge widerspiegelt die Stärke des Bezugs zur Vorlesung, d.h. auf die Inhalte der letzt genannten Bücher wird im Rahmen der Vorlesung nur geringer Bezug genommen. Darüber hinaus finden sich in den Folien zur Vorlesung weitere Quellenangaben zu ausgewählten Problemstellungen.

[Krüger 2014] Krüger, G.; Hansen, H.: Java-Programmierung - Das Handbuch zu Java 8, Addison-Wesley, München, 2014 (ISBN 978-3-95561-514-7)

- [Brügge 2004] Brügge, B.; Dutoit, A. H.: Objektorientierte Softwaretechnik mit UML, Entwurfsmustern und Java, Pearson Studium, München 2004 (ISBN 3-8273-7261-5)
- [Inden 2012] Inden, M.: Der Weg zum Java-Profi Konzepte und Techniken für die professionelle Java-Entwicklung, dpunkt.verlag, Heidelberg 2012 (ISBN 978-3-86490-005-1)
- [Inden 2015] Inden, M.: Java 8 Die Neuerungen (Lambdas, Streams, Date and Time API und JavaFX 8), dpunkt.verlag, Heidelberg 2012 (ISBN 978-3-86490-290-1)

# 6 Kurs zur OO-Programmierung am HPI



Abbildung 1: Kurs am HPI - https://open.hpi.de/courses/javaeinstieg2020

# Plan der Vorlesungen – SoSe2023

Termin	Inhalt	Bemerkung und Hinweis
18.04.2023	Einführung zur Vorlesung	
03.05.2023	Klassen, Objekte und Methoden	
10.05.2023	Vererbung und Polymorphie	2 Möglichkeiten zur Vorstellung von Kurzübungen
12.05.2023	Schnittstellen-Konzept (interfaces)	2 Möglichkeiten zur Vorstellung von Kurzübungen
16.05.2023	Persistente und transiente Objekte	2 Möglichkeiten zur Vorstellung von Kurzübungen
23.05.2023	Java Ausnahmebehandlung	2 Möglichkeiten zur Vorstellung von Kurzübungen
31.05.2023	Exkurs: Desktop GUI-Programmierung (Swing, JavaFX)	2 Möglichkeiten zur Vorstellung von Kurzübungen
05.06.2023	Java Collections (Einführung/Basis)	4 Möglichkeiten zur Vorstellung von Kurzübungen
06.06.2023	Java Collections (Java Collection Frameworks)	4 Möglichkeiten zur Vorstellung von Kurzübungen
12.06.2023	Java Generics	
14.06.2023	Reviews zur Belegarbeit	max. 4 Reviews
19.06.2023	Reviews zur Belegarbeit	max. 2 Reviews
21.06.2023	Reviews zur Belegarbeit	max. 4 Reviews

<u>Bem.</u>: Jedes Team muss <u>mindestens</u> eine realisierte Kurzübung (d.h. Antworten auf Fragezeichen eines Vorlesungsparts oder eines Exkurses) in den Vorlesungen präsentieren. Die Dauer der Präsentation ist auf max. 10 min. sowie zzgl. 5 min. Diskussion begrenzt. Die letzte Präsentationsmöglichkeit ist am 06.06.2023.

Bitte berücksichtigen Sie, dass die Anzahl der möglichen Vorstellungen je Vorlesung entsprechend der o.g. Tabelle limitiert ist!

### Virtuelles Klassenzimmer - SoSe2023

Wird nur nach ausdrücklicher Bekanntgabe durch den Dozenten und in Abstimmung mit allen Studierenden genutzt!

#### Standard-Klassenzimmer:

Link für das "BigBlueButton" – Konferenzsystem:

https://moodle.hwr-berlin.de/mod/bigbluebuttonbn/view.php?id=998450

- Bitte geben Sie Vor- und Nachname beim Betreten des virtuellen Klassenzimmers ein.
- Bitte schalten Sie das Mikrophon auf lautlos!
- Nutzen Sie die Chat-Funktion für Anmerkungen und Fragen.