

# Objektorientierte Programmierung

## Konzepte imperativer und objektorientierter Programmierung

Vorkurs + Vorlesung + Übungen

SoSe 2020

Prof. Dr. Margarita Esponda

## 1. Teil (Vorkurs)

### Konzepte imperativer Programmierung

- Syntax und operationelle Semantik imperativer Programmiersprachen
- Python als Beispiel

## 2. Teil

### Formale Verfahren zur Spezifikation und Verifikation imperativer Programme

- Bedingungen im Zustandsraum (*assertions*)
- Hoare-Kalkül und partielle Korrektheit von Programmen
- Terminiertheit von Programmen/Algorithmen

### Analyse von Laufzeit und Speicherbedarf

## **3. Teil**

**Konzepte objektorientierter Programmierung (Java)**

**Grundlegende Datenstrukturen und Datenabstraktion**

**Programmierungsmethodik**

**Berechenbarkeit**

## 3. Teil

### Konzepte objektorientierter Programmierung (Java)

- Primitive und Zusammengesetzte Datentypen,
- Methoden (Prozeduren und Funktionen), Parameterübergabe, Überladung
- Module, Klassen, Objekte
- Klassenhierarchien, Vererbung, abstrakte Klassen, Schnittstellen, Polymorphie
- Ausnahmebehandlung (Exceptions)

## 3. Teil

### Programmiermethodik

- Umwandlung von Rekursion in Iteration
- Teile und herrsche
- Rücksetzverfahren (*Backtracking*)
- Dynamische Programmierung

### Analyse von Laufzeit und Speicherbedarf

- O-Notation
- Analyse von Such- und Sortieralgorithmen
- Algorithmen und Datenstrukturen

### Berechenbarkeitsbegriff

- Universelle Registermaschinen

Unsere Lernziele sind:

### **Programmieren im Kleinen**

Algorithmen entwerfen und implementieren

### **Grundkonzepte von imperativen Programmiersprachen im allgemeinen beherrschen**

Datentypen, Anweisungen, Kontrollstrukturen

Unterprogramme, Funktionen, Parameterübergabe, Zeigertypen und Referenz-Typen

### **Korrektheit von Programmen analysieren**

Tests entwerfen und ausführen

### **Konzepte Objektorientierter Programmierung beherrschen**

Klassen, Objekte, Kapselung, Vererbung,

Polymorphie und Datenabstraktion

### **Theoretische Grundlagen kennen**

Registermaschinen als Berechenbarkeitsmodell und andere Berechnungsmodelle

Hoare-Kalkül zur Spezifikation / Verifikation

# Vorlesungen & Übungen

Grundlegende Regeln zum Erfolg sind:

1. Regelmäßige und aktive Teilnahme in der Vorlesung

**Fragen stellen!**

2. Regelmäßige und aktive Teilnahme an den Übungen

**Selbständige Lösung der Übungsblätter!**

## Tutoren

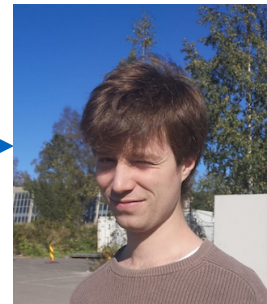
Robin Niclas Hofmann



Simon Franke



Fabian Halama



Maximilian Glinka





## Kriterien für die Bestätigung der aktiven Teilnahme:

1. Eine **regelmäßige Teilnahme** an den **Online**-Übungsterminen ist unerlässlich.
2. Die Abgabe der Übungsblätter erfolgt **Dienstags** um **12:10 Uhr in Whiteboard**.  
Nach diesem Termin können Übungsblätter nicht mehr abgegeben werden.
3. Jeder Student muss an **mindestens n-2 Übungsterminen anwesend** sein.
5. Jeder Student muss **60% der maximal erreichbaren Punktzahl** aus allen Übungsblättern erreichen.
6. Jeder Student muss mindestens **zwei Mal seine Lösungen präsentieren**.
7. Alle Übungsblätter müssen **bearbeitet** werden, indem jeweils mindestens **25% der Punktzahl** erreicht wird.
8. Alle Übungsblättern müssen **On-Line in Whiteboard** abgegeben werden.  
Abgabe von Übungsblättern per Email ist nicht erlaubt.

## Notenvergabe

Am Ende der Veranstaltung findet eine Klausur statt.

(Stoff des Vorkurses + Vorlesung + Übungen)

## Verbesserungsmöglichkeiten

- Zusätzlich ist es den Teilnehmern möglich durch besonders aktive mündliche Teilnahme an den Tutorien die **Gesamtnote um maximal 0,7 zu verbessern**. Diese Bewertung wird von den Tutoren individuell vorgenommen.
- Durch diese Verbesserung kann ein Bestehen der Klausur nicht erreicht werden.
- Die beste zu erreichende Note ist **1,0**.

# Fragen

Inhaltliche Fragen sind immer willkommen  
– während und nach jeder Vorlesung

## Wo?

- Forum
- Email (Tutoren, Dozentin)
- Chat?