

# Programmieren II

Lukas Abelt

[lukas.abelt@airbus.com](mailto:lukas.abelt@airbus.com)

DHBW Ravensburg  
Wirtschaftsinformatik

Ravensburg  
26. Februar 2019

# Inhalt

1 Über mich

2 Vorlesung

3 Datentypen

# Inhalt

1 Über mich

2 Vorlesung

3 Datentypen

# Allgemeines

Wer bin ich?

- Lukas Abelt

# Allgemeines

## Wer bin ich?

- Lukas Abelt
- 21 Jahre (Jahrgang '97)

# Allgemeines

## Wer bin ich?

- Lukas Abelt
- 21 Jahre (Jahrgang '97)
- Ursprünglich aus Werder (Havel)

# Allgemeines

## Wer bin ich?

- Lukas Abelt
- 21 Jahre (Jahrgang '97)
- Ursprünglich aus Werder (Havel)
  - ...in Brandenburg

# Allgemeines

## Wer bin ich?

- Lukas Abelt
- 21 Jahre (Jahrgang '97)
- Ursprünglich aus Werder (Havel)
  - ...in Brandenburg
  - ...bei Potsdam

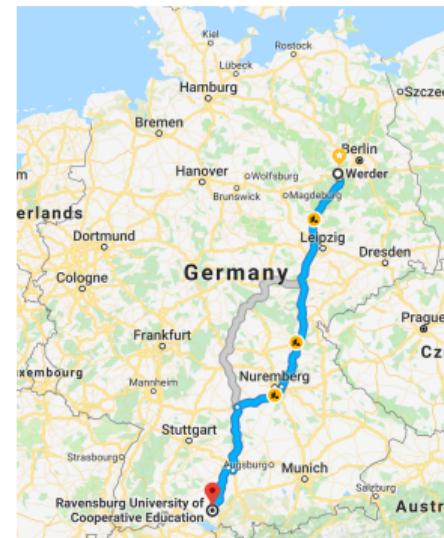
# Allgemeines

## Wer bin ich?

- Lukas Abelt
- 21 Jahre (Jahrgang '97)
- Ursprünglich aus Werder (Havel)
  - ...in Brandenburg
  - ...bei Potsdam
    - ...bei Berlin

# Allgemeines

Wo komme ich her?



Bildquelle: [https://upload.wikimedia.org/wikipedia/commons/4/47/Werder\\_an\\_der\\_Havel.jpg](https://upload.wikimedia.org/wikipedia/commons/4/47/Werder_an_der_Havel.jpg) (Abgerufen: 26.02.2019)

# Beruflicher&Akademischer Werdegang

- **Juli 2015** - Abitur
- **Ab September 2015** - Duales Studium
  - Hochschule: DHBW Ravensburg **Campus Friedrichshafen**
  - Studiengang: Informationstechnik (Mobile Informatik)
  - Firma: Airbus Defence and Space (Immenstaad)
- **September 2018** - Bachelorarbeit und -abschluss
- **Seit Oktober 2018** - Software Architect bei Airbus

# Was habe ich bisher gemacht?

## Praxisphasen

- Arbeit im Bereich SIGINT(Signal Intelligence)
- Implementierung des TCP Stacks zur Übertragung von Signaldaten(C++, Matlab, Simulink)
- Später Abteilungswechsel zu Simulationssoftware
- Implementierung eines neuen Schadenmodells in das bestehende System (C++)
- **Analyse und Bewertung neuer Methoden zur Durchführung simulationsgestützter Parameterstudien mit cloud-basierten Systemen (Bachelorarbeitsthema)**

# Was habe ich bisher gemacht?

## Theoriephasen

- Mathetutorium für Semester 1
- Mathetutorium für Semester 2
- Studienarbeit: Entwickeln eines selbstlernenden Chatbots (Tensorflow, Python)
  - ▣ Erfolgreiche Zielerreichung fraglich?
- 3. Platz beim Bierathlon 2016
- Organisator des alljährlichen Glühweingrillens Friedrichshafen (Nächster Termin: April 2019)

# Was habe ich bisher gemacht?

Studienarbeit

"they're just fucking fucking fucking not just fucking with"

# Inhalt

1 Über mich

2 Vorlesung

3 Datentypen

# Allgemeines

## Skript

- Mit  $\text{\LaTeX}$  erstellt
- Im Druck verfügbar...
  - ...wenn man bezahlt hat
- Digitale Version als PDF verfügbar
- Source Code zum selbst compilieren auf GitHub verfügbar:
  - [https://github.com/LuAbelt/WI18\\_ProgrammierenII](https://github.com/LuAbelt/WI18_ProgrammierenII)

## Git repository clonen

```
git clone https://github.com/LuAbelt/WI18_ProgrammierenII
```

# Allgemeines

Skript

Fork me on GitHub



# Allgemeines

## Organisation

- Insgesamt 60 UE über 15 Termine
- Also 4 UE pro Termin
- Aufteilung (Vorschlag)
  - 2 UE Theorie
  - Kaffeepause
  - 2 UE praktische Anwendung

# Allgemeines

## Organisation

- Zu (fast) jedem Termin wird es eine praktische Aufgabe zur Implementierung geben (Für den zweiten Vorlesungsteil)
- Aufgaben & meine Beispielimplementierung: Nach der Vorlesung im Git Repo zu finden
- Beispielimplementierung wird immer in Java sein
- Hinweise:
  - Ich empfehle IntelliJ als IDE (Für Studenten kostenlose Pro Version)
  - DHBW Rechner haben (leider) nur Eclipse
  - Im Git wird es Projektdateien für bei IDE's geben

# Allgemeines

## Feedback

- Auch ich bin nicht unfehlbar
  - Hauptsächlich in C++ unterwegs
  - Dadurch ggf. Ungenauigkeiten und Fehler bei Java spezifischen Aspekten
- Fragen gerne immer und sofort
- ...Gleiches gilt für (themenbezogene) Diskussionen
- Feedback gerne über alle Kanäle wie zum Beispiel:
  - Persönlich
  - Per Mail
  - Über GitHub/GitLab
  - usw...
- Kontaktdaten am Ende jedes Foliensatzes

# Ziele der Vorlesung I

Laut Modulbeschreibung

## Fachkompetenz

Die Studierenden kennen fortgeschrittene Konzepte objektorientierter Programmiersprachen. Sie besitzen Kenntnisse über wichtige Algorithmen und Datenstrukturen sowie Methoden zur Beurteilung der Effizienz und Qualität von Algorithmen.

# Ziele der Vorlesung II

Laut Modulbeschreibung

## Methodenkompetenz

Die Studierenden können fortgeschrittene Konzepte der Objektorientierung anwenden und autonom mittlere bis größere lauffähige Programme implementieren und testen. Sie sind in der Lage, Algorithmen verschiedener Darstellungsarten zu verstehen und ihre Effizienz bzw. Qualität zu beurteilen, aber auch selbstständig Algorithmen und dazu erforderliche Datenstrukturen zu entwickeln und zu implementieren.

# Ziele der Vorlesung III

Laut Modulbeschreibung

## Personale und Soziale Kompetenz

Die Studierenden können eigenständig Algorithmen und Lösungsverfahren erarbeiten. Sie können stichhaltig und sachangemessen über Konzepte und eigene Algorithmen und deren Implementierung und die damit verbundenen Probleme argumentieren, eigene Umsetzungen plausibel darstellen und eventuelle Fehler nachvollziehbar gegenüber anderen begründen.

# Ziele der Vorlesung IV

Laut Modulbeschreibung

## Übergreifende Handlungskompetenz

Die Studierenden können unter Einsatz der Programmiersprache komplexe praktische Probleme modellieren, algorithmisch behandeln und in anwenderfreundliche und effiziente Lösungen umsetzen. Sie können praktische Problemstellungen analysieren und bekannte Algorithmen und Datenstrukturen effizienzorientiert darauf anwenden und falls notwendig an die Problemstellung anpassen.

# Ziele der Vorlesung V

## In meinen Worten

### Kurzgesagt

Am Ende der Vorlesung sollt ihr mit den vorgestellten Konzepten der fortgeschrittenen Objektorientierung vertraut sein. Dies beinhaltet unter anderem das theoretische Verständnis der zugrundeliegenden Konzepte, sowie die **sprachunabhängige** Anwendung des gelernten. Ziel ist **nicht** das lernen von Java, sondern das übergreifende Verständnis, sodass das gelernte (theoretisch) in jeder Sprache angewandt werden kann!

# Termine & Themen

- 01.04.2019, 14:00-17:15:
  - Vorstellung
  - Allgemeine Informationen zur Vorlesung
  - Datentypen
- 03.04.2019, 14:00-17:15:
  - Generische Interfaces & Klassen
  - Nutzung der Klassenbibliothek
- 08.04.2019, 14:00-17:15:
  - Algorithmen Teil 1
    - Beschreibung
    - Analyse

# Termine & Themen

□ 10.04.2019, 14:00-17:15:

- Listenstrukturen Teil 1
  - Grundoperationen
  - Arrays
  - Verkettete Listen

□ 24.04.2019, 14:00-17:15:

- Listenstrukturen Teil 2
  - Stacks
  - Queues
  - Bäume

# Termine & Themen

□ 25.04.2019, 14:00-17:15:

- Abstrakte Datentypen

- Collections
- Iteratoren

□ 29.04.2019, 14:00-17:15:

- Algorithmen Teil 2

- Suchverfahren

□ 02.05.2019, 14:00-17:15:

- Algorithmen Teil 3

- Sortierverfahren

# Termine & Themen

- 06.05.2019, 14:00-17:15:
  - Algorithmen Teil 4
    - Divide&Conquer
    - Backtracking
- 08.05.2019, 14:00-17:15:
  - Fortgeschrittene Konzepte Teil 1
    - Parallelisierung
    - Synchronisationskonzepte
- 13.05.2019, 14:00-17:15:
  - Fortgeschrittene Konzepte Teil 2
    - Synchronisationskonzepte
    - Ein- und Ausgabe über Streams

# Termine & Themen

- 15.05.2019, 14:00-17:15:
  - Aufbau grafischer Oberflächen Teil 1
    - Layout
    - Typische Komponenten
- 20.05.2019, 14:00-17:15:
  - Aufbau grafischer Oberflächen Teil 2
    - Typische Komponenten
    - Ereignisbehandlung
- 22.05.2019 & 27.05.2019:
  - Puffertermin

# Puffertermine

- Geplante Themen sind nach vorraussichtlich 13 Terminen abgearbeitet
- Sofern es nicht zu Verzögerungen kommt, könnte man diese nutzen für:
  - Wiederholungen
  - Exkurse zu anderen Themen der SW-Entwicklung
  - Freies arbeiten für die Prüfungsleistung
- Gestaltung der Puffertermine frei nach euren Interessen

# Prüfungsleistung

- Keine Klausur, denn:
  - Schon 6 Klausuren dieses Semester
  - Rein theoretische Überprüfungen für Programmieren sowieso eher fraglich
- Stattdessen: **Portfolioprüfung**
  - Besteht aus mehreren Teilleistungen (Hier: 3)
  - Teil 1: Kurztest
  - Teil 2&3: Programmentwurf inklusive Dokumentation

# Prüfungsleistung

## Kurztest

### Teil 1: Kurztest

- Zu Themenblöcken Algorithmen und Datentypen
- Bearbeitungszeit: 30 Minuten
- Noch kein fester Termin gesetzt
- Gewichtung an der Gesamtnote: **20%**

# Prüfungsleistung

## Programmentwurf

### Teil 2: Programmentwurf

- Wird Teilespekte aus (fast) allen Themenblöcken enthalten
  - Vermutlich Schwerpunkt auf Datenstrukturen
  - Vermutlich keine (oder wenig) Teile aus grafischen Overflächen
- Zu bearbeiten in Gruppen von 3-4 Personen
- Bearbeitungszeit: 4 Wochen
- Umsetzung empfohlen in Java
  - Jedoch nicht darauf beschränkt!
  - Bei Interesse Details bei mir erfragen
- Gewichtung an der Gesamtnote: **50%**

# Prüfungsleistung

## Dokumentation

### Teil 3: Dokumentation

- Dokumentation der im Programmentwurf implementierten Lösung
- Bearbeitungszeit: 4 Wochen (Parallel zum Programmentwurf)
- Gleiche Gruppen wie im Programmentwurf
- Gewichtung an der Gesamtnote: **30%**

# Inhalt

1 Über mich

2 Vorlesung

3 Datentypen

# Kontakt

- E-Mail: [lukas.abelt@airbus.com](mailto:lukas.abelt@airbus.com)
- GitHub: <https://www.github.com/LuAbelt>
- GitLab: <https://www.gitlab.com/LuAbelt>
- Telefon(Firma): 07545 - 8 8895
- Telegram: LuAbelt