Grundlagen der Programmierung

TECHNISCHE UNIVERSITÄT DARMSTADT

Vorlesungsskript zum Sommersemester 2020

1. Vorlesung (20. April 2020)

Dr. Jin Gerlach
Christian Olt
Fachgebiet Wirtschaftsinformatik | Software & Digital Business
Fachbereich Rechts- und Wirtschaftswissenschaften
Technische Universität Darmstadt

WIRTSCHAFTS INFORMATIK

Eine Programmiersprache lernen? Wozu eigentlich?

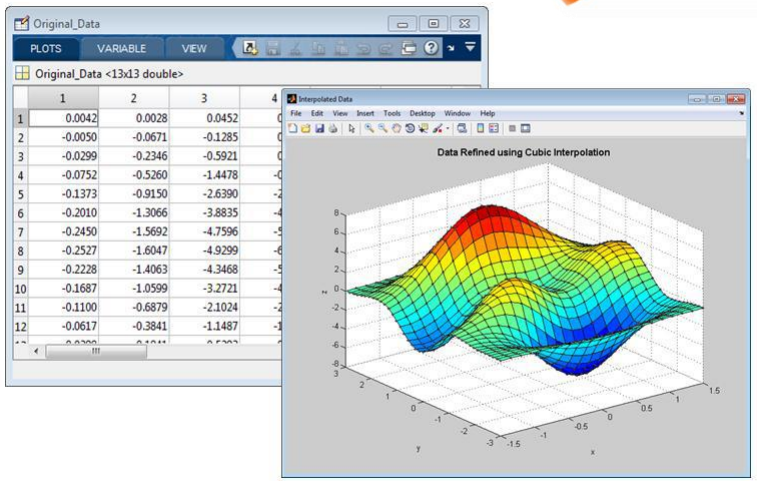
https://www.youtube.com/watch?v=MwLXrN0Yguk

Eine Programmiersprache lernen? Wozu eigentlich? (Perspektive Studium)

TECHNISCHE UNIVERSITÄT DARMSTADT

- Wir lernen die Programmiersprache Java, die grundlegenden Konzepte sind jedoch auf viele andere Programmiersprachen übertragbar, z. B.
  - Matlab
  - R
  - C/C++

- Die Inhalte sind die Grundlage für andere Lehrveranstaltungen (WI-ET: Softwarepraktikum)



WIRTSCHAFTS INFORMATIK

Warum Java?

TECHNISCHE UNIVERSITÄT DARMSTADT

- ✓ Ist in der Praxis weit verbreitet
- ✓ Ist im universitären Umfeld weit verbreitet
- ✓ Ist kostenlos verfügbar
- ✓ Ist für fast alle Betriebssysteme verfügbar
- ✓ Umfasst (fast) alle wichtigen Konzepte einer Programmiersprache

Abstrakte Lernziele

TECHNISCHE UNIVERSITÄT DARMSTADT

Die Studierenden sind nach den Veranstaltungen in der Lage,

- **Grundbegriffe der Programmierung** zu kennen,
- **Operatoren, Daten- und Kontrollstrukturen** zu kennen, um einfache Aufgabenstellungen algorithmisch lösen zu können,
- Grundlagen der **Objektorientierten Programmierung** kennen und anwenden,
- **Fortgeschrittene Konzepte der Objektorientierung** kennen (und anwenden),
- einfache **Java-Programme** zu lesen und zu schreiben und
- Programme mit **UML-Diagrammen** zu modellieren.

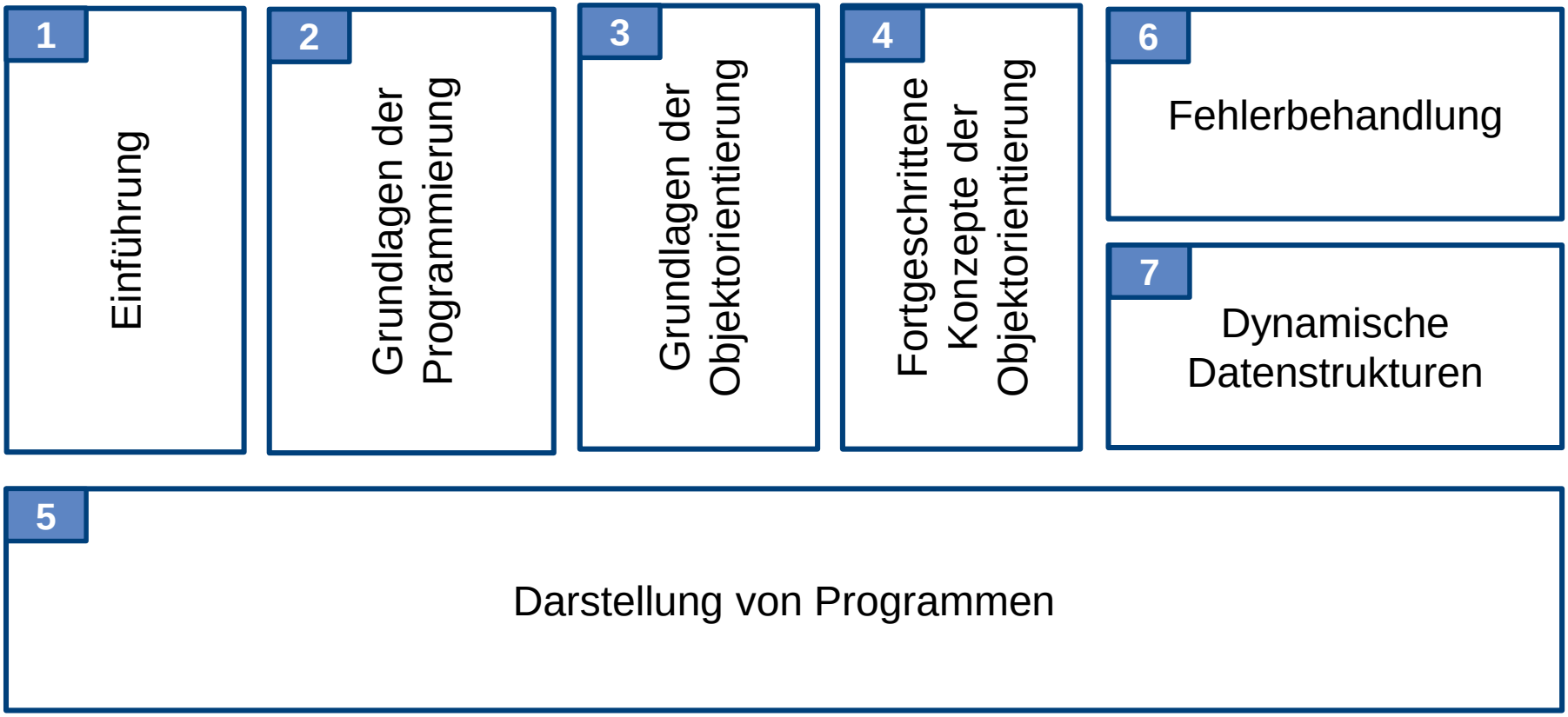
Umfrage zum Vorwissen

TECHNISCHE UNIVERSITÄT DARMSTADT

- Die Vorlesung richtet sich an unerfahrene Programmierer_innen
- Um einen Überblick, über das Vorwissen zu bekommen, führen wir eine Umfrage durch
- Teilnahme über Moodle ab heute (anonym)

Thematische Übersicht

TECHNISCHE UNIVERSITÄT DARMSTADT



WIRTSCHAFTS INFORMATIK

Ergänzende Literatur

TECHNISCHE UNIVERSITÄT DARMSTADT

- **Java ist auch eine Insel (Online-Quelle, auch als Buch vorhanden):**
  http://openbook.galileocomputing.de/javainsel/

- Oracle Java-Tutorials (Online-Quelle):
  https://docs.oracle.com/javase/tutorial/
- Einstieg in Java und OOP (Buch):
  ISBN 978-3-540-78615-3
- Grundkurs Programmieren in Java (Buch):
  ISBN 978-3-446-41268-2
- Java 7 – Das Übungsbuch (Buch):
  ISBN 978-3-8266-9203-1

WIRTSCHAFTS INFORMATIK

Kapitel 1: Einführung

TECHNISCHE UNIVERSITÄT DARMSTADT

- Algorithmen und deren Eigenschaften
- Generationen von Programmiersprachen
- Vom Algorithmus zur Ausführung
- Programmierfehler

Lernziele:

- Algorithmen und deren Eigenschaften erklären und abgrenzen können
- „Generationen“ von Programmiersprachen kennen
- Den Prozess vom Algorithmus zum ausführbaren Programm für die Programmiersprache Java kennen
- Typen von Programmierfehlern kennen und abgrenzen können
- Zusammenhänge der vorgestellten Begriffe erklären können

Algorithmen

TECHNISCHE UNIVERSITÄT DARMSTADT

- **Algorithmen** sind präzise Vorschriften zur Lösung eines Problems, welche die Handlungen und ihre Abfolge beschreiben
- Z. B. Kochrezepte, Bauanleitungen, Spielregeln…
- Ein Algorithmus kann von einem Menschen oder einer Maschine durchgeführt werden
- Wesentliche **Bestandteile** eines Algorithmus:
  - Eine zu bearbeitende Menge von Objekten in einem definierten Anfangszustand
  - Eine an den Objekten auszuführende Menge von Operationen
  - Gewünschter Endzustand der Objekte

Algorithmus: Kochrezept

TECHNISCHE UNIVERSITÄT DARMSTADT

Eingabe

- 3 Eier
- 400g Mehl
- 750 ml Milch
- 1 Prise Salz
- 7 EL Zucker
- 2 Pck. Vanillezucker

Berechnungsvorschrift

- Das Mehl mit Eiern, Milch und Salz zu einem glatten Teig verrühren.
- Beim Rühren den Zucker und den Vanillezucker hinzufügen.
- Sobald alles gut verrührt ist und keine Klumpen mehr da sind, den Teig für 30 Minuten zugedeckt in den Kühlschrank stellen.
- Nach dem Ruhen nochmals verrühren.
- Sonnenblumenöl in einer Pfanne erhitzen und den Teig portionsweise von beiden Seiten zu goldgelben Pfannkuchen backen.

Ausgabe

Photo by Calum Lewis on Unsplash

Algorithmen in der Informatik

TECHNISCHE UNIVERSITÄT DARMSTADT

- **Informatik** = Information + Automatik (d.h. die Informatik als Wissenschaft von der maschinellen Informationsverarbeitung)
- Wichtige (wünschenswerte) Eigenschaften eines Algorithmus:
  - **Korrektheit**
    - Im Allgemeinen kann die Korrektheit durch Testfälle nicht nachgewiesen werden
    - Lediglich die Anwesenheit von Fehlern kann aufgedeckt werden
  - **Effizienz** (d.h. Sparsamkeit bezüglich des Ressourceneinsatzes)
    - Benötigter Speicherplatz
    - Benötigte Rechengeschwindigkeit
  - **Determinismus** (d. h. die nächste anzuwendende Regel ist zu jedem Zeitpunkt definiert)
  - **Terminiert** (d.h. ist in endlicher Zeit beendet)

Beispiel: Der euklidische Algorithmus

TECHNISCHE UNIVERSITÄT DARMSTADT

Algorithmus zur Bestimmung des größten gemeinsamen Teilers (ggT) von Euklid (ca. 300 v. Chr):

```
Solange x ungleich y ist, wiederhole:
  Wenn x größer als y ist, dann:
    ziehe y von x ab und weise das Ergebnis x zu.
  Andernfalls:
    ziehe x von y ab und weise das Ergebnis y zu.
Wenn x gleich y ist, dann:
  x (bzw. y) ist der gesuchte größte gemeinsame
  Teiler.
```

WIRTSCHAFTS INFORMATIK

Beispiel: Der euklidische Algorithmus (II)

TECHNISCHE UNIVERSITÄT DARMSTADT

Gesucht ist der ggT von x=24 und y=9

	Werte von	
Verarbeitungsschritt	**x**	**y**
Initialisierung		
x=24, y=9	24	9
x=x-y	15	9
x=x-y	6	9
y=y-x	6	3
x=x-y	3	3

Ergebnis: ggT = 3

Von Algorithmen zu Programmen

TECHNISCHE UNIVERSITÄT DARMSTADT

- Ein **Programm** ist die Formulierung eines **Algorithmus** in einer **Programmiersprache** als eine Abfolge von Anweisungen, die von einem Prozessor ausgeführt werden können
- Grundelement eines Programms ist die **Anweisung**, die aus einem nicht teilbaren Einzelschritt besteht (z. B. Ein-/Ausgabe von Text)
- Der **Prozessor bearbeitet eine Anweisung** des Programms nach der anderen

Programmiersprachen

TECHNISCHE UNIVERSITÄT DARMSTADT

- **Anweisungen** werden als Text an den Computer übergeben und von diesem ausgeführt
- Der **Programmtext** wird nach den Regeln einer Programmiersprache formuliert (wird dann als **Quellcode** bezeichnet)
- Frühe **Programmiersprachen** arbeiteten mit sehr hardwarenahen Operationen (z. B. Maschinencode, Assembler)
- Höhere (moderne) Programmiersprachen abstrahieren von der Hardware und gestatten eine stärkere Orientierung an dem zu lösenden Problem (C, Java, Python)

Generationen von Programmiersprachen

TECHNISCHE UNIVERSITÄT DARMSTADT

1. Generation: **Maschinensprachen**
   - Interne Sprache eines bestimmten Prozessors
   - Maschinenbefehle sind als Zahlen (Bitmuster) codiert

```
B81000
BB0500
F6E3
```

2. Generation: **Maschinenorientierte Sprachen**
   - Assembler: Maschinenbefehle in einer für Menschen lesbaren Form (Mnemonics)

```
MOV AX,10H
MOV BX,5
MUL BX
```

3. Generation: **Höhere Programmiersprachen**
   - Abstrahieren vom verwendeten Prozessor
   - Das zu lösende Problem steht im Vordergrund
   - Beispiele sind C, Python, Java

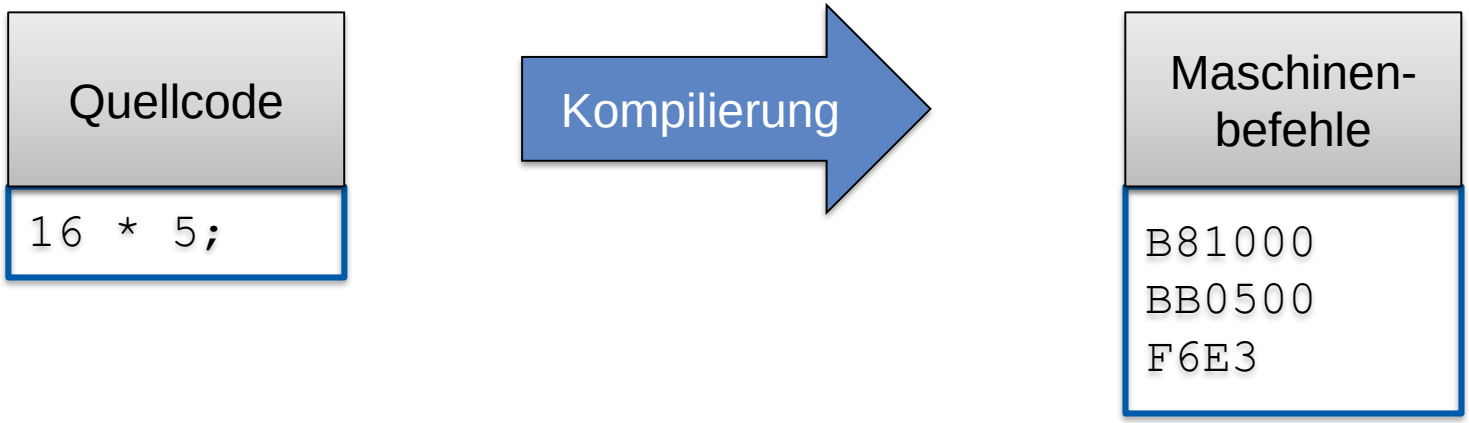
```
16 * 5;
```

WIRTSCHAFTS INFORMATIK

Vom Programm zur Maschine

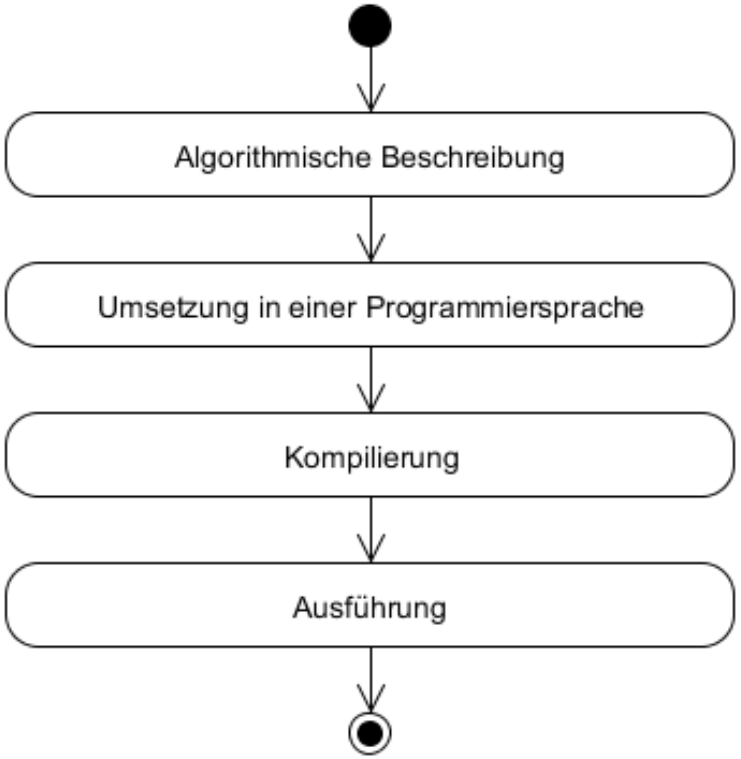
TECHNISCHE UNIVERSITÄT DARMSTADT

- Programme, die in einer höheren Programmiersprache geschrieben sind, können nicht unmittelbar auf einem Rechner ausgeführt werden
- Sie sind anfangs in einer Textdatei gespeichert (**Quellcode**)
- Quellcode muss in **Maschinenbefehle** übersetzt (kompiliert) werden, die der Computer unmittelbar ausführen kann

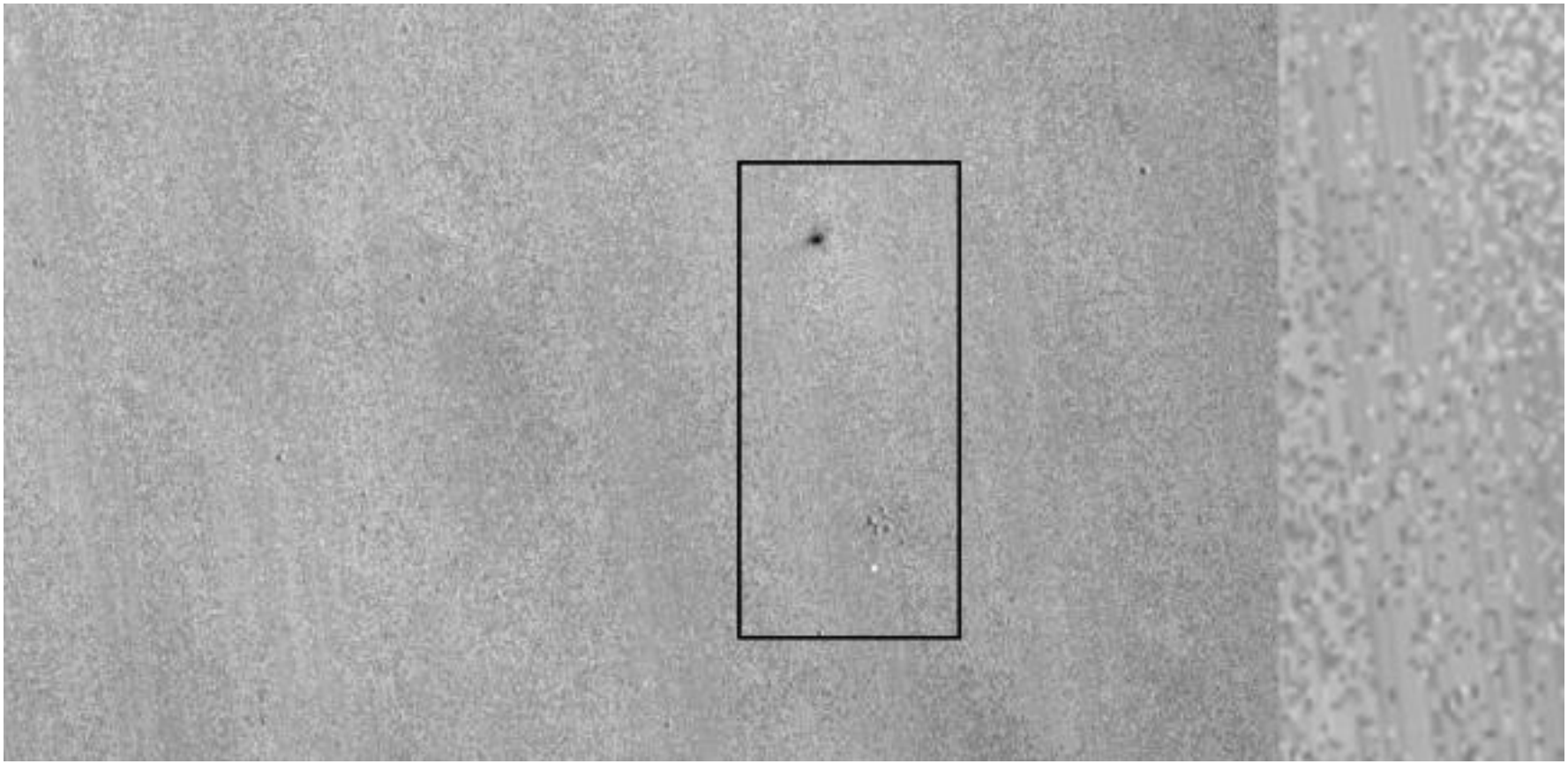


WIRTSCHAFTS INFORMATIK

Zusammenfassung: Programmieren

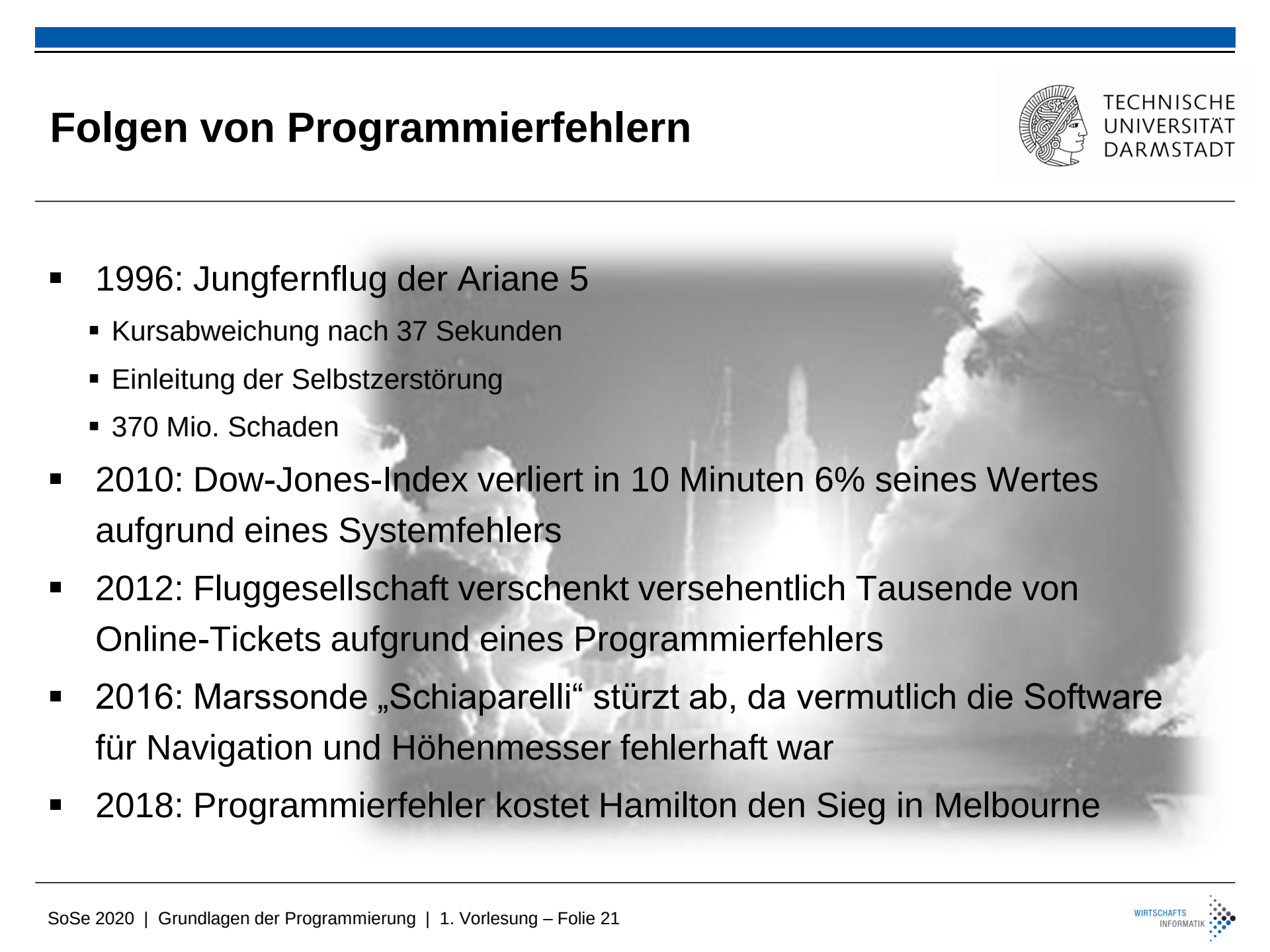


Was ist das?



Quelle: Spiegel Online

Folgen von Programmierfehlern



- 1996: Jungfernflug der Ariane 5
  - Kursabweichung nach 37 Sekunden
  - Einleitung der Selbstzerstörung
  - 370 Mio. Schaden
- 2010: Dow-Jones-Index verliert in 10 Minuten 6% seines Wertes aufgrund eines Systemfehlers
- 2012: Fluggesellschaft verschenkt versehentlich Tausende von Online-Tickets aufgrund eines Programmierfehlers
- 2016: Marssonde „Schiaparelli“ stürzt ab, da vermutlich die Software für Navigation und Höhenmesser fehlerhaft war
- 2018: Programmierfehler kostet Hamilton den Sieg in Melbourne

9/9

0800 Arctan started { 1.2700 9.037 847 025
1000 " stopped - arctan ✓ 9.037 846 995 correct
13"UC (032) MP - MC ~~1.9826447000~~ ~~2.13047764 15 (-3)~~ 4.615925059 (-2)
(033) PRO 2 2.130476415
correct 2.130676415

Relays 6-2 in 033 failed special speed test
in relay " 11.00 test.
Relays changed

1100 Started Cosine Tape (Sine check)
1525 Started Mult + Adder Test.

1545 
Relay #70 Panel F
(moth) in relay.

First actual case of bug being found.

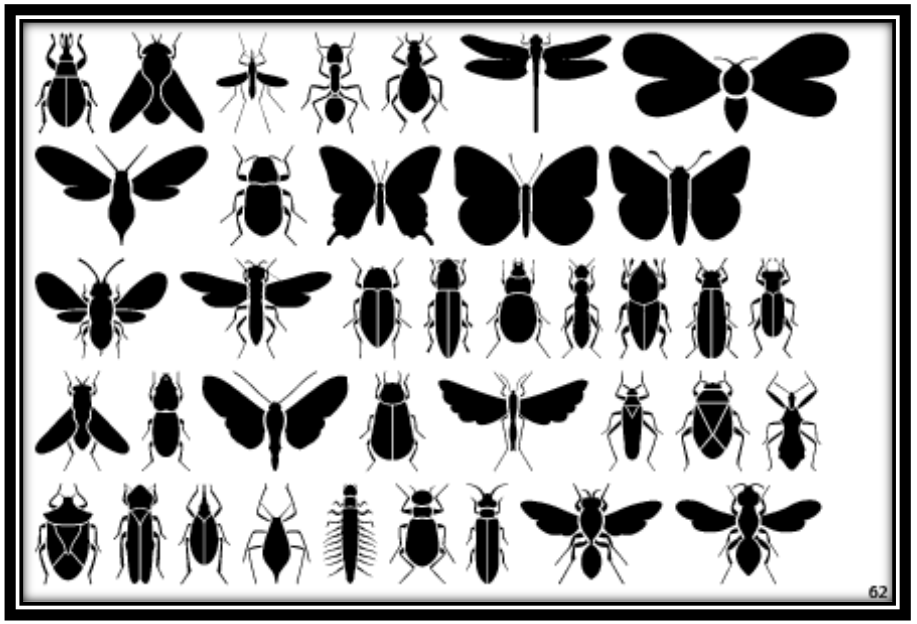
~~1535~~ 1630 Arctangent started.
1700 closed down.

Programmierfehler

TECHNISCHE UNIVERSITÄT DARMSTADT

Programme können Programmfehler (Bugs) enthalten:

- **Syntaxfehler**: Verstöße gegen die Grammatik der Programmiersprache
- **Semantische Fehler**: Korrekte Syntax, jedoch falsches Ergebnis aufgrund einer fehlerhaften Logik



WIRTSCHAFTS INFORMATIK