universität innsbruck

Einführung und Motivation

Programmiermethodik

Lukas Kaltenbrunner, Simon Priller
Universität Innsbruck

Ziele der Vorlesung (1)

- Grundlagen der Programmierung vertiefen
- Anknüpfen an Wissen aus Einführung in die Programmierung
- Grundkonzepte der Objektorientierung verstehen und anwenden
  - Analyse
  - Design
  - Implementierung
- Erlernen einer modernen objektorientierten Programmiersprache
- Verstehen moderner Programmierprinzipien
- Best practices lernen

Ziele der Vorlesung (2)

Nach dieser Vorlesung können Sie:

1. Die Grundkonzepte der objektorientierten Programmierung wiedergeben und anwenden
2. Bestehende Programme analysieren, deren Funktionsweise und die verwendeten Techniken identifizieren
3. Problemstellungen analysieren, beschreiben und daraus die Architektur eines Programms erarbeiten und umsetzen
4. Codequalität bewerten und damit „guten“ von „schlechtem“ Code unterscheiden und dies im eigenen Code auch umsetzen.

Inhalt

- Grundelemente der Programmierung in Java
- Grundlagen der Objektorientierung
- Einführung in UML
- Vererbung und Polymorphie
- Ausnahmenbehandlung
- Unit-Tests
- Java Collections
- Generische Programmierung in Java
- Funktionale Programmierung in Java
- Streams in Java
- Refactoring
- GUI-Programmierung
- Einführung in die Java Virtual Machine

- Clean Code

Allgemeine Literatur

Christian Ullenboom: **Java ist auch eine Insel: Einführung, Ausbildung, Praxis**, Rheinwerk Verlag, 16. Auflage, 2022 (Java 17)
12. Auflage frei verfügbar: http://openbook.rheinwerk-verlag.de/javainsel/

Bernhard Lahres, Gregor Rayman, Stefan Strich: **Objektorientierte Programmierung: Das umfassende Handbuch**, Rheinwerk Verlag, 5. Auflage, 2021
2. Auflage frei verfügbar: http://openbook.rheinwerk-verlag.de/oop/

Joachim Goll, Cornelia Heinisch: **Java als erste Programmiersprache**, Springer Verlag, 8. Auflage, 2016
7. Auflage frei verfügbar (Uni-Netz):
http://link.springer.com/book/10.1007%2F978-3-8348-2270-3

Simon Harrer, Jörg Lenhard, Linus Dietz: **Java by Comparison. Become a Java Craftsman in 70 Examples.** The Pragmatic Bookshelf, 1st Edition, 2018
Frei verfügbar (Uni-Netz): https://bibsearch.uibk.ac.at/AC16131176

Lehrkonzept

Zeit

Vorbereitung

- Vorlesungsfolien
- Vorlesungsvideos
- Quiz lösen
- Fragen stellen/notieren

Vorlesungseinheit

- Fragen stellen
- Wiederholung von schwierigen Themen
- Aktive Lerntätigkeit (Programmieren)

Nachbereitung

- Vertiefung mit Vorlesungsunterlagen

Organisatorisches

- OLAT
  - Vorlesungsfolien, Vorlesungsvideos, Quiz, weitere Links, Ankündigungen
- GIT
  - Alle Beispiele auf https://git.uibk.ac.at/c7031278/PMExamples
- Vorlesung
  - Vorlesungsvideos werden für die entsprechende Woche via OLAT veröffentlicht
  - Fragestunde: Do. 08:15 - 10:00, HS A
- Prüfung
  - Schriftliche Klausur über alle Themen, welche in der Vorlesung besprochen wurden
  - Termine
    - 1. Klausur: 04.07.2022 09:00 – 12:00
    - 2. Klausur: 26.09.2022 09:00 – 12:00

Kontakt & Fragen

- E-Mail: lukas.kaltenbrunner@uibk.ac.at
- Chat Matrix/Element: @lukas.kaltenbrunner:uibk.ac.at
- Sprechstunde: Di. 10:00 – 11:00 nach Vereinbarung
- ARSnova (https://arsnova.uibk.ac.at/mobile/#id/13421235)
- Q&A
- OLAT Forum

ARSnova

Folien - Codebeispiele

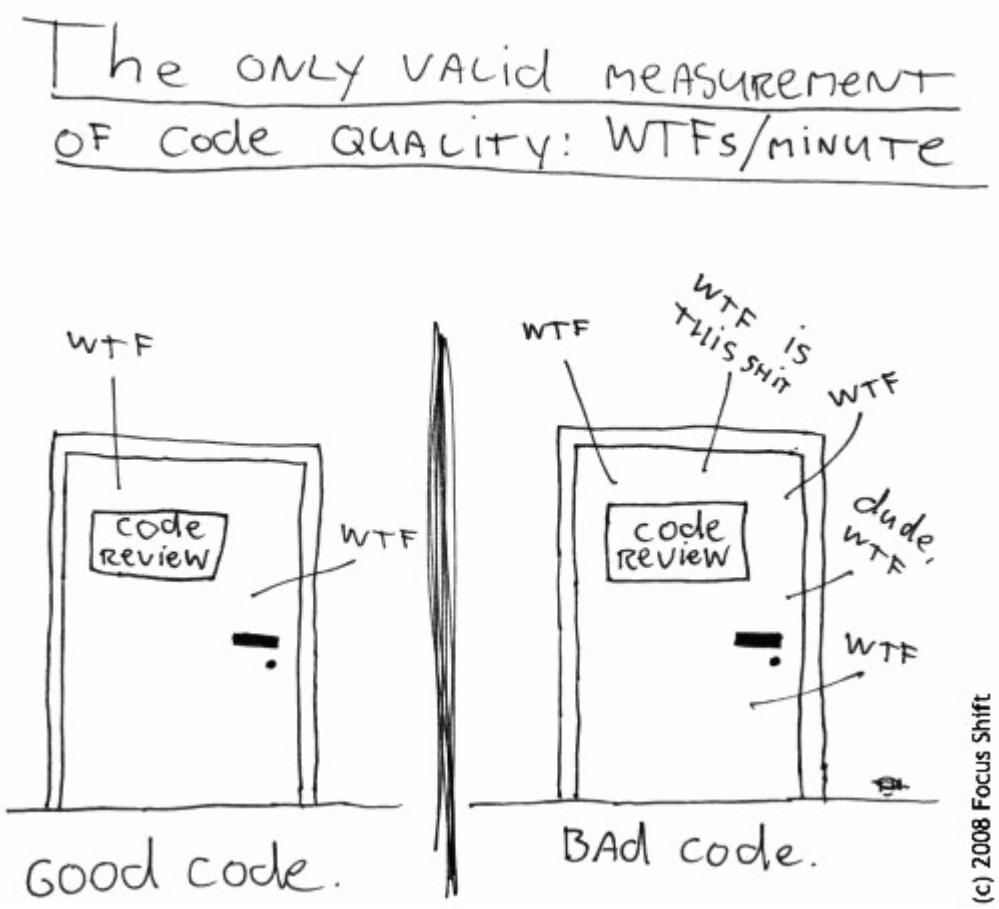
```
private static List<Integer> getFlaggedCells(List<Integer> l) {
    List<Integer> r = new ArrayList<Integer>();
    for (Integer x : l) {
        if (x == 4) {
            r.add(x);
        }
    }
    return r;
}
```

Verbesserungswürdiger Code
(schlechtes Beispiel)

Pfad zu Sourcecode im Git-Repository
(direkt verlinkt; hier nur Platzhalter)

src/at/ac/uibk/pm/basics/HelloWorld.java

Clean Code


The ONLY VALID MEASUREMENT OF CODE QUALITY: WTFs/MINUTE
WTF
code review
WTF
Good code.
WTF
WTF IS THIS SHIT
WTF
code review
dude, WTF
WTF
BAd code.
(c) 2008 Focus Shift

Avoid Single-Letter Names

Clean Code Tipp

Bezeichnung

```
private static List<Integer> getFlaggedCells(List<Integer> gameBoard) {
    List<Integer> flaggedCells = new ArrayList<Integer>();
    for (Integer cell : gameBoard) {
        if (cell == 4) {
            flaggedCells.add(cell);
        }
    }
    return flaggedCells;
}
```

Verbesserter Code (gutes Beispiel)

- Vorher:
  - Leser muss Bedeutung (Semantik) der Variablen selbst herausfinden.
  - Variablen geben keinerlei Auskunft über Inhalt.
- Nachher: lesbarer Code

Proseminar (1)

- Anwesenheitspflicht
- Aufgaben, Präsentation(en) und zwei Tests bestimmen die Note
  - 1. Midterm-Test: Mo. 02.05.2022, 17:15 – 19:00, für alle Gruppen vor Ort!
  - 2. Midterm-Test: Mo. 16.06.2022, 17:15 – 19:00, für alle Gruppen vor Ort!
- OLAT
  - Übungsaufgabe, Quiz, Abgaben, Ankündigungen

Proseminar (2)

- 12 Übungsgruppen:
  - Gruppe 1: Benedikt Hupfauf (Mo, 08:15 – 10:00, rr15)
  - Gruppe 2: Lukas Kaltenbrunner (Mo, 12:15 – 12:00, rr15)
  - Gruppe 3: Lukas Kaltenbrunner (Mo, 14:15 – 16:00, rr15)
  - Gruppe 4: Eduard Frankford (Mo, 16:15 – 18:00, rr15)
  - Gruppe 5: Simon Priller (Mo, 08:15 – 10:00, rr26)
  - Gruppe 6: Simon Priller (Mo, 12:15 – 14:00, rr26)
  - Gruppe 7: Alexander Blaas (Mo, 14:15 – 16:00, rr26)
  - Gruppe 8: Alexander Blaas (Mo, 16:15 – 18:00, rr26)
  - Gruppe 9: Melanie Ernst (Mo, 12:15 – 14:00, rr25)
  - Gruppe 10: Umutcan Simsek (Mo, 14:15 – 16:00, rr25)
  - Gruppe 11: Manfred Moosleitner (Mo, 10:15 – 12:00, eLecture)[EWS]
  - Gruppe 12: Elwin Huaman Quispe (Mo, 17:15 – 19:00, eLecture)[EWS]
- Bitte Übungsgruppen-Zuordnung überprüfen

Credits

- Foliensatz Programmiermethodik (Sommersemester 2012)
  Stefan Podlipnig
- Foliensatz Programmiermethodik (Sommersemester 2013)
  Rene Thiemann
- Foliensatz Programmiermethodik (Sommersemester 2014 – 2020)
  Eva Zangerle

Icons made by Freepik from www.flaticon.com