

252-0027

Einführung in die Programmierung Übungen

Woche 14: Abschluss

Timo Baumberger

Departement Informatik

ETH Zürich





Organisatorisches

- Mein Name: Timo Baumberger
- Website: timobaumberger.com
- Bei Fragen: tbaumberger@student.ethz.ch
 - Mails bitte mit «[EProg24]» im Betreff
- Neue Aufgaben: **Dienstag Abend** (im Normalfall)
- Abgabe der Übungen bis **Dienstag Abend (23:59)** Folgewoche
 - Abgabe immer via Git
 - Lösungen in separatem Projekt auf Git

Programm

- **Bonusaufgabe**
- **Loop-Invariante Aufgabe**
- **Tipps und Tricks für die Prüfung**
- **Abschluss**

Bonusaufgabe Lösung

Loop-Invariante Aufgabe (Übungsblatt 12)

Wichtigste Themen

- EBNF (mehr überprüfen als erstellen)
- Hoare Logik (Weakest Precondition, Loop Invariante)
 - Ist das Hoare Tripel valid?
 - Bestimme die weakest Precondition
 - Bestimme die Loop invariante
- Operator Precendence & Evaluation Order (Evaluation of expressions)
- Java Basics (String, Int, Boolean, Arrays, Loops (For, While, Do While))
 - Schaut euch die Methoden in den Klassen an
- OOP
 - Dynamic Dispatch / Polymorphismus
- Java Collections (Map, Set, List, Comparator und Wrapper Klassen)
 - Macht euch mit HashMap, HashSet, TreeSet, TreeMap, ArrayList und LinkedList vertraut

Wie soll ich eine Programmieraufgabe angehen?

- Lest zuerst die komplette Aufgabenstellung durch
- Seid nicht voreingenommen
- Probiert zum Teil auf Papier aufzuzeichnen, was eure Idee ist
- Bevor ihr etwas programmiert:
 - Seid kritisch mit eurer Idee (eine falsche Idee kostet euch viel Zeit)
 - Probiert gewisse Fälle durchzudenken
- Geht zur nächsten Aufgabe, falls ihr feststeckt

Übliche Struktur

- **Aufgabe 1 (eher leicht, Arrays, String, keine Rekursion, Loops, Matrix Aufgaben)**
 - Wenig Code aber nicht direkt klar, wie die Aufgabe umgesetzt werden kann
- **Aufgabe 2 (mittelschwer, Rekursion, Collections, BFS, DFS, erinnern an Graphen Aufgaben)**
 - Wenig Code aber nicht direkt klar, wie die Aufgabe umgesetzt werden kann
- **Aufgabe 3 (mittelschwer / schwer, Klassen, Interfaces, Enums, Polymorphismus, Collections)**
 - Mehr Code, aber nicht direkt klar, wie die Aufgabe umgesetzt werden kann

Tipps und Tricks für die Prüfung / Lernphase

- Löst alte Theorieprüfungen auf VIS
- Messt die Zeit für die Theorieprüfung
- Die Zeit ist eher knapp
- Investiert mehr Zeit in die Aufgaben, die mehr Punkte geben
- Programmiert viel und löst viele Programmieraufgaben!
- Seid ehrlich mit euch beim Lösen
- Probiert Zeit zu sparen

Happy Holidays

