

Informatik II

Vorlesung am D-BAUG der ETH Zürich

Felix Friedrich & Hermann Lehner

FS 2019

Willkommen!

Vorlesungshomepage:

<http://lec.inf.ethz.ch/baug/informatik2/2019/>

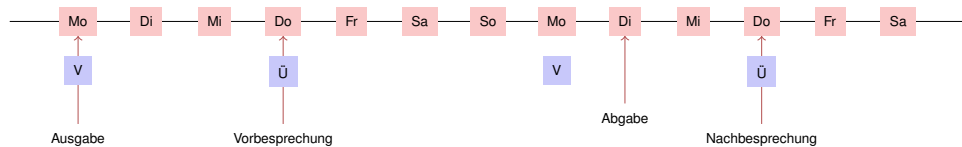
Das Team:

Dozenten	Felix Friedrich Hermann Lehner
Assistenten	Patrick Gruntz Aristeidis Mastoras Chris Wendler Manuel Winkler
Back-Office	Katja Wolff

1

2

Übungsbetrieb



- Übungsblattausgabe zur Vorlesung (online).
- Vorbereitungen in der folgenden Übung.
- Bearbeitung der Übung bis spätestens 2 Tage vor der nächsten Übungsstunde (23:59h).
- Nachbereitung der Übung in der nächsten Übungsstunde. Feedback zu den Abgaben innerhalb einer Woche nach Nachbereitung.

Zu den Übungen

- Bearbeitung der wöchentlichen Übungsserien ist freiwillig, wird aber *dringend* empfohlen!

3

4

Es ist so einfach!

Für die Übungen verwenden wir eine Online-Entwicklungsumgebung, benötigt lediglich einen Browser, Internetverbindung und Ihr ETH Login.

Falls Sie keinen Zugang zu einem Computer haben: in der ETH stehen an vielen Orten öffentlich Computer bereit.

5

Relevantes für die Prüfung

Prüfungsstoff für die Endprüfung schliesst ein

- Vorlesungsinhalt (Vorlesung, Handout) und
- Übungsinhalte (Übungsstunden, Übungsaufgaben).

Prüfung ist schriftlich

Es wird sowohl praktisches Wissen (Kenntnis von Algorithmen, Programmierfähigkeit) als auch theoretisches Wissen (Hintergründe, Systematik) geprüft.

7

Literatur

Algorithmen und Datenstrukturen, *T. Ottmann, P. Widmayer*, Spektrum-Verlag, 5. Auflage, 2011

Algorithmen - Eine Einführung, *T. Cormen, C. Leiserson, R. Rivest, C. Stein*, Oldenbourg, 2010

Introduction to Algorithms, *T. Cormen, C. Leiserson, R. Rivest, C. Stein*, 3rd ed., MIT Press, 2009

Algorithmen Kapiern, *Aditya Y. Bhargava*, MITP, 2019.

6

Unser Angebot

- Bearbeitung der wöchentlichen Übungsserien → Bonus von maximal 0.25 Notenpunkten für die Prüfung.
- Bonus proportional zur erreichten Punktzahl von **speziell markierten Bonus-Aufgaben**. Volle Punktzahl $\hat{=}$ 0.25.
- **Zulassung** zu speziell markierten Bonusaufgaben kann von der erfolgreichen Absolvierung anderer Übungsaufgaben abhängen.

8

Unser Angebot (konkret)

- Insgesamt 3 Bonusaufgaben; 2/3 der Punkte reichen für 0.25 Bonuspunkte für die Prüfung
- Sie können also z.B. 2 Bonusaufgaben zu 100% lösen, oder 3 Bonusaufgaben zu je 66%, oder ...
- Bonusaufgaben müssen durch erfolgreich gelöste Übungsserien freigeschaltet (→ Experience Points) werden
- Es müssen wiederum nicht alle Übungsserien vollständig gelöst werden, um eine Bonusaufgabe freizuschalten
- Details: Übungsstunden, Online-Übungssystem (Code Expert)

Akademische Lauterkeit

Regel: Sie geben nur eigene Lösungen ab, welche Sie selbst verfasst und verstanden haben.

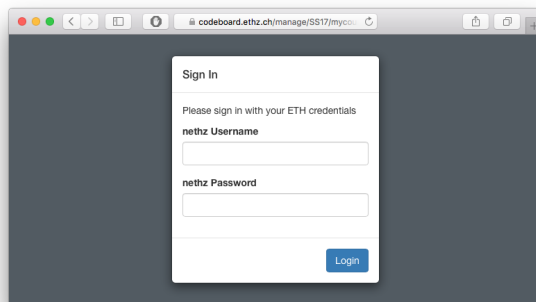
Wir prüfen das (zum Teil automatisiert) nach und behalten uns disziplinarische Massnahmen vor.

9

10

Einschreibung in Übungsgruppen - I

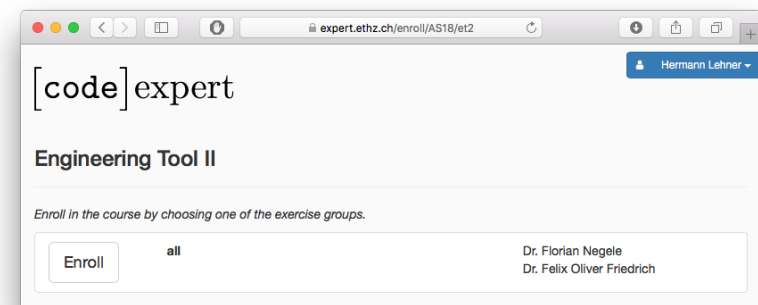
- Besuchen Sie <http://expert.ethz.ch/enroll/SS19/ifbaug2>
- Loggen Sie sich mit Ihrem nethz Account ein.



11

Einschreibung in Übungsgruppen - II

Schreiben Sie sich im folgenden Dialog in eine Übungsgruppe ein.



12

Übersicht

The screenshot shows the [code]expert web interface. At the top, it displays the user's name 'Felix Oliver Friedrich' and the semester 'Autum 2017'. Below this, there are tabs for 'Enrolled Courses', 'My Exercise Groups', and 'My Courses'. The main content area is titled 'Demo Course' and lists 'Coding Demo Exercise' with 'Earned XP' (1,000 / 1,000) and 'Submissions' (1,000 ✓, 100%). It also shows 'Handout Date' (9. Sep. 2017 00:00) and 'Due Date' (31. Dez. 2027 00:00). A specific exercise 'Quadratic Equations In C++' is highlighted with a 'Hand in now' button. Below this, there is a 'Markdown Editor Manual' section with 'Submissions' and 'Handout Date' (1. Aug. 2017 00:00) and 'Due Date' (1. Aug. 2017 00:01).

Programmierübung

The screenshot shows a code editor with C++ code for a min-max problem. The code is as follows:

```
1 #include <iostream>
2
3 int main () {
4     int min; int max;
5     std::cin >> min; std::cin >> max;
6     max = min;
7     for (int i = 0; i < 8; ++i){ // (there is
8         int v;
9         std::cin >> v;
10        if (v < min) min = v;
11        if (v > max) max = v;
12    }
13    std::cout << min << "/" << max << std::endl;
14 }
```

Annotations on the screenshot include:

- A red box labeled 'D: Beschreibung' pointing to the code.
- A red box labeled 'E: History' pointing to the right sidebar.
- A red box labeled 'A: compile' pointing to the compile button (A).
- A red box labeled 'B: run' pointing to the run button (B).
- A red box labeled 'C: test' pointing to the test button (C).

13

14

Testen und Abgeben

The screenshot shows the code editor with test results and a submission button. The test results are as follows:

```
Running tests.....
min_first passed
min_last passed
min_middle passed
max_first failed
max_middle failed
unique passed

Input:
100250 45 0 0 1 -1000001 45 -25065 1
expected output:
-1000001/100251
actual output:
-1000001/100250
.....
max_test passed
max_middle passed
unique passed

Tests result: passed 6 of 7 / score: 86% [ ]
```

Annotations on the screenshot include:

- A red box labeled 'Abgeben' pointing to the 'Submit' button.
- A red box labeled 'Testen' pointing to the test results.

15

Wo ist der Save Knopf?

- Das Filesystem ist transaktionsbasiert und es wird laufend gespeichert („autosave“). Beim Öffnen eines Projektes findet man immer den zuletzt gesehenen Zustand wieder.
- Der derzeitige Stand kann als (benannter) *Snapshot* festgehalten werden. Zu gespeicherten Snapshots kann jederzeit zurückgekehrt werden.
- Der aktuelle Stand kann als Snapshot abgegeben (submitted) werden. Zudem kann jeder gespeicherte Snapshot abgegeben werden.

16

Snapshots

Wenn es Probleme gibt ...

The screenshot shows a coding environment with a C++ program in the editor, test results in the console, and a submission history panel on the right. The program is a simple min-max algorithm. The test results show that all tests passed. The submission history panel shows three snapshots: 'Really Working Version', 'First Working Version', and 'Initial Snapshot'. Annotations with red arrows point to the 'Create Snapshot' button, the 'Really Working Version' entry, and the 'Initial Snapshot' entry.

```
1 #include <iostream>
2
3 int main () {
4     int min; int max;
5     std::cin >> min; std::cin >> max;
6     max = min;
7     for (int i = 0; i < 8; ++i) { // (there is a bug here)
8         int v;
9         std::cin >> v;
10        if (v < min) min = v;
11        if (v > max) max = v;
12    }
13    std::cout << min <<
14 }
```

Running tests.....

```
min_first passed
min_last passed
min_middle passed
max_first passed
max_last passed
max_middle passed
unique passed

Tests result: passed 7 of 7 / score: 100%
```

Annotations in the image:

- Snapshot betrachten (points to the 'Really Working Version' entry)
- Abgabe (points to the 'Create Snapshot' button)
- Zurück zu (points to the 'Initial Snapshot' entry)

- mit dem Kursinhalt
 - unbedingt alle Übungen besuchen
 - dort Fragen stellen
 - und/oder Übungsleiter kontaktieren
- alle weiteren Probleme
 - Email an Dozenten (Felix Friedrich, Hermann Lehner)
- Wir helfen gerne!