

Datenstrukturen und Algorithmen

Vorlesung am D-Math (CSE) der ETH Zürich

Felix Friedrich

FS 2019

Willkommen!

Vorlesungshomepage:

<http://lec.inf.ethz.ch/DA/2019>

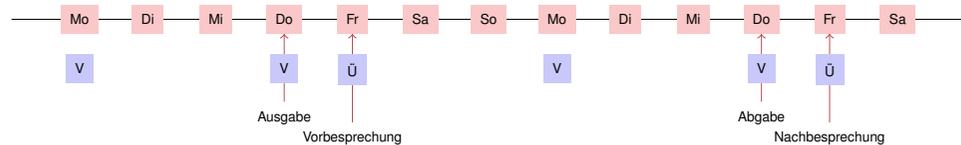
Das Team:

| | |
|-------------|---|
| Assistenten | Philippe Schlattner Jan Stratmann Robin Worreby Robin Vogtland |
| Back-Office | Aritra Dhar Pesho Ivanov |
| Dozent | Felix Friedrich |

1

2

Übungsbetrieb



- Übungsblattausgabe zur Vorlesung (online).
- Vorbesprechung in der folgenden Übung.
- Bearbeitung der Übung bis spätestens am Tag vor der nächsten Übungsstunde (23:59h).
- Nachbesprechung der Übung in der nächsten Übungsstunde. Feedback zu den Abgaben innerhalb einer Woche nach Nachbesprechung.

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, Übungsblätter).

Prüfung (120 min) ist schriftlich. Hilfsmittel: vier A4-Seiten (bzw. 2 A4-Blätter doppelseitig) entweder handgeschrieben oder mit Fontgrösse minimal 11 Punkt.

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

The C++ Programming Language, *B. Stroustrup*, 4th ed., Addison-Wesley, 2013.

The Art of Multiprocessor Programming, *M. Herlihy, N. Shavit*, Elsevier, 2012.

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 \cong 0.25.
- **Zulassung** zu speziell markierten Bonusaufgaben kann von der erfolgreichen Absolvierung anderer Übungsaufgaben abhängen.

8

Unser Angebot (Konkret)

- Insgesamt 4 Bonusaufgaben; 3/4 der Punkte reichen für 0.25 Bonuspunkte für die Prüfung
- Sie können also z.B. 3 Bonusaufgaben zu 100% lösen, oder 4 Bonusaufgaben zu je 75%, 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.

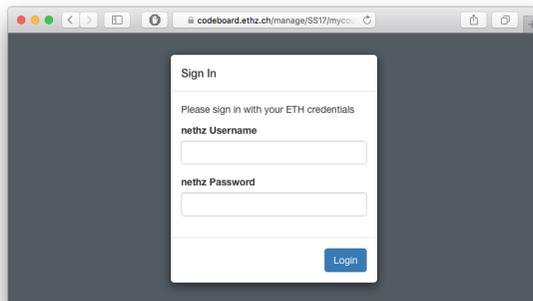
Wir prüfen das (zum Teil automatisiert) nach und behalten uns disziplinarische Massnahmen vor.

9

10

Einschreibung in Übungsgruppen - I

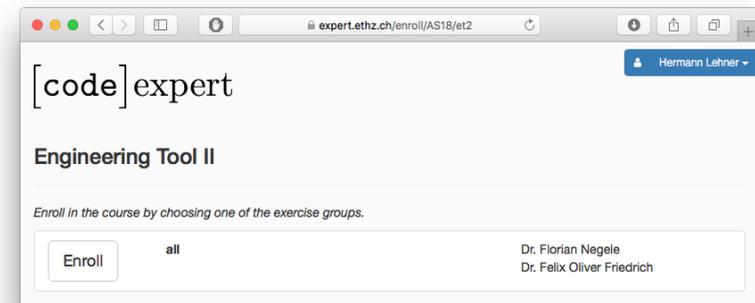
- Besuchen Sie <http://expert.ethz.ch/enroll/SS19/da>
- Loggen Sie sich mit Ihrem nethz Account ein.



11

Einschreibung in Übungsgruppen - II

Schreiben Sie sich im folgenden Dialog in eine Übungsgruppe ein.



12

Übersicht

The screenshot shows the [code]expert interface for a user named Felix Oliver Friedrich. It displays a 'Demo Course' with a table of exercises. The 'Coding Demo Exercise' section shows a submission for 'Quadratic Equations In C++' with 100% completion. Below this, there is a 'Markdown Editor Manual' section with links to 'Basic Markdown Syntax' and 'Code Blocks and Inline Code'.

| Coding Demo Exercise | Earned XP | Submissions | Handout Date | Due Date |
|----------------------------|---------------|-------------|--------------------|---------------------|
| Tasks Solutions | 1,000 / 1,000 | | 9. Sep. 2017 00:00 | 31. Dez. 2027 00:00 |
| Quadratic Equations In C++ | 1,000 ✓ | 100% | | |

| Markdown Editor Manual | Submissions | Handout Date | Due Date |
|------------------------|-------------|--------------------|--------------------|
| Tasks Solutions | | 1. Aug. 2017 00:00 | 1. Aug. 2017 00:01 |

Programmierübung

The screenshot shows a code editor with C++ code for a min-max problem. The code is annotated with letters A, B, and C. A red box labeled 'D: Beschreibung' and 'E: History' points to the right side of the editor. Another red box labeled 'A: compile', 'B: run', and 'C: test' points to the bottom left of the editor.

```
1 #include <iostream>
2
3 int main () {
4     int min; int max;
5     std::cin >> min; std::cin >> max;
6     max = min - 1;
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 }
```

13

14

Testen und Abgeben

Wo ist der Save Knopf?

The screenshot shows a code editor with C++ code for a min-max problem. A red box labeled 'Abgeben' points to the 'Submit' button. Another red box labeled 'Testen' points to the 'Run' button. The console output shows the test results for the code.

```
Running tests.....
min_first passed
min_last passed
min_middle passed
max_first failed
max_middle failed
unique passed
Tests result: passed 6 of 7 / score: 86% [ ]
```

- 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.

15

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 snapshot history panel on the right. Red arrows point from text boxes to specific UI elements:

- Snapshot betrachten**: Points to the 'Really Working Version' entry in the snapshot history.
- Abgabe**: Points to the 'Create Snapshot' button.
- Zurück zu**: Points to the 'Go back to current version' link.

```
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%
```

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