

Informatik II

Übung 1

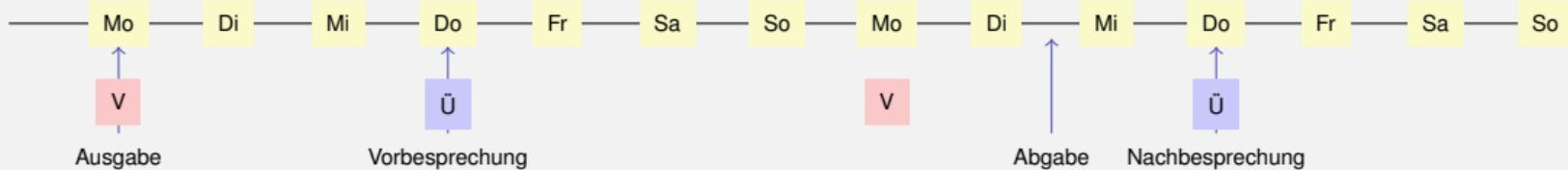
Giuseppe Accaputo, Felix Friedrich, Patrick Gruntz, Tobias Klenze, Max Rosmanek, David Sidler, Thilo Weghorn

FS 2017

Heutiges Programm

- 1 Ablauf der Übungen
- 2 Wiederholung der wichtigsten Begriffe

Ablauf



- Übungsblattausgabe zur Vorlesung (online).
- Vorbesprechung am folgenden Donnerstag. Möglichkeit, Fragen zur Vorlesung zu stellen!
- Bearbeitung der Übung bis spätestens am Dienstag (23:59) darauf.
- Nachbesprechung der Übung am Donnerstag. Feedback zu den Abgaben (auch online) innerhalb einer Woche nach Nachbesprechung.

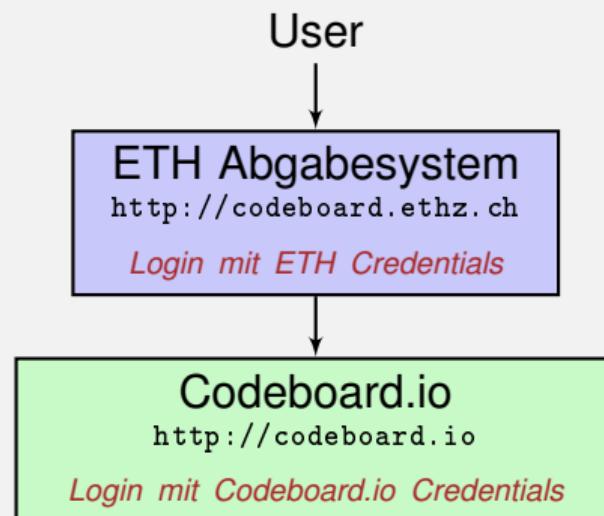
Abgabe mit codeboard

- Account erstellen
- Einloggen
- Übungen lösen und einschicken

Codeboard @ETH

Codeboard besteht aus zwei unabhängigen Systemen, die miteinander kommunizieren:

- **Das ETH Abgabesystem:**
Ermöglicht es uns, ihre Aufgaben zu bewerten
- **Die Online IDE:** Die Programmierumgebung



Codeboard

Codeboard ist eine Online-IDE: Programmieren im Browser!

- Falls vorhanden, bringen Sie ihren Laptop/Tablet/... mit in den Unterricht.
- Sie können direkt in der Vorlesung Beispiele ausprobieren, ohne dass Sie irgendwelche Tools installieren müssen.

The screenshot shows the Codeboard IDE interface. On the left, there's a file tree for 'IFMP16_S12E2a' with a single file 'averager.cpp'. The code in 'averager.cpp' is:

```
1 // Informatik - Serie 12 - Aufgabe 2a
2 // Program: averager.cpp
3 // Autor: ... (Gruppe ...)
4
5 // #include "tests.h" // remove slashes at beginning of
6 // #include <iostream>
7
8
9
10 int main () {
11     Averager avg;
12     int n;
13     std::cin >> n;
14     for (int i = 0; i < n; ++i) {
15         double input;
16         std::cin >> input;
17         avg.add_value(input);
18         std::cout << avg.average_value() << "\n";
19     }
20     std::cout << "\n";
21
22     return 0;
23 }
```

Below the code editor, a message says 'This will display the output.' To the right, a sidebar shows the task details:

Task 2a: Averager
Not submitted yet
Solve the task and hand in your solution using the green 'Submit' button.

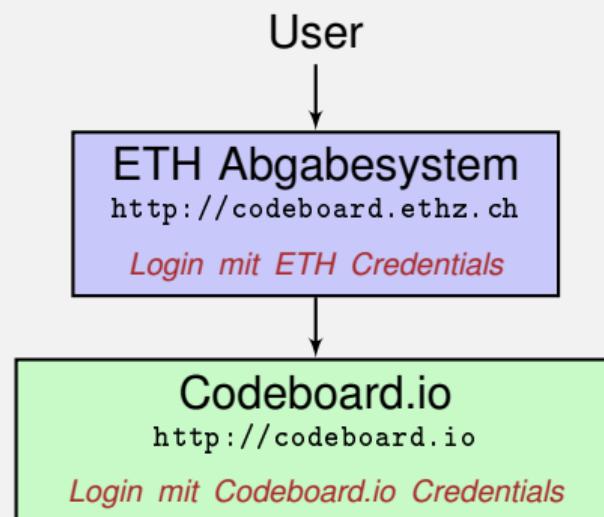
Task Description
Write a class **Averager** that computes averages of given values of type **double**. It shall offer at least the members **add_value** (which provides another value of type **double** to the class) and **average_value** (which returns the average of the values provided so far as a **double**).
(based on: Exam Summer 2012, ex.6)

At the bottom, there are buttons for 'Send' and 'codeboard.io'.

Codeboard @ETH

Codeboard besteht aus zwei unabhängigen Systemen, die miteinander kommunizieren:

- **Das ETH Abgabesystem:**
Ermöglicht es uns, ihre Aufgaben zu bewerten
- **Die Online IDE:** Die Programmierumgebung



Codeboard

Codeboard.io Registrierung

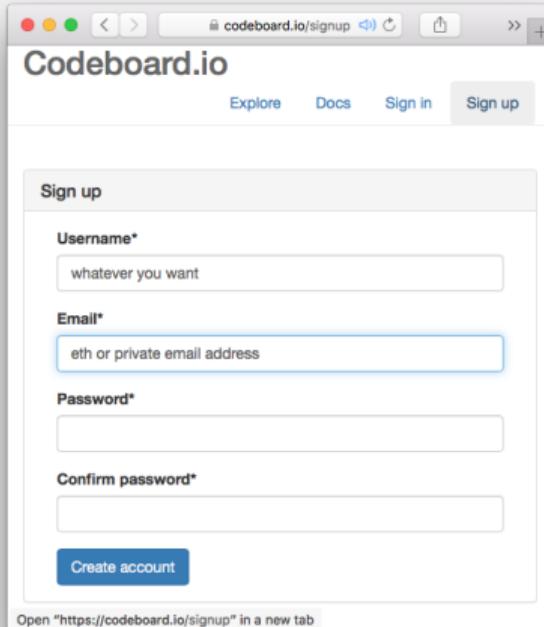
Gehen Sie auf <http://codeboard.io> und erstellen Sie dort ein Konto, bleiben Sie am besten eingeloggt.

Einschreibung in Übungsgruppen

Gehen Sie auf <http://codeboard.ethz.ch/ifbaug2> und schreiben Sie sich dort in eine Übungsgruppe ein.

Codeboard.io Registrierung

Falls Sie noch keinen **Codeboard.io** Account haben ...

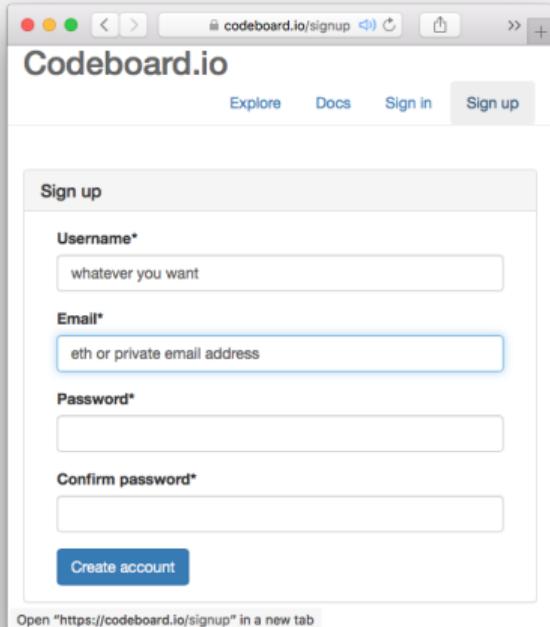


A screenshot of a web browser showing the sign-up page for Codeboard.io. The URL in the address bar is `codeboard.io/signup`. The page has a light gray header with the Codeboard.io logo and navigation links for Explore, Docs, Sign in, and Sign up. The main content area is titled "Sign up". It contains four input fields with validation asterisks: "Username*" (placeholder: "whatever you want"), "Email*" (placeholder: "eth or private email address"), "Password*", and "Confirm password*". Below these fields is a blue "Create account" button. At the bottom of the form, there is a link: "Open "`https://codeboard.io/signup`" in a new tab".

- Wir verwenden die Online IDE
Codeboard.io

Codeboard.io Registrierung

Falls Sie noch keinen **Codeboard.io** Account haben ...



The screenshot shows a web browser window with the URL "codeboard.io/signup" in the address bar. The page title is "Codeboard.io". Below the title, there are navigation links for "Explore", "Docs", "Sign in", and "Sign up". The main content is a "Sign up" form with the following fields:

- Username***: A text input field containing "whatever you want".
- Email***: A text input field containing "eth or private email address".
- Password***: An empty text input field.
- Confirm password***: An empty text input field.

At the bottom of the form is a blue "Create account" button. Below the form, a small note says "Open "<https://codeboard.io/signup>" in a new tab".

- Wir verwenden die Online IDE **Codeboard.io**
- Erstellen Sie dort einen Account, um Ihren Fortschritt abzuspeichern und später Submissions anzuschauen

Codeboard.io Registrierung

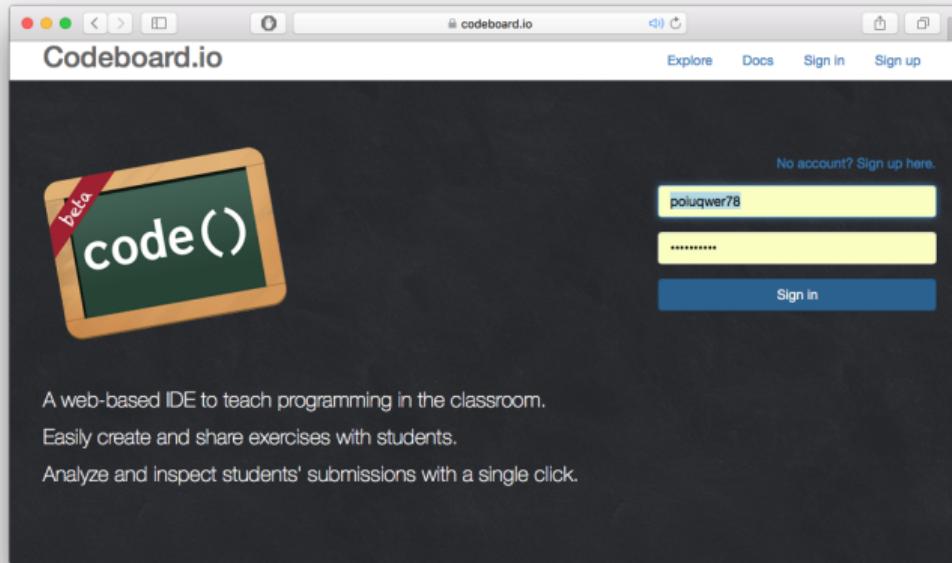
Falls Sie noch keinen **Codeboard.io** Account haben ...

The screenshot shows a web browser window with the URL "codeboard.io/signup". The page title is "Codeboard.io". Below the title, there are navigation links for "Explore", "Docs", "Sign in", and "Sign up". The main content area is titled "Sign up". It contains four input fields: "Username*" with placeholder text "whatever you want", "Email*" with placeholder text "eth or private email address", "Password*", and "Confirm password*". At the bottom is a blue "Create account" button. A small note at the bottom left says "Open \"https://codeboard.io/signup\" in a new tab".

- Wir verwenden die Online IDE **Codeboard.io**
- Erstellen Sie dort einen Account, um Ihren Fortschritt abzuspeichern und später Submissions anzuschauen
- Anmelddaten können beliebig gewählt werden! *Verwenden Sie nicht das ETH Passwort.*

Codeboard.io Login

Falls Sie schon einen Account haben, loggen Sie sich ein:



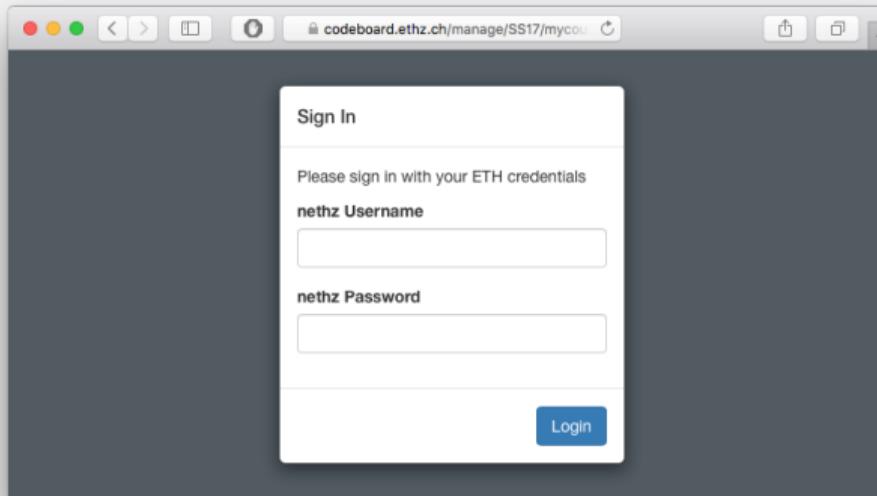
A web-based IDE to teach programming in the classroom.

Easily create and share exercises with students.

Analyze and inspect students' submissions with a single click.

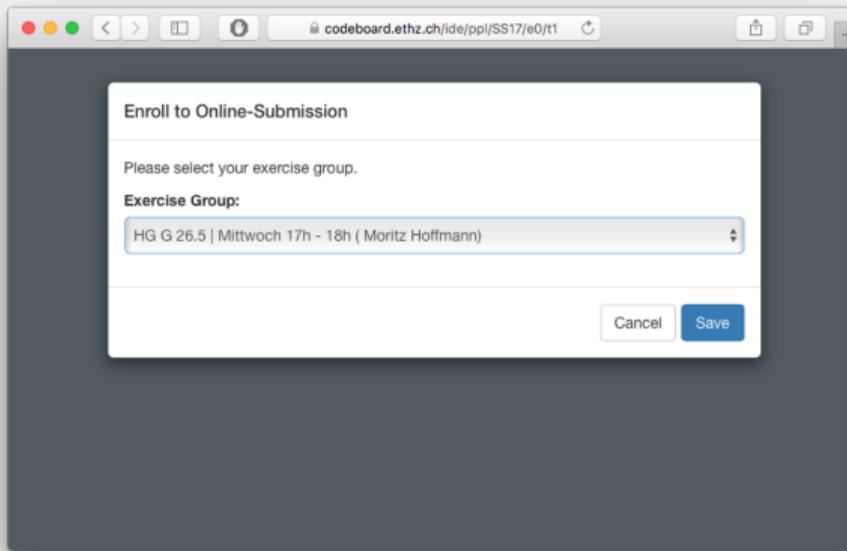
Einschreibung in Übungsgruppen - I

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



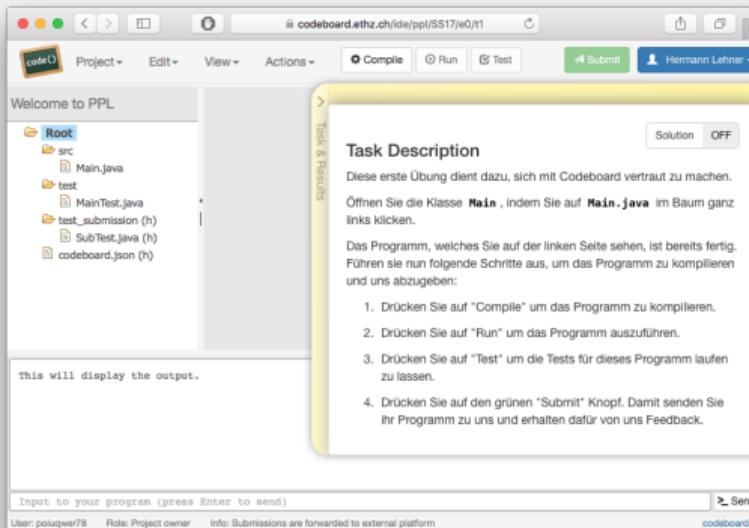
Einschreibung in Übungsgruppen - II

Schreiben Sie sich in diesem Dialog in eine Übungsgruppe ein.



Die erste Übung

Sie sind nun eingeschrieben und die erste Übung ist geladen. Folgen Sie den Anweisungen in der gelben Box. Das Übungsblatt auf der Kurshomepage enthält weitere Anweisungen und Erklärungen.



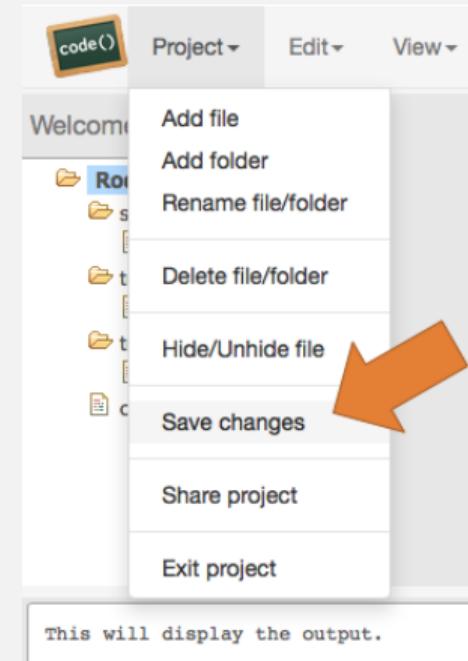
Die erste Übung - Codeboard.io Login

Falls Sie diese Nachricht sehen, klicken Sie auf [Sign in now](#) und melden Sie sich dort mit Ihrem **Codeboard.io** Account ein.



Die erste Übung - Fortschritt speichern!

Achtung! Speichern Sie ihren Fortschritt regelmässig ab. So können Sie jederzeit an einem anderen Ort weiterarbeiten.



Algorithmus

Algorithmus: wohldefinierte Berechnungsvorschrift, welche aus Eingabedaten (*input*) Ausgabedaten (*output*) berechnet.

Beispielproblem

Input : Eine Folge von n Zahlen (a_1, a_2, \dots, a_n)

Beispielproblem

Input : Eine Folge von n Zahlen (a_1, a_2, \dots, a_n)

Output : Eine Permutation $(a'_1, a'_2, \dots, a'_n)$ der Folge $(a_i)_{1 \leq i \leq n}$, so dass
 $a'_1 \leq a'_2 \leq \dots \leq a'_n$

Beispielproblem

Input : Eine Folge von n Zahlen (a_1, a_2, \dots, a_n)

Output : Eine Permutation $(a'_1, a'_2, \dots, a'_n)$ der Folge $(a_i)_{1 \leq i \leq n}$, so dass
 $a'_1 \leq a'_2 \leq \dots \leq a'_n$

Mögliche Eingaben

$(1, 7, 3), (15, 13, 12, -0.5), (1) \dots$

Beispielproblem

Input : Eine Folge von n Zahlen (a_1, a_2, \dots, a_n)

Output : Eine Permutation $(a'_1, a'_2, \dots, a'_n)$ der Folge $(a_i)_{1 \leq i \leq n}$, so dass
 $a'_1 \leq a'_2 \leq \dots \leq a'_n$

Mögliche Eingaben

$(1, 7, 3), (15, 13, 12, -0.5), (1) \dots$

Jedes Beispiel erzeugt eine *Probleminstanz*.

Euklidischer Algorithmus: Java vs. Pascal

```
public class Euclidean {  
  
    public static void main(String[] args){  
        int a = 24;  
        int b = 20;  
        while (b != 0) {  
            int h = a % b; // modulo!  
            a = b;  
            b = h;  
        }  
        System.out.println("ggt(24,20)=" + a);  
    }  
}
```

```
program Euklid;  
var a, b, h: integer;  
begin  
    a := 24;  
    b := 20;  
    while b <> 0 do begin  
        h := a mod b;  
        a := b;  
        b := h;  
    end;  
    writeln('ggt(24,20)=',a:5);  
end.
```

Mit einer Funktion (Methode)

```
public class Euclidean {  
    // PRE: a, b >= 0  
    // POST: gibt GGT(a,b) zurueck  
    static int ggt(int a, int b){  
        while (b != 0) {  
            int h = a % b;  
            a = b;  
            b = h;  
        }  
        return a;  
    }  
    public static void main(String[] args){  
        System.out.println("ggt(24,20)= " + ggt(24,20));  
    }  
}
```

Mit Eingabe

```
import java.util.Scanner;

public class Euclidean {

    static int ggt(int a, int b){ ... } // wie oben

    public static void main(String[] args){
        Scanner input = new Scanner(System.in);
        int a = input.nextInt();
        int b = input.nextInt();
        System.out.println("ggt(" + a + "," + b + ")= " + ggt(a,b));
    }
}
```

Zeichenketten (Strings)

Strings sind *Objekte* in Java.

Zuweisung eines Stringliterals:

```
String hello = "Hallo Leute";
```

Stringlänge:

```
int len = hello.length();
```

Elementzugriff¹

```
char c = hello.charAt(5);
```

Verkettung

```
String helloLong = hello + ". Alles wird gut.;"
```

¹Nur lesend. Strings sind unveränderlich

Fragen oder Anregungen?