

Informatik II

Woche 08, 23.2.2017

Giuseppe Accaputo

g@accaputo.ch

Über mich

- Giuseppe Accaputo
- 4. Semester RW/CSE Master
- E-Mail: g@accaputo.ch
- Homepage: <http://accaputo.ch>
- Homepage zur Assistenz:
<http://accaputo.ch/hilfsassistenz/informatik-2-d-baug-2017>

Fragen und Allgemeines

- Fragen stellen ist erwünscht
- Fragen stellen ist zu jedem Zeitpunkt erlaubt
- Versucht die Übungen so gut wie möglich zu lösen
 - Learning by doing!
 - **Tipp:** Musterlösungen anschauen ist erlaubt wenn man nicht weiterkommt. Versucht jedoch danach das Problem gleich nochmals ohne Musterlösungen zu lösen!

Informatik II

Übung 1

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

FS 2017

Heutiges Programm

1 Ablauf der Übungen

2 Wiederholung der wichtigsten Begriffe

Ablauf



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

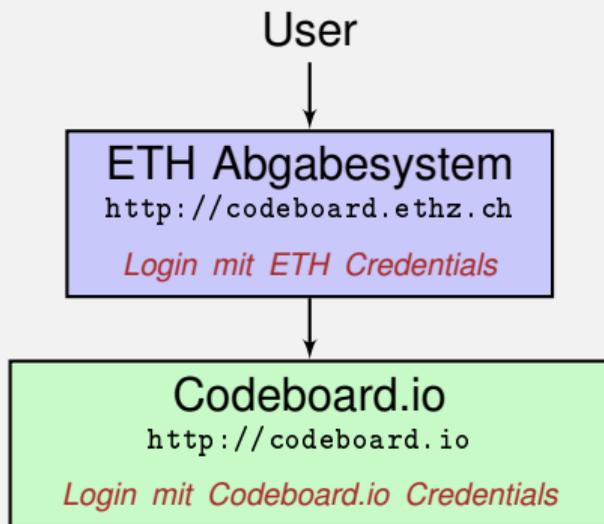
Abgabe mit codeboard

- Account erstellen
- Einloggen
- Übungen lösen und einschicken

Codeboard @ETH

Codeboard besteht aus zwei unabhängigen Systemen, die miteinander kommunizieren:

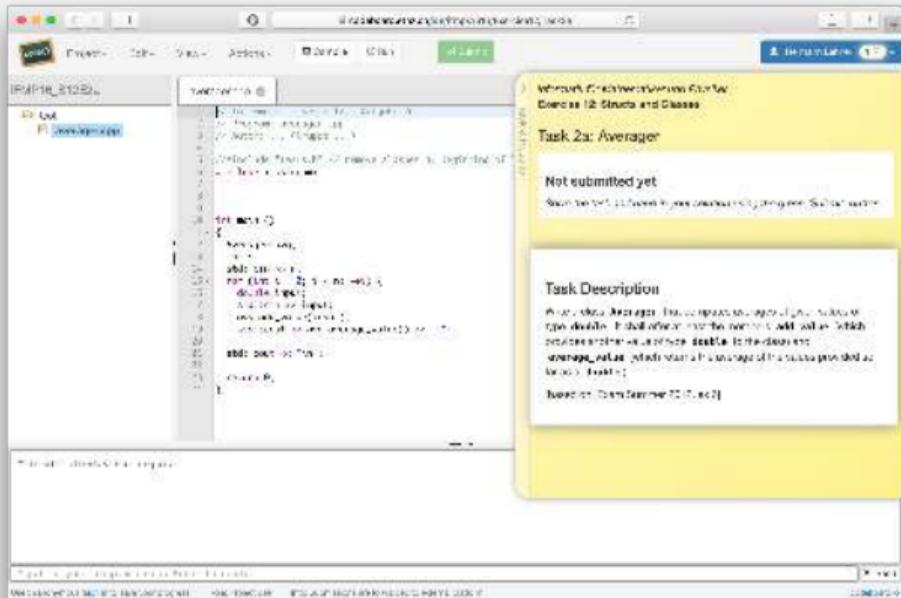
- **Das ETH Abgabesystem:**
Ermöglicht es uns, ihre Aufgaben zu bewerten
- **Die Online IDE:** Die Programmierumgebung



Codeboard

Codeboard ist eine Online-IDE: Programmieren im Browser!

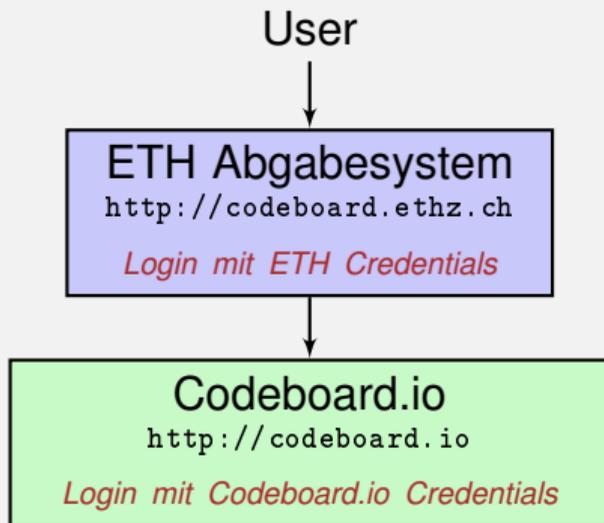
- Falls vorhanden, bringen Sie ihren Laptop/Tablet/... mit in den Unterricht.
- Sie können direkt in der Vorlesung Beispiele ausprobieren, ohne dass Sie irgendwelche Tools installieren müssen.



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 ...



The image shows a browser window with the URL `codeboard.io/signup`. The page title is "Codeboard.io" and it has 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 the placeholder text "whatever you want".
- Email***: A text input field containing the placeholder text "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 button labeled "Create account". Below the browser window, a status bar indicates: "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 image shows a browser window with the URL `codeboard.io/signup`. The page title is "Codeboard.io" and it has navigation links for "Explore", "Docs", "Sign in", and "Sign up". The main content is a "Sign up" form with the following fields:

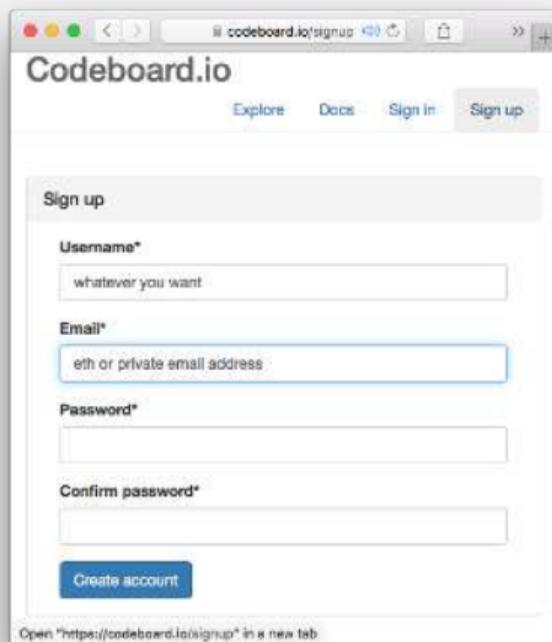
- Username***: A text input field containing the placeholder text "whatever you want".
- Email***: A text input field containing the placeholder text "eth or private email address".
- Password***: An empty text input field.
- Confirm password***: An empty text input field.

At the bottom of the form is a blue button labeled "Create account". Below the browser window, a status bar reads "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 image shows a browser window with the URL `codeboard.io/signup`. The page title is "Codeboard.io" and it has navigation links for "Explore", "Docs", "Sign in", and "Sign up". The main content is a "Sign up" form with the following fields:

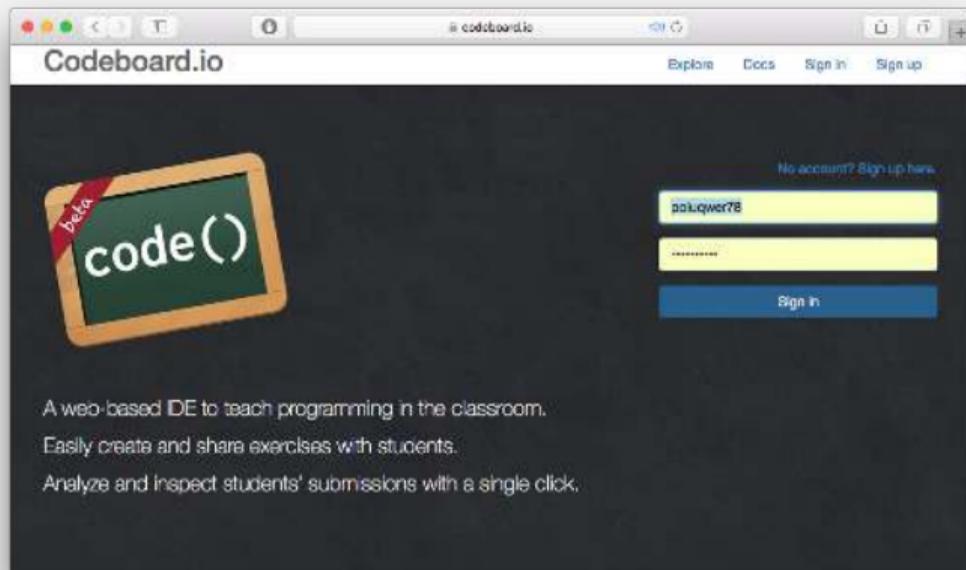
- Username***: A text input field containing the placeholder text "whatever you want".
- Email***: A text input field containing the placeholder text "eth or private email address".
- Password***: A text input field.
- Confirm password***: A text input field.

At the bottom of the form is a blue button labeled "Create account". Below the browser window, a small status bar reads "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
- Anmeldedaten können beliebig gewählt werden! *Verwenden Sie nicht das ETH Passwort.*

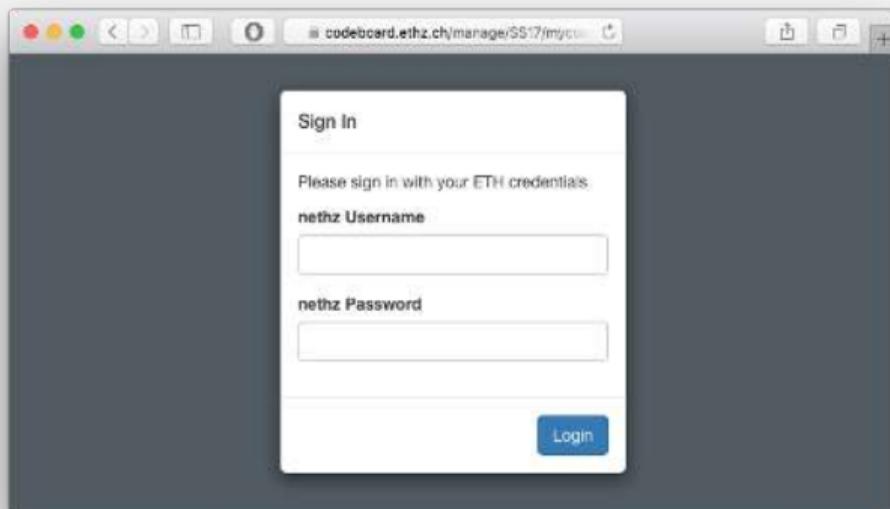
Codeboard.io Login

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



Einschreibung in Übungsgruppen - I

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

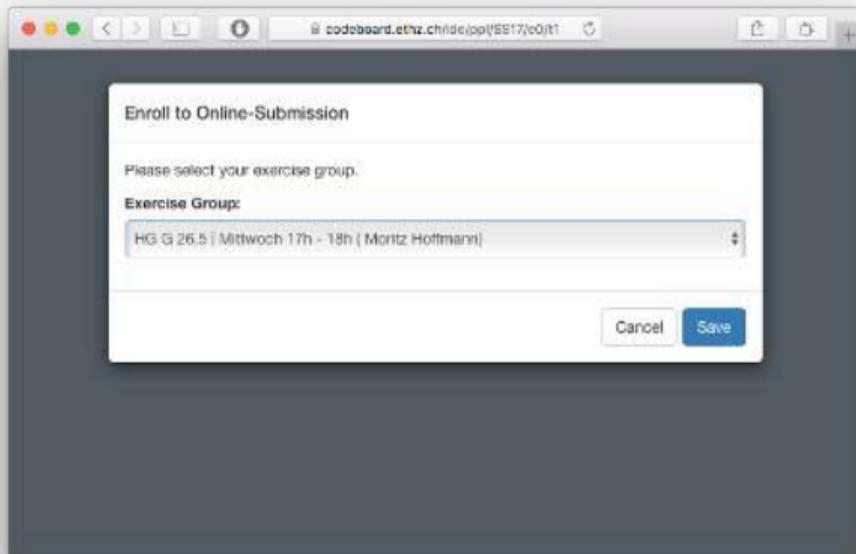


The image shows a browser window with the URL `codeboard.ethz.ch/manage/SS17/myc...`. The page content is a sign-in form with the following elements:

- Sign In** (Section Header)
- Please sign in with your ETH credentials.
- nethz Username** (Label) followed by an input field.
- nethz Password** (Label) followed by an input field.
- Login** (Button)

Einschreibung in Übungsgruppen - II

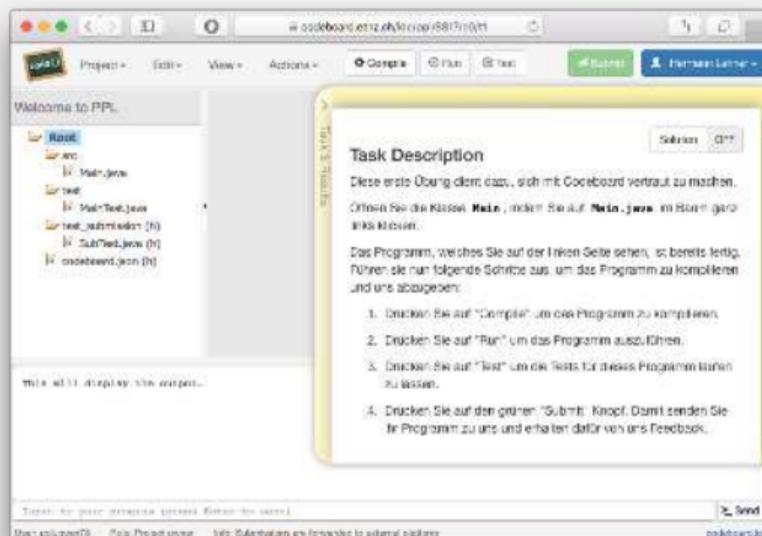
Schreiben Sie sich in diesem Dialog in eine Übungsgruppe ein.



The image shows a browser window with the URL `codeboard.ethz.ch/ide/jsp/ES17/e0/t1`. A modal dialog titled "Enroll to Online-Submission" is displayed. The dialog contains the text "Please select your exercise group." followed by a label "Exercise Group:" and a dropdown menu. The dropdown menu is currently open, showing the selected option: "HG G 26.5 | Mittwoch 17h - 18h | Moritz Hoffmann". At the bottom right of the dialog, there are two buttons: "Cancel" and "Save".

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.



The screenshot shows the CodeBoard IDE interface. On the left, a file explorer shows a project structure with files like `Main.java`, `test`, `test_ausfuehren.java`, `test_ausfuehren.java (2)`, `test_ausfuehren.java (3)`, and `codeboard.java`. The main editor area is currently empty, displaying the text `mitk.wiki.display.vhs.welcome...`. A yellow pop-up window titled "Task Description" is overlaid on the right side of the editor. The window contains the following text:

Task Description

Diese erste Übung dient dazu, sich mit Codeboard vertraut zu machen.

Öffnen Sie die Klasse `Main`, indem Sie auf `Main.java` im linken ganz links klicken.

Das Programm, welches Sie auf der linken Seite sehen, ist bereits fertig. Führen sie nun folgende Schritte aus, um das Programm zu kompilieren und uns abzugeben:

1. Drücken Sie auf "Compile" um das Programm zu kompilieren.
2. Drücken Sie auf "Run" um das Programm auszuführen.
3. Drücken Sie auf "Test" um die Tests für dieses Programm laufen zu lassen.
4. Drücken Sie auf den grünen "Submit" Knopf. Damit senden Sie Ihr Programm zu uns und erhalten dafür von uns Feedback...

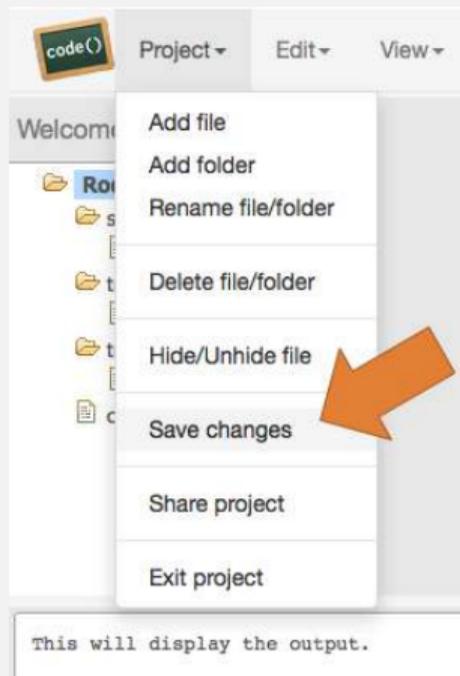
Die erste Übung - Codeboard.io Login

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?