

Informatik II

Woche 11, 16.03.2017

Giuseppe Accaputo

g@accaputo.ch

Programm heute

- Nachbesprechung Übung 3
 - Einige Tipps
 - Besprechung Aufgabe 3.1 und 3.2
 - Live-Programmierung: Aufgabe 3.3 (Artificial Words)
- Repetition: Rekursion, Divide and Conquer
- Vorbesprechung Übung 4

Nachbesprechung: Übung 3

Tipps

- Beachtet bitte die Ausgabe der Tests auf der Konsole!
 - Zusätzlich: Keine unnötigen Ausgaben in Code platzieren wenn Code validiert werden soll
- Bei for-Schleifen: Deklariert die Zählvariable von neu (wenn möglich)
- Aufgabe 3.1: `nextInt()` statt `nextDouble()`
- Konsolen-Input mittels CTRL+C und CTRL+V wiederverwenden (z.B. Matrix bei Aufgabe 3.2)
 - Time-Out wurde entfernt

Aufgaben 3.1 und 3.2

- Musterlösung kurz zusammen anschauen
- Matrix-Multiplikation: $A \in \mathbb{R}^{n \times m}$, $B \in \mathbb{R}^{m \times p}$, $C \in \mathbb{R}^{n \times p}$

$$\mathbf{C} = \mathbf{A} \cdot \mathbf{B}$$

$$C_{ij} = \sum_{k=1}^m A_{ik} \cdot B_{kj}$$

Aufgabe 3.3

- Live-Programmierung

Repetition: **Rekursion, Divide and Conquer**

Prüfung 08.2015 Aufgabe 1b

- Was wird auf der Konsole ausgegeben?

```
static double recursive(double x){  
    if (x > 1)  
        return ((int)x) * recursive(x-1);  
    return x;  
}
```

```
System.out.println(recursive(2.5));
```

```
System.out.println(recursive(5.2));
```


Vorbesprechung: Übung 4

Aufgabe 4.1: Recursive Function Analysis

- *Preconditions*: Bedingungen an die Parameter einer Funktion, die erfüllt sein müssen bevor die Funktion ausgeführt wurde, um einen korrekten Ablauf zu garantieren
- *Postconditions*: Bedingung die wahr ist, nachdem die Funktion ausgeführt wurde

Aufgabe 4.2: Money

- Aufbau einer Rekursiven Funktion:
 - Abbruchbedingung
 - Schritt in Richtung Erfüllung der Abbruchbedingung
 - Rekursiver Aufruf
- Wie lautet die Abbruchbedingung für die Funktion

```
int partition(int amount, int[] billsAndCoins, int start)
```

Fragen oder Anregungen?