



Informatik I

Übungsstunde 7

Herbst 2020

Hausaufgaben

■ Fragen?

Einführung

sum(n) berechnet die Summe der Integer von **1** bis **n**. Wie würdest du die Summe aller Integer von **1** bis **n+1** berechnen?

Einführung

sum(n) berechnet die Summe der Integer von **1** bis **n**. Wie würdest du die Summe aller Integer von **1** bis **n+1** berechnen?

```
public static int sum(int n) {  
    int x = 0;  
    for(int i = 1; i <= n; i++) {  
        x = x + i;  
    }  
    return x;  
}
```

Einführung

sum(n) berechnet die Summe der Integer von **1** bis **n**. Wie würdest du die Summe aller Integer von **1** bis **n+1** berechnen?

```
public static int sum(int n) {  
    if(n == 1) return 1; sum(1)  
    else return n + sum(n - 1);  
}
```

Einführung

fact(n) berechnet das Produkt der Integer von **1** bis **n**. Wie würdest du das Produkt aller Integer von **1** bis **n+1** berechnen?

Einführung

$$n! = n \cdot (n-1)! = \dots$$

fact(n) berechnet das Produkt der Integer von **1** bis **n**. Wie würdest du das Produkt aller Integer von **1** bis **n+1** berechnen?

```
public static int fact(int n) {  
    if (fact == 1) return 1; ←  
    else n * fact(n - 1);  
    return x;  
}
```

Aufgabe 1: Repeating Input

Die folgende Methode wiederholt den Input des Benutzers (durch "end" beendet):

```
public static void repeatInput() {  
    while(true) {  
        String input = In.readLine();  
        if (input.equals("end")) {  
            return;  
        }  
        Out.println(input);  
    }  
}
```

```
> hello  
hello  
> you  
you  
> end
```

■ Schreibe die Methode neu auf **rekursive Weise**.

Aufgabe 2: Reversing the Input

Verändere die rekursive Methode so, dass alle Eingaben in **umgekehrter Reihenfolge** ausgegeben werden nachdem "end" eingegeben wurde.

```
> hello  
> you  
> end  
you  
hello
```

Iterative Potenz

Die folgende Methode **poweri** gibt die n -te positive Potenz x^n einer Zahl $x \in \mathbb{R}$ mit $n > 0$ zurück:

// pre: $n > 0$

// post: return x^n

```
public static double poweri(double x, int n){
```

```
    double result = 1;
```

```
    while (n > 0){
```

```
        result *= x;
```

```
        --n;
```

```
    return result;
```

```
}
```

$$x^n = \underbrace{x \cdot x \cdot \dots \cdot x}_x$$

Aufgabe 3: Rekursive Potenz

- Implementiere eine **rekursive** Methode **powerrr**, die die n -te positive Potenz x^n einer Zahl $x \in \mathbb{R}$ mit $n > 0$ zurückgibt.

Benutze keine Schleifen (keine **while**, **for** oder **do-while**).

Aufgabe 4: Optimierte Rekursive Potenz

$$x^n$$

- Implementiere eine rekursive Methode **powers**, die die n -te positive Potenz einer Zahl $x \in \mathbb{R}$ mit $n > 0$ zurückgibt **unter Verwendung einer geringen Anzahl von Multiplikationen**. Benutze keine Schleifen (keine **while**, **for** oder **do-while**).
- Optional: Verbessere die Methode so, dass sie zusätzlich negative Potenzen berechnet werden können.

~~$$x^4 = x \cdot x \cdot x \cdot x$$~~

$$x^2 = x \cdot x$$

$$x^4 = x^2 \cdot x^2$$