

PSE - Vorkurs Tag 2

Linus, Philipp, Tillmann, Tobias

FIUS - Fachgruppe Informatik Universität Stuttgart

07.09.2025



Recap Tag 1



Recap Tag 1

```
1  int alter = 20;    // Deklaration + Initialisierung
2  int zahl;         // Deklaration
3  zahl = 42;        // Initialisierung
```



Recap Tag 1

```
1 int alter = 20; // Deklaration + Initialisierung
2 int zahl; // Deklaration
3 zahl = 42; // Initialisierung
```

Operator	Bedeutung	Beispiele
+	Addition / String-Verkettung	3+2, "ä"+"b"
-	Subtraktion	7-5
*	Multiplikation	3*4
/	Division	10/2
%	Modulo	5%2

Recap Tag 1



Recap Tag 1

```
1 System.out.print("Hallo");  
2 System.out.println("Welt");  
3 System.out.println("Zahl: " + 42);
```



Recap Tag 1

```
1 System.out.print("Hallo");  
2 System.out.println("Welt");  
3 System.out.println("Zahl: " + 42);
```

```
1 import java.util.Scanner;  
2  
3 Scanner sc = new Scanner(System.in);  
4 int x = sc.nextInt();  
5 int y = sc.nextInt();  
6 System.out.println(x + y);
```



Boolean - **Wahr** oder **Falsch**



Boolean - Wahr oder Falsch

- ▶ Datentyp mit zwei Werten: `true` oder `false`



Boolean - Wahr oder Falsch

- ▶ Datentyp mit zwei Werten: `true` oder `false`
- ▶ Wahrheitswerte speichern und verarbeiten



Boolean - Wahr oder Falsch

- ▶ Datentyp mit zwei Werten: `true` oder `false`
- ▶ Wahrheitswerte speichern und verarbeiten
- ▶ z.B:

```
1  boolean heuteDienstag = true;
```

Boolean Operatoren



Boolean Operatoren

Wenn a und b Werte sind, prüfen diese Operatoren Beziehungen zwischen ihnen:

- ▶ `heuteDienstag == esRegnet` Wahr, wenn `heuteDienstag` gleich `esRegnet` ist

```
1 boolean result = (heuteDienstag == esRegnet); // result true,  
   ↪ wenn heuteDienstag gleich esRegnet
```

 **Achtung:**für Strings: `equals()` benutzen!

Boolean Operatoren



Boolean Operatoren

- ▶ `heuteDienstag != esRegnet` Wahr, wenn `heuteDienstag` ungleich `esRegnet` ist

```
1 boolean result = (heuteDienstag != esRegnet); // result true,  
   ↪ wenn heuteDienstag ungleich esRegnet
```

Boolean Operatoren

- ▶ `heuteDienstag != esRegnet` Wahr, wenn `heuteDienstag` ungleich `esRegnet` ist

```
1 boolean result = (heuteDienstag != esRegnet); // result true,  
  ↪ wenn heuteDienstag ungleich esRegnet
```

- ▶ `preis < geld` Wahr, wenn `geld` kleiner als `preis` ist

```
1 boolean result = (preis < geld); // result true, wenn preis  
  ↪ kleiner als geld
```

Boolean Operatoren

- ▶ `heuteDienstag != esRegnet` Wahr, wenn `heuteDienstag` ungleich `esRegnet` ist

```
1 boolean result = (heuteDienstag != esRegnet); // result true,  
  ↪ wenn heuteDienstag ungleich esRegnet
```

- ▶ `preis < geld` Wahr, wenn `geld` kleiner als `preis` ist

```
1 boolean result = (preis < geld); // result true, wenn preis  
  ↪ kleiner als geld
```

- ▶ analog bei `preis > geld`, `preis <= geld`, `preis >= geld`

Boolean Operatoren



Boolean Operatoren

▶ $\neg a \rightarrow$ NICHT a



Boolean Operatoren

▶ `!a` → NICHT a

▶ `a && b` → a UND b



Boolean Operatoren

▶ `!a` → NICHT a

▶ `a && b` → a UND b

a	b	a && b
true	true	true
true	false	false
false	true	false
false	false	false

Boolean Operatoren

▶ $!a \rightarrow$ NICHT a

▶ $a \ \&\& \ b \rightarrow$ a UND b

a	b	a && b
true	true	true
true	false	false
false	true	false
false	false	false

▶ $a \ || \ b \rightarrow$ a ODER b

Boolean Operatoren

▶ `!a` → NICHT a

▶ `a && b` → a UND b

a	b	a && b
true	true	true
true	false	false
false	true	false
false	false	false

▶ `a || b` → a ODER b

a	b	a b
true	true	true
true	false	true
false	true	true
false	false	false

Boolean Operatoren

▶ `!a` → NICHT a

▶ `a && b` → a UND b

a	b	a && b
true	true	true
true	false	false
false	true	false
false	false	false

▶ `a || b` → a ODER b

a	b	a b
true	true	true
true	false	true
false	true	true
false	false	false

Klammern priorisieren:

```
1 true || false && false
2 // -> true
3 (true || false) && false
4 // -> false
```



if-Verzweigung



if-Verzweigung

- ▶ Ausführung nur wenn Bedingung `true`



if-Verzweigung

- ▶ Ausführung nur wenn Bedingung true

```
1  if (Bedingung) {  
2    // Code bei true  
3  }
```

if-Verzweigung

- ▶ Ausführung nur wenn Bedingung true

```
1  if (Bedingung) {  
2    // Code bei true  
3  }
```

- ▶ Beispiel:

```
1  boolean heuteDienstag = true;  
2  if (heuteDienstag) {  
3    System.out.println("Crazyyy heute ist Dienstag");  
4  }
```

if-Verzweigung

- ▶ Ausführung nur wenn Bedingung true

```
1  if (Bedingung) {  
2    // Code bei true  
3  }
```

- ▶ Beispiel:

```
1  boolean heuteDienstag = true;  
2  if (heuteDienstag) {  
3    System.out.println("Crazyyy heute ist Dienstag");  
4  }
```

Crazyyy heute ist Dienstag

if-Verzweigung



if-Verzweigung

- ▶ geht genauso mit `int` etc.



if-Verzweigung

- ▶ geht genauso mit int etc.
- ▶ Beispiel:

```
1  int Lieblingszahl = 42;  
2  if (Lieblingszahl == 42) {  
3      System.out.println("Du bist ein Highperformer!");  
4  }
```

if-Verzweigung

- ▶ geht genauso mit int etc.
- ▶ Beispiel:

```
1  int Lieblingszahl = 42;  
2  if (Lieblingszahl == 42) {  
3      System.out.println("Du bist ein Highperformer!");  
4  }
```

Du bist ein Highperformer!

if-else



if-else

- ▶ erweitert die if Anweisung



if-else

- ▶ erweitert die `if` Anweisung
- ▶ "wenn `if` Bedingung nicht erfüllt dann mach folgendes..."



if-else

- ▶ erweitert die if Anweisung
- ▶ "wenn if Bedingung nicht erfüllt dann mach folgendes..."

```
1  if (Bedingung) {  
2    // Code bei Bedingung true  
3  } else {  
4    // Code bei Bedingung false  
5  }
```

if-else



if-else

► Beispiel:

```
1  boolean heuteDonnerstag = false;
2  if (heuteDonnerstag) {
3      System.out.println("endlich Wochenende");
4  } else {
5      System.out.println(":( bestimmt ist bald wieder Donnerstag");
6  }
```

if-else

► Beispiel:

```
1  boolean heuteDonnerstag = false;
2  if (heuteDonnerstag) {
3      System.out.println("endlich Wochenende");
4  } else {
5      System.out.println(":( bestimmt ist bald wieder Donnerstag");
6  }
```

```
:( bestimmt ist bald wieder Donnerstag
```

else if



else if

- ▶ else if prüft mehrere Bedingungen



else if

- ▶ else if prüft mehrere Bedingungen
- ▶ Beispiel:

```
1  if (note == 1) {  
2      System.out.println("Sehr gut");  
3  } else if (note == 2) {  
4      System.out.println("Gut");  
5  } else if (note == 3) {  
6      System.out.println("Befriedigend");  
7  } else {  
8      System.out.println("Ausreichend oder schlechter");  
9  }
```

Check Yourself!



Frage 1: Was ergibt...

```
1 false || true
```



Frage 1: Was ergibt...

```
1 false || true
```

```
true
```



Frage 2: Was ergibt...

```
1 true || false && false
```



Frage 2: Was ergibt...

```
1 true || false && false
```

true



Frage 3: Was wird ausgegeben ?

```
1  boolean a = true;  
2  boolean b = false;  
3  boolean result = a && b;  
4  System.out.println(result);
```

Frage 3: Was wird ausgegeben ?

```
1  boolean a = true;  
2  boolean b = false;  
3  boolean result = a && b;  
4  System.out.println(result);
```

false

Frage 4: Was ergibt true?



Frage 4: Was ergibt true?

```
1 (true && (false || true)) || false
2 // und auch
3 (true || false) && (true && true)
```



Frage 5: Was wird ausgegeben?

```
1 boolean sonne = true;
2 boolean regen = false;
3 if (sonne && !regen) {
4     System.out.println("Perfektes Wetter!");
5 } else {
6     System.out.println("Bleib lieber drinnen.");
7 }
```

Frage 5: Was wird ausgegeben?

```
1 boolean sonne = true;
2 boolean regen = false;
3 if (sonne && !regen) {
4     System.out.println("Perfektes Wetter!");
5 } else {
6     System.out.println("Bleib lieber drinnen.");
7 }
```

Perfektes Wetter!



Frage 6: Was wird ausgegeben?

```
1 int note = 3;
2 if (note == 1) {
3     System.out.println("Sehr gut");
4 } else if (note == 2) {
5     System.out.println("Gut");
6 } else {
7     System.out.println("Der Rest");
8 }
```



Frage 6: Was wird ausgegeben?

```
1  int note = 3;
2  if (note == 1) {
3      System.out.println("Sehr gut");
4  } else if (note == 2) {
5      System.out.println("Gut");
6  } else {
7      System.out.println("Der Rest");
8  }
```

Der Rest



Frage 7: Was wird ausgegeben?

```
1  boolean a = false;
2  boolean b = true;
3  if (a = b) {
4      System.out.println("Wow!");
5  } else {
6      System.out.println("Nope!");
7  }
```



Frage 7: Was wird ausgegeben?

```
1  boolean a = false;
2  boolean b = true;
3  if (a = b) {
4      System.out.println("Wow!");
5  } else {
6      System.out.println("Nope!");
7  }
```

Wow!

$a = b$ ist eine Zuweisung, kein Vergleich!



Frage 8: Was wird ausgegeben?

```
1  boolean a = true;
2  boolean b = false;
3  boolean c = true;
4  if (a && (b || c)) {
5      System.out.println("1");
6  }
7  if ((a && b) || (c && !b)){
8      System.out.println("2");
9  }
10 if ((a || b) && !(c && b)) {
11     System.out.println("3");
12 }
```

Frage 8: Was wird ausgegeben?

```
1  boolean a = true;
2  boolean b = false;
3  boolean c = true;
4  if (a && (b || c)) {
5      System.out.println("1");
6  }
7  if ((a && b) || (c && !b)){
8      System.out.println("2");
9  }
10 if ((a || b) && !(c && b)) {
11     System.out.println("3");
12 }
```

- 1
- 2
- 3



while-Schleife



while-Schleife

- ▶ Wiederholt Anweisungen, solange eine Bedingung true ist

```
1 while (Bedingung) { /* Code */ }
```

while-Schleife



while-Schleife

```
1 Scanner scanner = new Scanner(System.in);
2 String eingabe = "";
3
4 while (!eingabe.equals("ok")) {
5     System.out.println("Bitte 'ok' eingeben:");
6     eingabe = scanner.nextLine();
7 }
```

while-Schleife

```
1 Scanner scanner = new Scanner(System.in);
2 String eingabe = "";
3
4 while (!eingabe.equals("ok")) {
5     System.out.println("Bitte 'ok' eingeben:");
6     eingabe = scanner.nextLine();
7 }
```

Bitte 'ok' eingeben:

...

Bitte 'ok' eingeben:

(Benutzer tippt 'ok') → Schleife endet



break



break

- ▶ Mit `break` kann man eine Schleife vorzeitig beenden



break

- ▶ Mit break kann man eine Schleife vorzeitig beenden

```
1 Scanner scanner = new Scanner(System.in);
2
3 String passwort = "diegrillung";
4 String abbruchBedingung = "abbruch";
5 String momentaneEingabe = "";
6
7 while (!momentaneEingabe.equals(passwort)) {
8     momentaneEingabe = scanner.nextLine();
9
10    if (momentaneEingabe.equals(abbruchBedingung)) {
11        break;
12    }
13 }
14 System.out.println("I'm in");
```

Code together



Was ist eine `for`-Schleife?



Was ist eine `for`-Schleife?

- ▶ Wiederholt Anweisungen eine festgelegte Anzahl von Malen

Was ist eine for-Schleife?

- ▶ Wiederholt Anweisungen eine festgelegte Anzahl von Malen

```
1  for (Start; Bedingung; Schritt) {  
2      // Schleifenrumpf  
3  }
```

Was ist eine for-Schleife?

- ▶ Wiederholt Anweisungen eine festgelegte Anzahl von Malen

```
1  for (Start; Bedingung; Schritt) {  
2      // Schleifenrumpf  
3  }
```

- ▶ Ablauf:

Was ist eine for-Schleife?

- ▶ Wiederholt Anweisungen eine festgelegte Anzahl von Malen

```
1  for (Start; Bedingung; Schritt) {  
2      // Schleifenrumpf  
3  }
```

- ▶ Ablauf:
 1. Wo starten wir? (z.B. bei 0)

Was ist eine for-Schleife?

- ▶ Wiederholt Anweisungen eine festgelegte Anzahl von Malen

```
1  for (Start; Bedingung; Schritt) {  
2      // Schleifenrumpf  
3  }
```

- ▶ Ablauf:
 1. Wo starten wir? (z.B. bei 0)
 2. Bedingung noch erfüllt?



Was ist eine for-Schleife?

- ▶ Wiederholt Anweisungen eine festgelegte Anzahl von Malen

```
1  for (Start; Bedingung; Schritt) {  
2      // Schleifenrumpf  
3  }
```

- ▶ Ablauf:
 1. Wo starten wir? (z.B. bei 0)
 2. Bedingung noch erfüllt?
 3. Falls ja, Code im Block ausführen



Was ist eine for-Schleife?

- ▶ Wiederholt Anweisungen eine festgelegte Anzahl von Malen

```
1  for (Start; Bedingung; Schritt) {  
2      // Schleifenrumpf  
3  }
```

- ▶ Ablauf:
 1. Wo starten wir? (z.B. bei 0)
 2. Bedingung noch erfüllt?
 3. Falls ja, Code im Block ausführen
 4. Zähler ändern i.d.R. einen hochzählen



for-Schleife

Somit wird aus...

```
1  int momentaneWurst = 4;
2  int letzteWurst = 8;
3  while (momentaneWurst <= letzteWurst) {
4      System.out.println("Schmeiß Wurst Nr." + momentaneWurst + " auf
   ↪   den Grill");
5      momentaneWurst++;
6  }
```



for-Schleife

Somit wird aus...

```
1  int momentaneWurst = 4;
2  int letzteWurst = 8;
3  while (momentaneWurst <= letzteWurst) {
4      System.out.println("Schmeiß Wurst Nr." + momentaneWurst + " auf
   ↪   den Grill");
5      momentaneWurst++;
6  }
```

ganz simpel...

```
1  for (int i = 4; i <= 8; i++) {
2      System.out.println("Schmeiß Wurst Nr." + i + " auf den Grill");
3  }
```



- ▶ Folien und Aufgaben:

<https://fius.de/index.php/pse-vk-folien/>

Wenn ihr Fragen habt, sagt Bescheid!

