PSE – Vorkurs Tag 2

Tobias, Philipp, Linus, Tillmann

FIUS - Fachgruppe Informatik Universität Stuttgart

10. Juli 2025



am Anfang immer Vortages recap?	3
alles einheitlich mit minted block oder auch inline?	4





Recap Tag 1

am Anfang immer Vortages recap?





Boolean - Wahr oder Falsch

- Datentyp mit zwei Werten: true oder false
- Wahrheitswerte speichern und verarbeiten
- Z.B: heuteDienstag = true;

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

alles einheitlich mit minted block oder auch inline?



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

▶ a == b Wahr, wenn a gleich b ist

```
1 boolean result = (a == b); // result true, wenn a gleich b
```



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

▶ a == b Wahr, wenn a gleich b ist

```
boolean result = (a == b); // result true, wenn a gleich b
```

a != b Wahr, wenn a ungleich b ist

```
1 boolean result = (a != b); // result true, wenn a ungleich b
```



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

▶ a == b Wahr, wenn a gleich b ist

```
1 boolean result = (a == b); // result true, wenn a gleich b
```

▶ a != b Wahr, wenn a ungleich b ist

```
1 boolean result = (a != b); // result true, wenn a ungleich b
```

a < b Wahr, wenn a kleiner als b ist</p>

```
1 boolean result = (a < b); // result true, wenn a kleiner als b
```





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

▶ a == b Wahr, wenn a gleich b ist

```
boolean result = (a == b); // result true, wenn a gleich b
```

▶ a != b Wahr, wenn a ungleich b ist

```
1 boolean result = (a != b); // result true, wenn a ungleich b
```

a < b Wahr, wenn a kleiner als b ist</p>

```
1 boolean result = (a < b); // result true, wenn a kleiner als b
```

▶ analog bei a > b, a <= b, a >= b





ightharpoonup !a ightharpoonup nicht a





- ightharpoonup !a \rightarrow nicht a
- ightharpoonup a && b ightharpoonup a UND b





- ightharpoonup !a \rightarrow nicht a
- ightharpoonup a && b ightharpoonup a UND b

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



- ightharpoonup !a \rightarrow nicht a
- ightharpoonup a && b ightharpoonup a UND b

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

ightharpoonup a | | b ightharpoonup a ODER b



- ightharpoonup !a ightharpoonup nicht a
- ightharpoonup a && b ightharpoonup a UND b

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

ightharpoonup a | | b \rightarrow a ODER b

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



- ightharpoonup !a ightharpoonup nicht a
- ightharpoonup a && b ightharpoonup a UND b

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

▶ $a \mid \mid b \rightarrow a \cup DER b$

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

Klammern priorisieren:

- ► true || false && false
- ► (true || false) && false





- ightharpoonup !a \rightarrow nicht a
- ightharpoonup a && b ightharpoonup a UND b

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

ightharpoonup a | | b \rightarrow a ODER b

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

Klammern priorisieren:

- ▶ true || false && false

 ⇒ true
- ► (true || false) && false





- ightharpoonup !a \rightarrow nicht a
- ightharpoonup a && b ightharpoonup a UND b

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

▶ $a \mid \mid b \rightarrow a \cup DER b$

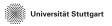
a	b	a b
true	true	true
true	false	true
false	true	true
false	false	false

Klammern priorisieren:

- ▶ true || false && false

 ⇒ true
- ► (true || false) && false

 ⇒ false





- ► Ausführung nur wenn Bedingnung true
- Syntax:

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



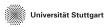
- Ausführung nur wenn Bedingnung true
- Syntax:

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

Beispiel:

```
boolean heuteDienstag = true;
if (heuteDienstag) {
    System.out.println("Crazyyy heute ist Dinestag");
}
```

Crazyyy heute ist Dinestag





▶ geht genauso mit int etc.



- geht genauso mit int etc.
- ► Beispiel:

```
int lieblingszahl = 42;
if (lieblingszahl == 42) {
    System.out.println("Du bist ein Highperformer!");
}
```

```
Du bist ein Highperformer!
```



if-else

- erweitert die if Anweisung
- "wenn if Bedingnung nicht erfüllt dann mach folgendes..."
- Syntax:

```
if (Bedingung) {
   // Code bei Bedingnung true
} else {
   // Code bei Bedingung false
}
```

if-else

- erweitert die if Anweisung
- "wenn if Bedingnung nicht erfüllt dann mach folgendes..."
- Syntax:

```
if (Bedingung) {
   // Code bei Bedingnung true
} else {
   // Code bei Bedingung false
}
```

► Beispiel:

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



if-else

► Beispiel:

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

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





else if

- lack else if prüft mehrere Bedingungen
- ► Beispiel:

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



while-Schleife

- ▶ Wiederholt Anweisungen, solange eine Bedingung true ist
- ► Syntax: while (Bedingung) { /* Code */ }

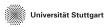


while-Schleife

- Wiederholt Anweisungen, solange eine Bedingung true ist
- ► Syntax: while (Bedingung) { /* Code */ }
- Beispiel:

```
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 ök") → Schleife endet
```





break

Mit break kann man eine Schleife vorzeitig beenden

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





WURST TOGETHER





Was ist eine for-Schleife?

- ► Wiederholt Anweisungen eine festgelegte Anzahl von Malen
- Syntax:

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



Somit wird aus...

ganz simpel...

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



