

Inhalt

1. Verzweigungen
2. Vergleichsoperatoren
3. IF Anweisung
4. Boolsche Algebra

Verzweigungen

- im Deutschen: Konditionssätze
 - Wenn das Wetter schön ist, gehe ich spazieren, sonst schlafe ich.
 - \Rightarrow Wenn das Wetter schlecht ist, gehe ich nicht spazieren und schlafe

```
WENN wetter IST schön  
DANN spazieren  
SONST schlafen
```

Verzweigungen in Java

- Abfrage, welcher Codeblock ausgeführt werden soll, wenn condition erfüllt wird

```
if (condition){  
    // then  
}else{  
    // else  
}
```

```
if (wetter == schön){  
    // spazieren  
}else{  
    // sonst  
}
```

If ohne else

- Ein if kann auch ohne else-Zweig existieren
- Der Codeblock `// spazieren` wird übersprungen, wenn das Wetter **NICHT** schön ist.

```
if (wetter == schön){  
    // spazieren  
}
```

Vergleichsoperatoren

Vergleiche

- gleich `==`
- ungleich `!=`
- größer `>`
- größer gleich `>=`
- kleiner `<`
- kleiner gleich `<=`

Vergleichsoperatoren

Vergleiche

a	b	a == b	a != b	a > b	a >= b	a < b	a <= b
1	1						
1	2						
2	1						
true	true						
fale	true						

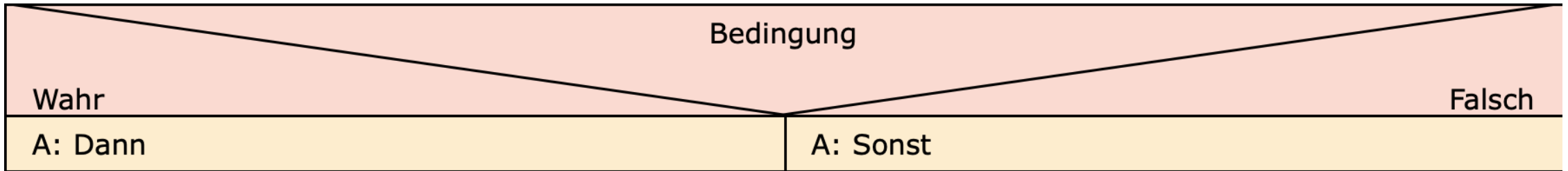
Vergleichsoperatoren

Vergleiche (Auflösung)

a	b	a == b	a != b	a > b	a >= b	a < b	a <= b
1	1	true	false	false	true	false	true
1	2	false	true	false	false	true	true
2	1	false	true	true	true	false	false
true	true	true	false				
false	true	false	true				

IF Syntax

Struktogramm (<https://struktog.openpatch.org>)



Java Code

```
if(Bedingung){  
    // Anweisung  
}else{  
    // Anweisung  
}
```


IF Anweisungen

```
int a = 1;  
int b = 1;  
  
if(a == b) // Bedingung  
{  
    // Dann – Anweisung  
}else{  
    // Sonst – Anweisung  
}
```

IF Anweisungen Wenn, Dann, sonst wenn, sonst

- Beliebiges Aneinanderreihen von Abfragen
- Sobald ein Zweig `true`, werden andere Zweige ignoriert

```
char typ = 's';  
  
if(typ == 's'){  
    System.out.println("Person ist Schüler");  
}  
else if(gender == 'p'){  
    System.out.println("Person ist in Pension");  
}  
else {  
    System.out.println("Person ist weder Schüler/in, noch Pensionist/in");  
}
```

Boolsche Algebra

- Bedingungen können verknüpft werden
 - UND `&&`
 - ODER `||`
 - XOR (exklusives oder) `^`
- Beispiele
 - Person ist Schüler und älter als 18
 - Person ist Lehrer oder Schüler
 - Person geht zur Schule und ist Schüler oder Lehrer

Boolsche Algebra Übungen

$a \geq b \ \&\& \ a \leq b$

a	b	$a \geq b$	$a \leq b$	$a \geq b \ \&\& \ a \leq b$
3	4			
3	3			

Boolsche Algebra Übungen

$a \geq b \ \&\& \ a \leq b$

a	b	$a \geq b$	$a \leq b$	$a \geq b \ \&\& \ a \leq b$
3	4	false	true	false
3	3	true	true	true

weitere Übungen

1. $A \Rightarrow B$

2. $(A \ \&\& \ !B) \ || \ (!A \ \&\& \ B)$

3. $A \wedge B$

4. $(A \ || \ !A) \ \&\& \ B \Rightarrow B$

5. $(A \ \&\& \ B) \Rightarrow (!A \ || \ B)$

<https://www.matheretter.de/wiki/implikation>

Verknüpfungen

- UND `&&`
 - `if (Bedingung1 && Bedingung2)`
 - Beide Bedingungen müssen true sein
- ODER `||`
 - `if (Bedingung1 || Bedingung2)`
 - Eines der beiden muss true ergeben

```
if(x>10 && x<20)
{
    System.out.println("x ist zwischen 10 und 20");
}
else
{ System.out.println("x nicht zwischen 10 und 20"); }
```

Vergleiche von Strings

- String ist ein komplexer Datentyp. `==` vergleicht Adressen, nicht Werte
- Methode `.equals(...)` vergleicht String-Werte

```
String str1 = "Hallo";  
String str2 = "Hallo";  
  
if(str1.equals(str2)){  
    // Werte von str1 und str2 sind gleich  
}else{  
    // Wenn str1 und str2 unterschiedliche Werte haben  
}
```


Verzweigungen aneinanderhängen

- Es können in einem `if` auch unterschiedliche Werte abgefragt werden.
- Hierzo werden `if` s immer mit `else if(bedingung)` verknüpft

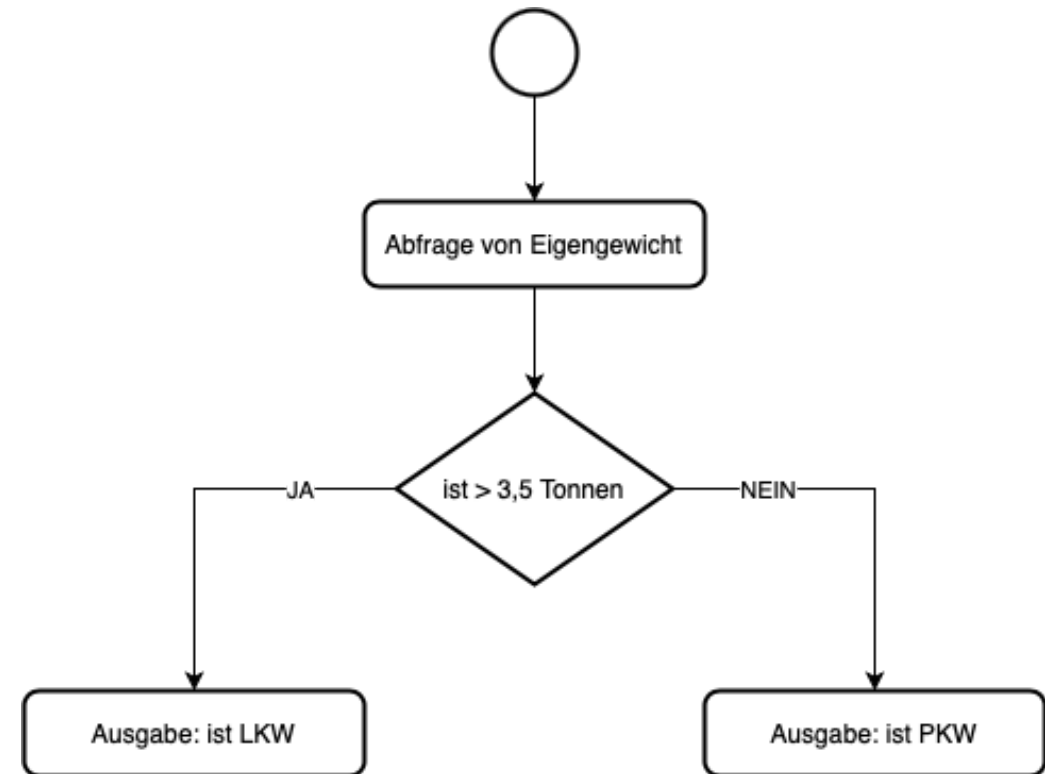
```
String str1 = "Hallo";

if(str1.equals("Hallo")){
    // str1 ist Hallo
}else if(str1.equals("Servus")){
    // str1 ist Servus
}else if(str1.equals("Hi")){
    // str1 ist Hi
}else{
    // Default Zweig.
}
```

Unified Modeling Language (UML)

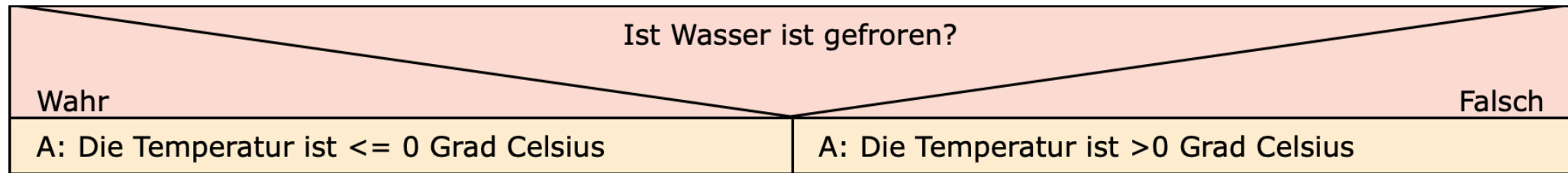
- Beschreibung von Softwareprozessen
- Grafische Annotation
- **Aktivitätsdiagramm** beschreibt Abläufe

```
if( weight > 3500 ) {  
    System.out.println("ist LKW");  
}else{  
    System.out.println("ist PKW");  
}
```



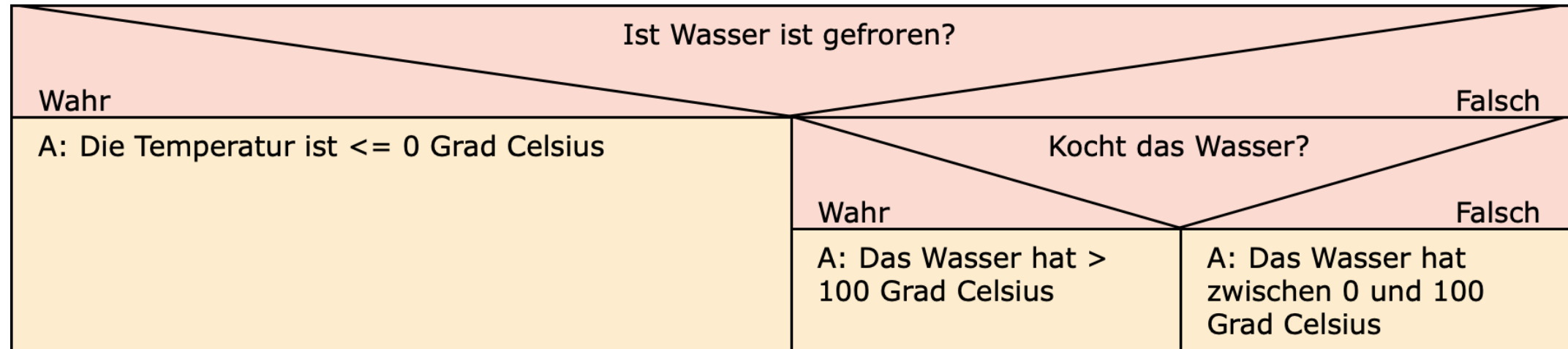
Struktogramme

- Einfachen Darstellung von Abläufen
- Verzweigungen, Fallunterscheidungen, Schleifen
- Online-Editor: <https://struktog.openpatch.org>













Struktogramme

- weitere Fallunterscheidung



struktog

- <https://struktog.openpatch.org>
- Elemente:

 Eingabe-Feld	 Ausgabe-Feld	 Anweisung
 Zählergesteuerte Schleife	 Kopfgesteuerte Schleife	 Fußgesteuerte Schleife
 Verzweigung	 Fallunterscheidung	 Try-Catch-Block
	 Funktionsblock	