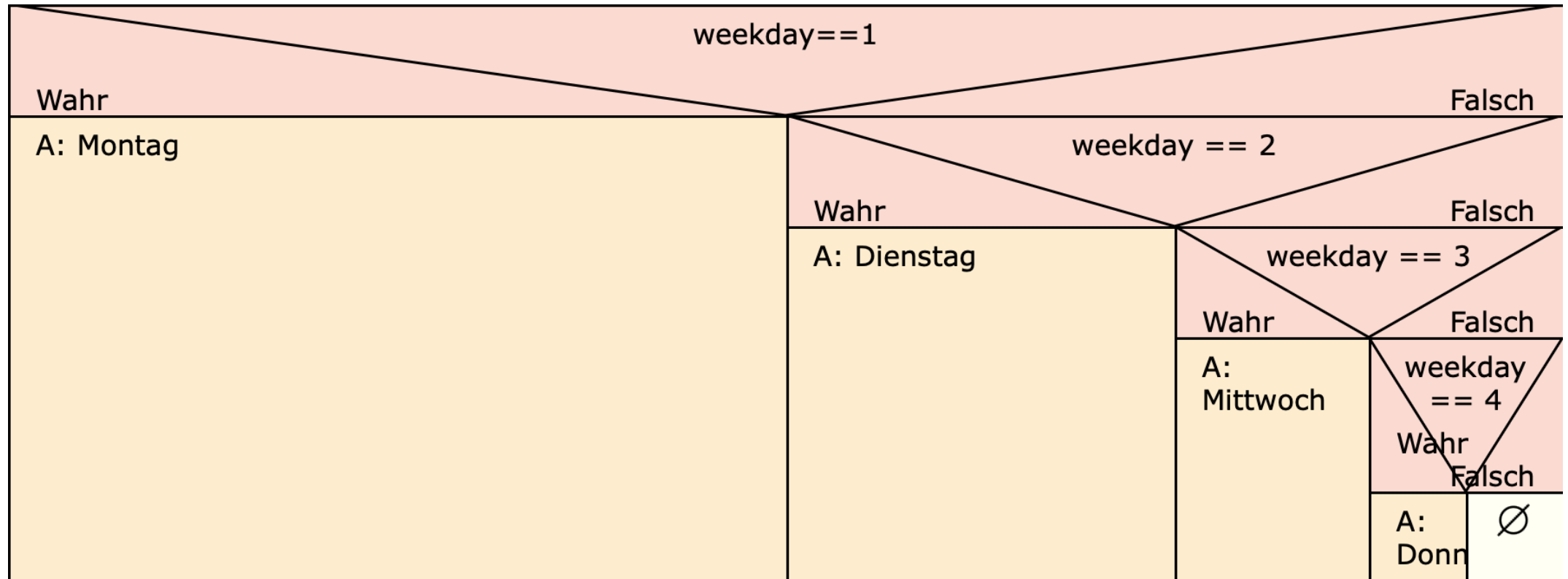


Inhalt

1. Motivation
2. Fallunterscheidung
3. Verkürzte Version
4. Hinweise

Unterscheidung von Wochentage

- mit if



Unterscheidung von Wochentage

- mit switch

Variable							
1	2	3	4	5	6	7	Sonst
A: Montag	A: Dienstag	A: Mittwoch	A: Donnerstag	A: Freitag	A: Samstag	A: Sonntag	A: kein Wochentag

Fallunterscheidung (switch)

- Kurzschreibweise für langes if (mit `==` Vergleichen)

```
switch (variable)
{
    case Bedingung1:
        Anweisung(en);
        break;           // break = beendet Anweisungsblock

    case Bedingung2:
        Anweisung(en);
        break;

    default:             // Wenn kein anderer Fall (ähnlich else)
        Anweisung(en);
}
```

Gültige Werte

- primitive Datentypen bis int (byte, short, char, int)
- String (*nur in neueren Versionen*)
- Funktioniert nur mit == nicht mit <=, <, > oder >=!

Beispiel

```
int num = 0;
switch (num)
{
    case 0:
        System.out.println("zahl ist 0"); // Hier
        break;
    case 1:
        System.out.println("zahl ist 1");
        break;
    default:
        System.out.println("nicht 0 oder 1");
}
```

Beispiel

```
int num = 3;
switch (num)
{
    case 0:
        System.out.println("zahl ist 0");
        break;
    case 1:
        System.out.println("zahl ist 1");
        break;
    default:
        System.out.println("nicht 0 oder 1"); // Hier
}
```

Unterschiede if-else und switch (1/3)

Bedingungstyp

- if-else: Kann auf komplexe Bedingungen überprüfen und unterstützt Ausdrücke, die nicht auf konstanten Werten basieren.
- Vergleicht eine Variable mit konstanten Werten.

Lesbarkeit

- if-else: Kann für Bedingungen verwendet werden, die schwer zu kategorisieren sind und verschiedene Bedingungen erfordern.
- switch: Eignet sich für Bedingungen mit vielen klaren und diskreten Fällen.

Unterschiede if-else und switch (2/3)

Mehrere Bedingungen

- if-else: Erlaubt die Überprüfung mehrerer Bedingungen in einer Abfolge von if- und else if-Anweisungen.
- switch: Kann nur eine Variable auf verschiedene konstante Werte überprüfen.

Default-Fall

- if-else: Benötigt keinen speziellen "Standardfall" und kann mit einer zusätzlichen else-Anweisung umgehen.
- switch: Verwendet default für den Fall, dass kein case-Ausdruck übereinstimmt.

Unterschiede if-else und switch (3/3)

Lesbarkeit

- if-else: Kann in komplexen logischen Bedingungen verwendet werden, was die Lesbarkeit beeinträchtigen kann.
- switch: Bietet eine klarere Struktur, wenn viele Bedingungen geprüft werden müssen.

Kombination

- gleiche Funktionalität bei mehreren Werten kann zusammengefasst werden.

```
int zahl = 18;
switch (zahl)
{
    case 0:
    case 1:
    case 2:
        System.out.println("0 bis 2");
        break;
    case 10:
        System.out.println("zahl ist 10");
        break;
    default:
        System.out.println("nicht 0,1,2,10");
}
```

Kombination

- in neuen Java-Versionen:

```
int zahl = 18;
switch (zahl)
{
    case 0, 1, 2:
        System.out.println("0 bis 2");
        break;
    case 10:
        System.out.println("zahl ist 10");
        break;
    default:
        System.out.println("nicht 0,1,2,10");
}
```

Switch mit Variablenzuweisung (lang)

```
int num = -1;
switch (weekday.toLowerCase()) {
    case "montag":
        num = 1;
        break;
    case "dienstag":
        num = 2;
        break;
    case "mittwoch":
        num = 3;
        break;
    case "donnerstag":
        num = 4;
        break;
    case "freitag":
        num = 5;
        break;
    case "samstag":
        System.out.println("WOCHENENDE");
        num = 6;
        break;
    case "sonntag":
        System.out.println("WOCHENENDE");
        num = 7;
        break;
    default:
        num = -1;
}
```

Switch mit Variablenzuweisung (kurz)

- Default-Zweig muss existieren
- Strichpunkt am Ende
- `weekday.toLowerCase()` verändert Variable weekday NICHT -> nur für switch

```
int numWeekday = switch (weekday.toLowerCase()) {  
    case "montag" -> 1;  
    case "dienstag" -> 2;  
    case "mittwoch" -> 3;  
    case "donnerstag" -> 4;  
    case "freitag" -> 5;  
    case "samstag" -> 6;  
    case "sonntag" -> 7;  
    default -> -1;  
};
```

Hinweise (1/2)

Wann verwendet man switch?

Die switch Anweisung eignet sich gut, wenn Sie eine Variable auf verschiedene Werte überprüfen und entsprechende Aktionen ausführen müssen.

Vergleich nur mit konstanten Werten

In Java können Sie nur konstante Werte wie Ganzzahlen, Zeichen und Aufzählungen in case-Anweisungen verwenden.

break nicht vergessen

Vergessen Sie nicht, `break` nach jedem `case` -Block hinzuzufügen, um einen Fall nicht in den nächsten Fall zu durchlaufen.

Hinweise (2/2)

Verwendung von default

Die `default`-Anweisung wird ausgeführt, wenn kein `case`-Ausdruck übereinstimmt.

switch vs. if-else

Bei komplexeren Bedingungen oder Bedingungen, die nicht auf konstanten Werten basieren, ist `if-else` oft flexibler. (zB. `temperatur < 0`)

Gültiger Datentyp

Der Ausdruck in der switch-Anweisung muss vom Datentyp her kompatibel mit den `case`-Ausdrücken sein.