**Verzweigungen**

**Verzweigungen** dienen dazu,

bestimmte Pro­grammteile nur dann auszuführen, wenn vorgegebene Bedingungen eintreten, die erst zur Laufzeit bekannt werden.

**if-Anweisung**

*Syntax*

**if** (bedingung)

anweisung;

Funktionsweise

1. Zuerst wird durch die **if-Anweisung** der Ausdruck bedingung ausgewertet.
2. Die Anweisung anweisung wird nur dann ausgeführt, wenn die Bedingung zutrifft[[1]](#footnote-1). Trifft der Ausdruck bedingung dagegen nicht zu[[2]](#footnote-2), wird die Anweisung nicht ausgeführt, sondern mit der ersten Anweisung nach der if-Anweisung fortgefahren.

*Beispiel für eine if-Anweisung:*

if (zahl == 5) {

System.out.println(„Die Zahl ist 5.“);

}

**if-else-Anweisung**

*Syntax*

**if** (bedingung)

anweisungl;

**else**

anweisung2;

Funktionsweise

1. Zuerst wird auch hier die **if-Anweisung** der Ausdruck bedingung ausgewertet.
2. Falls bedingung zutrifft, wird anweisung1 ausgeführt, anderenfalls anweisung2.

🡪 Es wird immer eine der beiden Anweisungen ausgeführt.

*Beispiel für eine if-else-Anweisung:*

if (zahl == 5) {

System.out.println(„Die Zahl ist 5.“);

}

else {

System.out.println(„Die Zahl ist nicht 5.“);

}

* Anstatt einer Anweisung sind auch Sequenzen (Blöcke) von Anweisungen möglich. Hier sind immer geschweifte Klammern notwendig.

**switch-Anweisung**

*Syntax*

**switch** (fall) {

**case** constant1:

anweisung1;

**case** constant2:

anweisung2;

**case** constant3:

anweisung3;

…

**default**:

}

Funktionsweise

Die **switch-Anweisung** ist eine **Mehrfachverzweigung**, bei der eine **fallweise Unterscheidung** stattfindet.

1. Zunächst wird der Ausdruck fall ausgewertet.
2. Danach wird die Sprungmarke (case) angesprungen, deren Konstante mit dem Ergebnis des Ausdrucks übereinstimmt (z.B. constant3).
3. Das optionale default-Label wird dann angesprungen, wenn keine passende Sprungmarke gefunden wird. Gibt es kein default-Label und auch keine passende Sprung­marke, wird keine der Anweisungen innerhalb der switch-Anweisung ausge­führt.
4. Jede Konstante eines case-Labels darf nur einmal auftauchen.
5. Das default-Label darf ebenfalls nur einmal verwendet werden (Es muss aber nicht verwendet werden.).
6. Nachdem ein case- oder default-Label angesprungen wurde, werden **alle folgenden Anweisungen** ausgeführt. Wenn dies nicht erwünscht ist, muss der **Kontrollfluss mithilfe einer break-Anweisung unterbro­chen werden**.

* Jedes **break** innerhalb einer switch-Anweisung führt dazu, dass **zum Ende der switch-Anweisung verzweigt** wird.
* Innerhalb von switch-Anweisungen sind **keine Schleifen** möglich!
* Der Ausdruck (hier fall) vom Typ byte, short, char oder int oder String[[3]](#footnote-3) sein!

Die Konstante und der Ausdruck müssen dabei zuweisungs­kompatibel sein, d.h. sie müssen den gleichen Typ haben.

*Beispiel für eine switch-Anweisung:*

int zahl = 5;

switch (zahl) {

case 1:

System.out.println(„Die Zahl ist 1.“);

break;

case 2:

System.out.println(„Die Zahl ist 2.“);

break;

case 3:

System.out.println(„Die Zahl ist 3.“);

break;

default:

System.out.println(„Die Zahl ist außerhalb des Bereiches von 1 bis 5.“);

}

1. Das Ergebnis des Ausdrucks ist dann wahr (true). [↑](#footnote-ref-1)
2. Das Ergebnis ist falsch (false). [↑](#footnote-ref-2)
3. Die Verwendung eines String als Ausdruck ist seit JDK 1.5 möglich. [↑](#footnote-ref-3)