

# Kapitel 7

# Verzweigungen/Bedingungen

# Verzweigungen

- Verzweigungen werden eingesetzt, wenn eine Befehlsfolge nur unter bestimmten Bedingungen ausgeführt werden soll.
- In C/C++ wird dies durch das Konstrukt *if/else* erreicht.
- Sinngemäß bedeutet das *if/else* Konstrukt: Wenn die Bedingung zutrifft, mache das (Befehlszeilen im *if*-Block). Ansonsten mache etwas anderes (Befehlszeilen im *else*-Block).

# if/else -Anweisung

- Mit der if/else -Anweisung kann eine von zwei Anweisungen abhängig von einer Bedingung ausgewählt werden.

```
if (x > 10)
{
    cout << "x ist größer als zehn" << endl;
}
else
{
    cout << "x ist kleiner oder gleich zehn" << endl;
}
```

# if/else -Anweisung

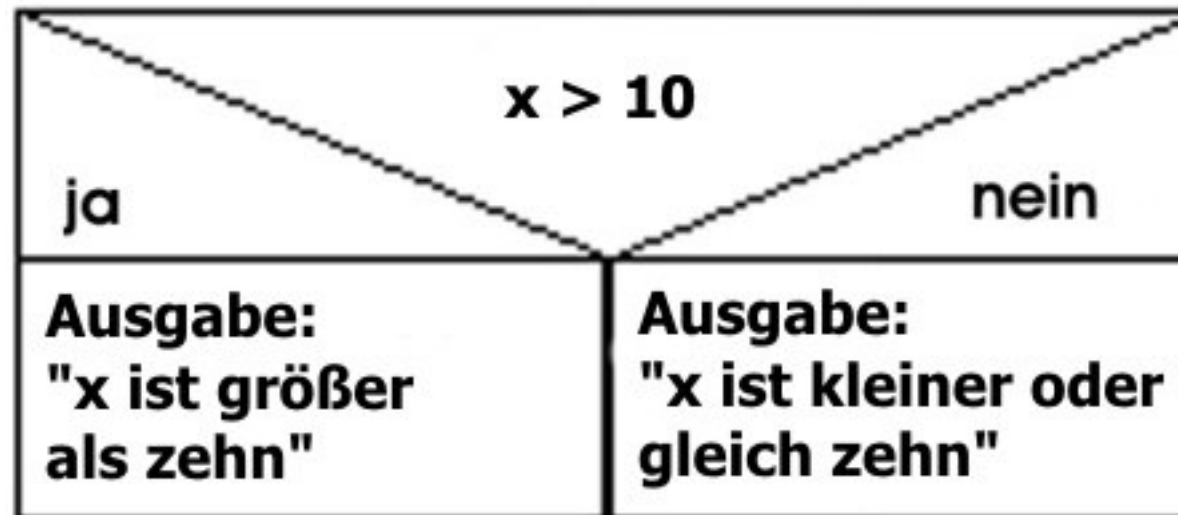
- Nach dem Schlüsselwort *if* folgt ein logischer Ausdruck in Klammern.
- Liefert der logische Ausdruck den Wert *true* (ungleich 0), dann wird die *if*-Anweisung ausgeführt. Im anderen Fall wird die *else*-Anweisung ausgeführt.

# if/else -Anweisung

- Wenn eine Anweisungsfolge nur aus einer Anweisung besteht, können die Blockklammern weggelassen werden.

```
if (x > 10)
    cout << "x ist größer als zehn" << endl;
else
    cout << "x ist kleiner oder gleich zehn" << endl;
```

# if/else -Anweisung



# if -Anweisung

- Der *else* -Zweig kann auch entfallen.

```
if (nenner == 0)  
    cout << "Division durch Null!" << endl;
```

Alternative Schreibweise:

```
if (nenner == 0) cout << "Division durch Null!" << endl;
```

# else/if -Anweisung

- Mit der else/if -Anweisung können eine oder mehrere weitere Bedingungen eingebunden werden.

```
if (x >= -1 && x <= 1)
    cout << "x liegt im Intervall [-1,1]" << endl;
else if (x > 10)
    cout << "x ist größer als zehn" << endl;
else if (x < 5)
    cout << "x ist kleiner als 5" << endl;
else
    cout << "x hat den Wert:" << x << endl;
```



# else/if -Anweisung

|                            |                      |                     |                    |
|----------------------------|----------------------|---------------------|--------------------|
|                            |                      |                     | $x = ?$            |
| $[-1,1]$                   | $> 10$               | $< 5$               | sonst              |
| Ausgabe:<br>$x \in [-1,1]$ | Ausgabe:<br>$x > 10$ | Ausgabe:<br>$x < 5$ | Ausgabe:<br>Wert ? |

# switch/case -Anweisung

- Mit der switch/case –Anweisung ist es möglich, mehr als zwei Fälle zu unterscheiden.
- Eine Folge von else/if –Anweisungen lässt sich auch durch eine switch/case –Anweisung realisieren.
- Nach jedem *case* folgt ein konstanter Ausdruck.
- Man kann einen sogenannten default-Fall einfügen, der abgearbeitet wird, wenn kein anderer Fall zutrifft.

# switch/case -Anweisung

```
cin >> zahl
switch (zahl)
{
    case 4:  cout << "vier";
            break;
    case 5:  cout << "fuenf";
            break;
    default: cout << "keine Zahl";
}

```

# switch/case -Anweisung

| <b>Wert 1</b>                | <b>Wert 2</b>                | <b>Wert 3</b>                | <b>Wert 4</b>                | <b>Variable</b><br><b>sonst</b>             |
|------------------------------|------------------------------|------------------------------|------------------------------|---|
| <b>Anweisungsblock<br/>1</b> | <b>Anweisungsblock<br/>2</b> | <b>Anweisungsblock<br/>3</b> | <b>Anweisungsblock<br/>n</b> | <b>Alternativ-<br/>block<br/>(optional)</b> |