C++ Teil 2.4-2.5

2.4. Ausdrücke und Anweisungen

Konstanten, Variablen und Funktionsergebnisse lasse sich in C++ zu *Ausdrücken* verbinden. Dabei gelten folgende Regeln:

- Jeder (Teil-)Ausdruck liefert einen Wert, der weiterverwendet werden kann.
- Die Auswertung richtet sich nach der Hierarchie der verwendeten Operatoren. Zuerst werden die Operatoren ausgewertet, die eine höhere Hierarchie besitzen (z.B. `*` vor `+`).
- Werden Klammer gesetzt, so werden diese zuerst ausgewertet. Bei einer Schachtelung von Klammern, kommen immer zuerst die inneren Klammern dran.

Beispiele:

a) `x = 1 + 2 * 3;` b) `x = y = z = 0;` c) `x = (y = (z = 0));`

Aus einen Ausdruck macht man eine *Anweisung*, indem man ihn mit einem Semikolon abschließt. Alle Anweisungen enden in C++ mit einem Semikolon.

Zu a): Hier wird 2 mal 3 gerechnet und dann 1 dazu addiert. Das Ergebnis 7 wird der Variablen x zugewiesen. Zu b) und c): Auch Zuweisungen haben einen Ergebniswert, der weiter verwendet werden kann. Beide Anweisungen sind gleichbedeutend, da die Abarbeitungsreihenfolge von rechts nach links geht.

2.5. Operatoren

Mit den Operatoren lassen sich in C++ Konstanten und Variablen zu Ausdrücken verbinden. Man unterscheidet verschiedene Arten von Operatoren. Einige dieser Arten sind:

- Zuweisungsoperatoren,
- unäre Operatoren (Inkrement und Dekrement),
- arithmetische Operatoren,
- logische Operatoren.

Zuweisungsoperatoren

Beispiel: `x = a + b;`
Der Zuweisungsoperator „=“, weist der Variablen `x` den Wert `a + b` zu.

Weitere Zuweisungsoperatoren kombiniert mit den arithmetischen Operatoren (siehe unten) findet man in der rechts stehenden Tabelle.

OP	Bedeutung	Bsp. (x = 6)	Ergebnis
`=`	Zuweisung	`x = 2`	`x = 2`
`+=`	Addition und Zuweisung	`x += 2`	`x = 8`
`-=`	Subtraktion und Zuweisung	`x -= 2`	`x = 4`
`*=`	Multiplikation und Zuweisung	`x *= 2`	`x = 12`
`/=`	Division und Zuweisung	`x /= 2`	`x = 3`
`%=`	Modulo und Zuweisung	`x %= 2`	`x = 0`

Beispiel:

Es kommt oft vor, dass man zu dem Wert einer Variablen eine Zahl addieren muss. Dieses Problem kann man auf verschiedene Art und Weise lösen. Eine Möglichkeit ist folgende:

```
C = C + 1;
```

Zu dem Wert in der Variablen `C` wird `1` dazugezählt und der neue Wert dann wieder der Variablen zugewiesen. Mit Hilfe der Kombination aus Zuweisungs- und arithmetischem Operator geht es kürzer:

```
C += 1;
```

Inkrement und Dekrement

Sehr häufig kommt es in C++ vor, dass man den Wert einer Variablen um 1 erhöhen (Inkrementierung) oder vermindern (Dekrementierung) muss. C++ kennt dafür zwei eigene Operatoren zur Inkrementierung (`++`) und Dekrementierung (`--`) von Variablen. Das obige Beispiel kann also noch knapper geschrieben werden:

`C++;` (Jetzt wird auch klar, woher C++ seinen Namen hat.)

Arithmetische Operatoren

Es gibt fünf arithmetische Operatoren.

Addition, Subtraktion und Multiplikation funktionieren so, wie man es gewöhnt ist. Bei der Division können „Probleme“ auftauchen, wenn man mit Integer Zahlen operiert, dann ist z.B. `22 / 4 = 5` und nicht `5.5`, wie zu erwarten wäre. Mit „`%`“ (Modulo) berechnet man den Rest bei einer Integerdivision. In unserem Beispiel also `22 % 4 = 2`. Modulo ist bei der Teilbarkeitsüberprüfung sehr nützlich.

OP	Bedeutung	Bsp.	Ergebnis
`+`	Addition	`7 + 5`	`12`
`-`	Subtraktion	`7 - 5`	`2`
`*`	Multiplikation	`8 * 5`	`40`
`/`	Division	`8 / 2`	`4`
`%`	Modulo	`8 % 5`	`3`

Logische Operatoren

Die logischen Operatoren liefern als Ergebnis `true` (`=1`) oder `false` (`=0`) zurück. Beispiele findet man in der Tabelle rechts.

OP	Bedeutung	Bsp.	Ergebnis
`<`	kleiner als	`1 < 2`	`true`
`<=`	kleiner oder gleich	`1 <= 2`	`true`
`>`	größer als	`1 > 2`	`false`
`>=`	größer oder gleich	`1 >= 2`	`false`
`==`	gleich	`1 == 2`	`false`
`!=`	ungleich	`1!= 2`	`true`
`&&`	logisches `UND`	`true && false`	`false`
`||`	logisches `ODER`	`true || false`	`true`

UND

Eine logische UND-Anweisung wertet zwei Ausdrücke aus. Sind beide Ausdrücke wahr (`true`), dann ist auch die UND-Anweisung wahr.

Beispiel:

Wenn du Hunger hast UND auch Geld, dann kannst du dir etwas zum Essen kaufen.

```
(Magen == leer) && (Geldbörse == voll)
```

ODER

Eine logische ODER-Anweisung wertet ebenfalls zwei Ausdrücke aus. Ist einer der beiden wahr (`true`), dann ist auch die ODER-Anweisung wahr.

Beispiel:

Wenn du Geld hast ODER einen Scheck, dann kannst du deine Rechnung bezahlen.

```
(Geldbörse == voll) || (Scheck == gedeckt)
```

Aufgaben

1. Ist `x = 5 + 7` ein Ausdruck? Welchen Wert hat er?
2. Was ist der Wert von `201 / 4`?
3. Was ist der Wert von `201 % 4`?
4. `x`, `a` und `b` sind Integer Variablen. Welchen Wert haben sie nach folgenden Anweisungen?

   ```
   x = 23;          a = x++;          b = x--;
   ```

5. Welchen Wert hat der Ausdruck `8 + 2 * 3` ?
6. Was ist der Unterschied zwischen `x = 3` und `x == 3` ?
7. Sind die folgenden Ausdrücke wahr (`true`) oder falsch (`false`)?

   a) `0`, b) `1`, c) `-1`, d) `x = 0`, e) `x == 0` (Mit der Voraussetzung, dass `x` den Wert `0` hat.)
8. Schreibe ein Programm, das sowohl das Volumen als auch die Oberfläche einer Kugel berechnet.