

Ausdrücke und Anweisungen

Ausdrücke

Ein Ausdruck ist eine

- Kombination aus Operatoren (z.B. =, -, /, * . . .) und Operanden (Variablen, Konstanten, Funktionen),
- die einen Wert (einen Rückgabewert, ein Resultat) hat.

Beispiele in C++:

```
a+b
a>b
a+b/c
```

Es gibt zwei Typen von Ausdrücken:

- Left values (kurz: Lvalues): Inhalt einer Speicheradresse (Objekte), nur Lvalues können links in einer Zuweisung stehen.
- Right values (Rvalues): alle anderen Ausdrücke, also solche, die nicht links in einer Zuweisung stehen können.

Beispiele: Lvalues sind

```
x,y
```

denn dies sind Inhalte von Speicheradressen. Keine Lvalues, und daher Rvalues sind:

```
1.0, 1.0/x, x+y
```

Folgende Zuweisungen sind möglich, d.h. links vom = steht ein Lvalue:

```
x=a+b;
x=1.0+2.0/3.0;
x=1.0/x;
x=y;
```

Folgende Zuweisungen sind nicht möglich:

```
1.0/x=y;
2.0/x=1.0/y;
```

da links kein Lvalue steht.

Da ein Ausdruck immer einen Wert hat, hat er auch immer einen (Daten-)Typ, nämlich den Datentyp seines Wertes (Resultats).

Ein Ausdruck kann Seiteneffekte haben, s.u.

Anweisungen

Ein Ausdruck wird in C++ durch ein nachfolgendes Semikolon zur Anweisung. Das Semikolon bewirkt, dass der Rückgabewert des Ausdrucks unterdrückt wird.

Beispiel:

Zuweisungsausdruck: `a=b`

einfache Zuweisungsanweisung: `a=b;`

Neben einfachen Anweisungen gibt es

- zusammengesetzte
- wiederholende
- bedingte

Anweisungen und Kombinationen davon.

In einem Programm oder einer Programmiersprache spricht man auch von Kontrollstrukturen, die man benutzt, um solche komplexeren Anweisungen zu konstruieren.

Dokumentationstechniken

Ziele:

- Darstellung eines Algorithmus
- unabhängig von Programmiersprachen
- übersichtlich, kompakt
- implementierbar für Programmierer/in in beliebiger Sprache

Möglichkeiten:

- a) Pseudocode: programmiersprachen-ähnlich (meist ohne Datentypen, spezifische Details wie Semikolon bei C++) , Strukturierung durch Einrücken, *begin* und /oder *end*.
- b) Struktogramm: grafisch, gesamter Algorithmus ein großes Rechteck, das von oben nach unten abgearbeitet wird
- c) Flussdiagramm: grafisch, mit Pfeilen, Eingang (Beginn) oben, Algorithmus (und damit Programmfluss) folgt den Pfeilen.

Seiteneffekte

Ein Seiteneffekt eines Ausdrucks ist ein Effekt auf Speicherinhalte, der sich nicht aus einer Zuweisungsanweisung an einen Lvalue ergibt. Seiteneffekte sollten in einem gut geschriebenen Programm vermieden werden.

Beispiele für Seiteneffekte:

1. Fehler in einer bedingten Anweisung

```
if (a=b)
    cout << a;
else
    cout << b;
```

`a=b` gibt Wert zurück, nämlich den Wert von `b`.

- Ist `b= 0` (und ganzzahlig), dann hat `a=b` den Wert `0`, und das wird dem Booleschen Wert `false` zugeordnet.
- Ist `b≠ 0` (und ganzzahlig), dann hat `a=b` einen Wert ungleich `0`, und das wird dem Booleschen Wert `true` zugeordnet.

Die Zuweisung `a=b` ist ein Seiteneffekt der `if`-Anweisung. (Fehler: `a=b` statt `a==b`).

2. Anweisung: C++ erlaubt statt

```
b=a;
c=c+b;
```

zu schreiben:

```
c+=b=a;
```

Dies ist äquivalent zu `c=c+(b=a)`;

`b=a` ist hier ein Rvalue, d.h ein Ausdruck, keine Anweisung. Ihr Rückgabewert ist der Wert von `a`, der gleichzeitig `b` zugewiesen wird. Dies ist der Seiteneffekt der gesamten Anweisung. Besser ist also die ursprüngliche Version.

3. `i=i++`; Seiteneffekt `i++`.