

Aufgabe 1.30 :
Schreibe ein Programm `DM_in_Euro.java` zur Umrechnung eines DM-Betrags in Euro unter Verwendung einer Konstanten für den Umrechnungsfaktor. Das Programm soll den DM-Betrag als Parameter verarbeiten.

Der Aufruf von `DM_in_Euro 1.40` sollte die Ausgabe `1.40 DM = 0.51129 Euro` ergeben.

Aufgabe 1.31 :
Schreibe ein Programm `Rechnung1.java` Dieses Programm soll eine Artikelbezeichnung, und den zugehörigen Nettopreis vom Benutzer abfragen und dann eine vollständige Rechnung ausgeben. Die Mehrwertsteuer soll als Konstante verarbeitet werden.

Beispiel:
Artikelbezeichnung : Sortiment Schrauben
Nettopreis : 12.50

Rechnung

Sortiment Schrauben	Netto:	14.50 Euro
	16% MwSt:	2.32 Euro
	Betrag:	16.82 Euro

Aufgabe 1.32 :
Schreibe ein Programm `Rechnung2.java` Dieses Programm soll zusätzlich die Anzahl der Artikel abfragen

Ausdrücke

Bei einfachen Datentypen besteht ein Ausdruck immer aus mindestens einem Operator und einem oder mehreren Operanden:

- y = 70** Der Variablen y wird 70 als Wert zugewiesen
13 + 4 Die Summe der Zahlen 13 und 4 wird gebildet
x > y Es wird geprüft, ob der Inhalt der Variablen x größer ist als der von y (true oder false)

i++ Der Inhalt der Variablen i wird um 1 erhöht (inkrementiert)

Arithmetische Ausdrücke

Arithmetische Operatoren sind die Operatoren für die 4 Grundrechenarten (+, -, *, /), außerdem der Operator % für die Modulo-Division, die den Rest bei der ganzzahligen Division ergibt:

$$23 \% 5 = 3 \quad (23 : 5 = 4 \text{ Rest } 3)$$
$$69 \% 8 = 5 \quad (67 : 8 = 8 \text{ Rest } 5)$$

und den Inkrement- und Dekrement-Operatoren ++ und --, die eine Integer-Variable um eins erhöhen bzw. erniedrigen.

Zuweisungen

Zuweisungen werden immer von rechts nach links ausgewertet.

$x = x + 30$ **!! Für einen Mathematiker Unsinn**

Zuerst wird die Summe des Inhalts der Variablen y und der Zahl 30 gebildet . Dann wird dieser neue Wert der Variablen x zugewiesen.

Kompakte Zuweisungen:

Dies lässt sich auch kurz wie folgt ausdrücken:

$x += 7$ ist gleichbedeutend mit $X = X + 7$

$x -= 7$ ist gleichbedeutend mit $X = X - 7$

$x *= 7$ ist gleichbedeutend mit $X = X * 7$

$x /= 7$ ist gleichbedeutend mit $X = X / 7$

Für die **Erhöhung** einer Integer-Variablen **um 1** erhöht gibt es eine weitere Schreibweise:

$x = x + 1$ $x += 1$ bzw. $x++$

und entsprechend für die Erniedrigung um 1

$x = x - 1$ $x -= 1$ bzw. $x--$

VORSICHT: Zwischen x und $++$ bzw. $--$ darf kein Leerzeichen stehen!

Aufgabe 1.40: Was gibt das folgende Programm aus?

```
// Operatoren.java
// Programm zum Testen der Zuweisungsoperatoren

class Operatoren {
    public static void main(String[] args)
    {
        int x = 2, y = 1, z;

        x++;
        y += x;
        y *= x;
        y = y - 1;
        z = y % x;
        System.out.println("Wert von x: " + x);
        System.out.println("Wert von y: " + y);
        System.out.println("Wert von z: " + z);
    }
}
```

Ein typischer Anfängerfehler !!

Der Typ des Ergebnisses eines Ausdrucks hängt von den Typen der Operanden ab:

Wenn beide Operanden vom selben Typ sind, so ist das Ergebnis ebenfalls von diesem Typ.

Vorsicht bei der Division von Integervariablen :

$x = 4 / 3$ weist x den Wert 1.0 zu auch wenn x als eine Variable vom Typ Double deklariert ist!

$x = 4.0 / 3.0$ weist x den Wert 1.333.. zu.

Wenn Operanden verschiedenen Typs beteiligt sind, so werden die Operanden automatisch vor der Verknüpfung in den "größten" Datentyp gewandelt. Das Ergebnis einer Addition von einer int- und einer double- Variablen ist also vom Typ double.

$x = 4 / 3.0$ setzt x ebenfalls auf 1.33333333

Aufgabe 1.50:

Welchen Wert liefern die folgenden Anweisungen ?

(x sei jeweils eine Variable vom Typ double):

a) $x = 1/3$; b) $x = (\text{double})(1/3)$; c) $x = ((\text{double})(1)) / 3$;

Schreibe ein passendes Programm zum Test.

Aufgabe 1.51:

Schreiben Sie ein Programm, das den Umfang und den Flächeninhalt eines Kreises berechnet und ausgibt. Das Programm soll mit einem Argument, dem Radius des Kreises, aufgerufen werden. Der Wert von π ist in einer Konstanten zu speichern.

Aufgabe 1.52:

Schreiben Sie ein Programm, das den durchschnittlichen Benzinverbrauch je 100 km berechnet. Als Argumente werden der Anfangs- und Endkilometerstand und die verbrauchte Benzinmenge eingegeben.

Logische Ausdrücke

Vergleiche

Java besitzt diverse Operatoren, die bei Vergleichen verwendet werden. Bei der Auswertung solcher Vergleiche gibt es nur zwei Möglichkeiten: Abhängig davon, ob der Vergleich aufgeht oder nicht, wird der boolesche Wert `true` oder `false` zurückgegeben.

Beispiel :

```
boolean antwort; //Deklaration einer Variablen vom Typ boolean
int age = 31;
antwort = (age < 25); // Zuweisung von false an die Variable
// da 31<25 eine falsche Aussage ist!
```

Die Tabelle zeigt die einzelnen **Vergleichsoperatoren** :

Operator	Bedeutung	Beispiel
<code>==</code>	gleich	<code>x == 3</code>
<code>!=</code>	ungleich	<code>x != 3</code>
<code><</code>	kleiner als	<code>x < 3</code>
<code>></code>	größer als	<code>x > 3</code>
<code><=</code>	kleiner oder gleich	<code>x <= 3</code>
<code>>=</code>	größer oder gleich	<code>x >= 3</code>

Aufpassen: `=` ist kein logischer Vergleichsoperator sondern ein Zuweisungsoperator!!

Logische Operatoren

Diese Operatoren werden für die logischen Verknüpfungen **AND (UND)**, **OR (ODER)**, **XOR (exklusives ODER)** und **NOT (logisches NICHT)** verwendet.

Beispiel: boolean x=true; y=false

Operator	Bedeutung	Beispiel	Wert von z
&	UND (AND)	Z=x&y	false
&&	UND (AND)	Z=x&&y	false
	ODER (OR)	Z=x y	true
	ODER (OR)	Zx =x y	true
!	NICHT(NOT)	Z=!x	false
-^	EXODER(XOR)	Z=x-^y	true

Der Unterschied zwischen & und && bzw. | und || liegt in der Auswertung kombinierter Ausdrücke.

- Wenn & verwendet wird, werden immer die Ausdrücke auf beiden Seiten des & ausgewertet.
- Wenn dagegen && verwendet wird und die linke Seite von && false ergibt, wird der Ausdruck auf der rechten Seite nicht ausgewertet.

Wenn bei dem UND-Operator der linke Operand false ist, dann ist das Ergebnis der Undverknüpfung auf jeden Fall false !

Wertetabelle für die logischen Verknüpfungen:

Boolsche Variable		Negation NOT	Konjunktion AND	Disjunktion OR	ExDisjunktion XOR
A1	A2	! A1	A1 & A2	A1 A2	A1 -^ A2
true	true	false	true	true	false
true	false	false	false	true	true
false	true	true	false	true	true
false	false	true	false	false	false

Operatorhierarchie

Im allgemeinen gibt es folgende Reihenfolge, wobei der erste Eintrag der Liste die höchste Priorität besitzt:

- **Klammern**
 - **Inkrement- und Dekrement-Operationen**
 - **Arithmetische Operationen**
(Punkt- vor Strichrechnung)
 - **Vergleiche**
 - **Logische Operationen**
 - **Zuweisungsausdrücke**

Wenn zwei Operationen dieselbe Hierarchiestufe (Bindungskraft) besitzen, wird der Ausdruck von links beginnend nach rechts ausgewertet.

Punkt- vor Strichrechnung:

$y = 6 + 4/2$ weist y den Wert 8 zu

Klammern genießen allerhöchste Präzedenz:

$y = (6 + 4)/2$ weist y den Wert 5 zu

Klammern können auch verschachtelt werden.

Außerdem sollte man Klammern verwenden, wenn man sich der Rangfolge der verwendeten Operatoren nicht sicher ist. (**Verwende grundsätzlich Klammern !!**)

Prä- und Postfixe

Die Inkrement und Dekrement-Operatoren können sowohl **postfix (i++)** als auch **präfix (++i)** verwendet werden. Für die nachfolgenden Programmzeilen spielt es keine Rolle, ob prä- oder postfixe Form verwendet wird. Der Unterschied zeigt sich nur, wenn diese Operatoren in komplizierteren Ausdrücken verwendet werden. Dann gilt:

- Bei präfixer Verwendung (**++i**) wird zuerst inkrementiert und der inkrementierte Wert im aktuellen Ausdruck verwendet.
- Bei postfixer Verwendung (**i++**) wird der alte Wert im Ausdruck verwendet und erst danach inkrementiert.

Beispiel:

```
int x, y, z; // x, y, und z werden deklariert
```

```
x = 42;           // x wird der Wert 42 zugewiesen
y = x++;         // y wird der Wert von x (42) zugewiesen,
                // bevor x inkrementiert wird
                // anschließend wird x auf 43
                // erhöht
z = ++x;        // x wird auf 44 inkrementiert, und z wird
                // der Wert von x zugewiesen
```

Am Ende haben x und z den Wert 44, und y hat den Wert 42.