Tagesprogramm

Operatoren und Werte

Lösen einer Programmieraufgabe

Klammerung und Auswahl bei Operatoren

Stelligkeit

`-` in `-x` ist anderer Operator als in `x - y`

Priorität bei gleicher Stelligkeit

`*` höhere Priorität als `+` $\longrightarrow$
`x * y + z * 2` entspricht `(x * y) + (z * 2)`

Assoziativität bei gleicher Stelligkeit und Priorität

`/` ist linksassoziativ $\longrightarrow$ `x / y / z` entspricht `(x / y) / z`
`=` ist rechtsassoziativ $\longrightarrow$ `x = y = z` entspricht `x = (y = z)`

Typen der Operanden

bestimmen auszuführende Operation wenn überladen,
beeinflussen Stelligkeit, Priorität und Assoziativität aber nicht

Kurzschlussopteratoren

Kurzschlussoperatoren `&&` und `||` werten rechten Operanden nur wenn nötig aus

```
true  && b   → b ausgewertet
false && b   → b nicht ausgewertet
true  || b   → b nicht ausgewertet
false || b   → b ausgewertet
```

Operatoren `&` und `|` werten jedesmal beide Operanden aus

Operatoren für Zuweisungen

Inkrement und Dekrement ++ und --

```
int i=1; System.out.println(i++ + "," + i);  → 1,2
int i=1; System.out.println(++i + "," + i);  → 2,2
int i=1; System.out.println(i-- + "," + i);  → 1,0
int i=1; System.out.println(--i + "," + i);  → 0,0
```

Zuweisung =

```
int i=1; System.out.println((i = 3) + "," + i);  → 3,3
```

kombinierte Zuweisung *=, /=, %=, +=, -=, <<=, >>=, >>>=, &=, ^=, |=

```
int i=3; System.out.println((i *= 2) + "," + i);  → 6,6
int i=3; System.out.println((i /= 2) + "," + i);  → 1,1
int i=3; System.out.println((i += 2) + "," + i);  → 5,5
...
```

Aufgabe: Verzwickte Seiteneffekte

```
int x = 2;
x += x++ + x;
System.out.println(x);
```

Welcher Wert wird hier ausgegeben?

5

6

7

8

9

Final

außer Initialisierung keine Zuweisung an `final` Variable (oder Parameter)

```
final int i = 3;
final int j; j = 5;
final String r = "1. String", s = "2. String";
final String r, s; r = "1. String"; s = "2. String";
```

L- und R-Werte

L-Wert: Ausdruck, der links von = stehen kann

Variablen, außer wenn `final` und initialisiert

Elemente von **Arrays**:

```
String[] strings = new String[10];
strings[5] = "String";
```

R-Wert: Ausdruck, der rechts von = stehen kann

alle Ausdrücke außer noch nicht initialisierten Variablen

Operand von ++ und -- ist gleichzeitig L-Wert und R-Wert

Typumwandlung

elementare Typen nach Größe geordnet (außer `boolean`)

`byte`, `short`, `char`, `int`, `long`, `float`, `double`

automatische **Typumwandlung** von von kleinerem zu größerem Typ (außer `char`)

```
byte b = 8; int i = b;
int j = 3; double p = j + b;
char c = 'a'; long l = c;
```

Cast ist explizite Typumwandlung

```
long k = 1L; short x = (short)k;
int y = 2; double v = 3.0; int z = y + (int)v;
```

Achtung: Typumwandlungen können zu Datenverlust führen

Aufgabe: Verzwickte Typumwandlungen

Welche der folgenden Ausgabe-Anweisungen geben den angegebenen Wert aus?

A `System.out.println('a' + 1);` → b

B `System.out.println((char)('a' + 1));` → b

C `System.out.println((long)1.5 + 1.5);` → 3

D `System.out.println((long)1.5 + 1.5);` → 2.5

E `System.out.println(1.5 + (long)1.6);` → 3.5

F `System.out.println(1.5 + (long)1.6);` → 2.5

Schritte beim Lösen einer Aufgabe

Analyse

Entwurf

Implementierung

Testen

Debuggen

Aufgabe: Berechnung der Faktoriellen

Entwickeln Sie eine Methode zur Berechnung der Faktoriellen einer ganzen Zahl.