**Ausdrücke und Anweisungen**

**Ausdrücke**

- kleinstes ausführbares Element eines Programms

- bestehen aus mindestens einem Operator und einem oder mehreren Operanden

Aufgaben: 1) Variablen Werte zuweisen

2) nummerische Berechnung durchführen

3) logische Bedingungen formulieren

**Arten von Operatoren**

Arithmetische

+ positives Vorzeichen +n ist gleichbedeutend mit n

- negatives Vorzeichen -n ist gleichbedeutend mit n

+ Addition a + b ist die Summe von a und b

- Differenz a – b ist die Differenz von a und b

\* Produkt a \* b ist das Produkt von a und b

/ Quotient a / b ist der Quotient von a und b

% Restwert (Modulo) a % b ist der Restwert von a / b

++ Präinkrement ++a erhöht a um 1 und gibt dann a

++ Postinkrement a++ gibt a und erhöht dann a um 1

-- Präinkrement --a verringert a um 1 und gibt dann a

-- Postinkrement a—gibt a und verringert es dann um 1

Relationale

== Gleich a == b ist true wenn a gleich b ist

!= Ungleich a != b ist true wenn a ungleich b ist

< Kleiner a < b ist true wenn a kleiner b ist

<= Kleiner Gleich a <= b ist true wenn a kleiner oder gleich b ist

> Größer a > b ist true wenn a größer b ist

>= Größer gleich a >= b ist true wenn a größer oder gleich b ist

Logische

! logisches nicht !a ist true wenn a false ist und andersrum

&& Und (Short-Circuit) ist true wenn a und b true sind (ist a falsch wird b nicht geprüft)

|| Oder (Short-Circuit) ist true wenn a oder b true sind (ist a true wird b nicht geprüft)

& Und ist true wenn a und b true sind

| Oder ist true wenn a oder b true sind

^ Exklusives Oder ist true wenn a und b unterschiedlich sind

Zuweisungen

= Einfache Zuweisung a = b weist a den Wert b zu und gibt b

+= Additionszuweisung a += b weist a den Wert a + b zu und gibt ihn

-= Subtraktionszuweisung a -= b weist a den Wert a - b zu und gibt ihn

\*= Multiplikationszuweisung a \*= b weist a den Wert a \* b zu und gibt ihn

/= Divisionszuweisung a /= b weist a den Wert a / b zu und gibt ihn

%= Modulo Zuweisung a %= b weist a den Wert a % b zu und gibt ihn

Sonstige

? Fragezeichenoperator a ? b : c gibt b wenn a true ist sonst c

(type) Type-Operator dient der expliziten Konvertierung

String +- String Verkettung verkettet Strings

**Anweisungen**

Leere Anweisung ; - einfachste Anweisung in Java - wird verwendet wenn Anweisung benötigt aber nicht sinnvoll ist

Anweisungsblock {

Anweisung1; - Zusammenfassung von Anweisungen

Anweisung2; - wird verwendet wenn mehrere gebraucht

… aber nur eine Anweisung erlaubt ist

Anweisungn;

}

**Kontrollfluss**

- Bezeichnet in der Informatik die zeitliche Abfolge der Befehle eines Computerprogramms

- Ist gewöhnlich durch Reihenfolge der Befehle im Programm festgelegt

- Kann durch Kontrollstrukturen vom sequenziellen Ablauf abgewandt werden

(z.B. Schleifen und Verzweigungen)

**Verzweigungen**

If-Anweisung - Ausdruck kann aus einem relationalen Operator oder mehreren, die mit logischen verbunden sind bestehen

- Anweisung wird ausgeführt wenn Ausdruck true ist

**if (ausdruck)**

**anweisung**;

If-Else-Anweisung - Anweisung1 wird ausgeführt wenn Ausdruck true ist, sonst Anweisung2

**if (ausdruck)**

**anweisung1;**

**else**

**anweisung2;**

If-Else-If-Anweisung - Anweisung1 wird ausgeführt wenn Ausdruck1 true ist, sonst wird Ausdruck2 ausgewertet

- ist Ausdruck2 dann true wird Anweisung2 ausgeführt sonst Anweisung3

**if (ausdruck1)**

**anweisung1;**

**else if (ausdruck2)**

**anweisung2;**

**else**

**anweisung3;**

Switch-Anweisung - bietet Möglichkeit der Mehrfachverzweigung

- Ausdruck wird ausgewertet und dann mit den Konstanten verglichen

- Bei Gleichheit werden alle Anweisungen des Case ausgeführt

- kann mit break verhindert werden

- passt keine Konstante wird die Optionale Default-Anweisung ausgeführt

**switch (ausdruck) {**

**case konstante1**

**anweisung1;**

**case konstante2**

**anweisung2;**

**…**

**default:**

**default\_anweisung;**

**}**

**Schleifen**

While-Schleife - Kopfgesteuert

- Ausdruck muss Boolean sein

- Anweisung wird ausgeführt solange Ausdruck true ist

**while (ausdruck)**

**anweisung;**

Do-While-Schleife - Fußgesteuert

- Anweisung wird mindestens einmal ausgeführt

- Rest siehe While-Schleife

**do**

**anweisung;**

**while (ausdruck);**

For-Schleife - Zählende Schleife

- init: initialisiert lokalen Schleifenzähler

- test: Abbruchbedingung

- update: verändert Schleifenzähler

- init, test, update sind optional

- gibt es ein test wird die Anweisung ausgeführt wenn das Ergebnis true ist

- gibt es kein test setzt der Compiler true als konstante

**For (init, test, update)**

**anweisung;**

**Break- und Continue-Anweisung**

Break - Schleife wird verlassen

- erster Befehl nach Schleife wird ausgeführt

Continue - Aktueller Schleifendurchlauf wird beendet

- Programm springt zum Ende der Schleife und fängt mit erneutem Durchlauf mit Überprüfung der Abbruchbedingung an