

1. Ganze Zahlen

Für die ganzen Zahlen sind folgende Operatoren definiert: Addition, Subtraktion, Multiplikation, Division und Restberechnung. Beide Operanden und Ergebnis sind ganze Zahlen. Ein eher seltener Operator ist die Invertierung des Vorzeichens.

```
VAR i, j, k : integer;    // Platz für Deklarationen
BEGIN
    i := 7; j := 3;
    k := i+j;    WRITELN('      Addition:', k);
    k := i-j;    WRITELN('      Subtraktion:', k);
    k := i*j;    WRITELN('      Produkt:', k);
    k := i div j; WRITELN('Ganzzahlige Division:', k); // Operator / wäre falsch!
    k := i mod j; WRITELN('      Rest der Division:', k);
END.
```

```
func main() {
    var i, j, k int    // Platz für Deklarationen
    i = 7; j = 3
    k = i+j; fmt.Println("      Addition:", k)
    k = i-j; fmt.Println("      Subtraktion:", k)
    k = i*j; fmt.Println("      Produkt:", k)
    k = i/j; fmt.Println("Ganzzahlige Division:", k)
    k = i%j; fmt.Println("      Rest der Division:", k)
}
```

Aufgabe 1. Schreiben Sie Programme, wo Sie Invertierung (unäre Operation) für ganze Zahlen testen. Testen Sie auch kompliziertere Berechnungen.

Aufgabe 2. Schreiben Sie Programme, wo Sie in beiden Sprachen folgende Operatoren für ganze Zahlen testen: +=, -=, *=, /=, ++, -- (bekannt aus den C-artigen Sprachen C/C++/Java). Merken Sie sich die Bedeutung dieser Operatoren.

2. Fließkommazahlen

Für Fließkommazahlen sind fast die gleichen Operatoren definiert, außer Rest der Division, der ausnahmsweise in einigen Sprachen definiert werden kann. Beide Operanden und Ergebnis sind Fließkommazahlen.

Aufgabe 3. Schreiben Sie Programme, wo Sie die Operatoren für Fließkommazahlen testen, die Sie aus ganzen Zahlen kennen. Funktioniert Rest der Division für Fließkommazahlen? Merken Sie sich die Bedeutung aller Operatoren.

Aufgabe 4. Schreiben Sie Programme, wo Sie Uhrzeiten eingeben, berechnen und ausgeben. Legen Sie die ganzzahligen Variablen für Stunde, Minute und Sekunde an. Weisen Sie den Variablen Werte zu, welche die Uhrzeit 09:42:51 darstellen. Geben Sie die drei Variablen als Uhrzeit aus (mit : dazwischen). Lesen Sie von der Tastatur Stunden, Minute und Sekunde ein. Berechnen Sie Anzahl der Sekunden seit Mitternacht und geben Sie sie aus. Berechnen Sie die Anzahl der noch verbleibenden Sekunden des Tages und geben Sie sie aus. Berechnen sie, wie viel Prozent des Tages bereits verstrichen sind, und geben Sie diesen Wert mit zwei Nachkommastellen aus.

3. Zeichenketten

Die Basisoperatoren für Zeichenketten sind Verkettung (Konkatenation), Subtraktion (aus einer Zeichenkette eine andere subtrahieren), Berechnung der Länge der Zeichenkette, Konvertierung in Groß- oder Kleinbuchstaben.

```
VAR s : string = 'Max Frei';
VAR t : string = 'Student';
VAR k : integer;
BEGIN
    s := s + t; WRITELN('Verkettung:', s);
    s := copy(s,2,8); WRITELN('Substr:', s);
    k := length(s); WRITELN('Länge:', k);
END.
```

```
func main() {
    var s string = "Max Frei"
    var t string = "Student"
    var k int
    s = s + t; fmt.Println("Verkettung:", s)
    t = s[0:7]; fmt.Println("Substr:", t)
    k = len(s); fmt.Println("Länge:", k)
}
```

Aufgabe 5. Schreiben Sie Programme, in denen die Konvertierung einer von der Tastatur eingegebenen Zeichenkette in Groß- oder in Kleinbuchstaben stattfindet. Was passiert, wenn die Zeichenkette nicht nur die Buchstaben, sondern auch Ziffern oder spezielle Symbole beinhaltet?

4. Logische Variablen

Logische Variablen haben den Datentyp boolean/bool und können nur einen Wert (true oder false) enthalten. Sie entstehen als Ergebnisse der logischen Ausdrücken (Vergleiche).

```
VAR b : boolean = true;
VAR k : integer = 7;
BEGIN
    b := k > 10;          // Weitere Vergleichsoperatoren: <, >=, <=, <>, =
    WRITELN('b ist: ', b);
END.
```

```
func main() {
    var b bool = true
    var k int = 7
    b = k > 10           // Weitere Vergleichsoperatoren: <, >=, <=, !=, ==
    fmt.Println("b ist: ", b)
}
```

Aufgabe 6. Recherchieren Sie, wie sehen die logischen Operatoren AND, OR, NOT in beiden Sprachen aus. Schreiben Sie Programme, wo Sie die logischen Variablen mit Hilfe dieser Operatoren verknüpfen (berechnen). Ergebnisse (true oder false) sollen auf dem Bildschirm ausgegeben werden.

Aufgabe 7. Schreiben Sie Programme, wo Sie die ganzen Zahlen mit einander vergleichen, die Fließkommazahlen, Zeichenketten und logische Variablen. Ergebnisse (true oder false) sollen auf dem Bildschirm ausgegeben werden. Recherchieren Sie, wie man die Zeichenketten mit einander vergleicht. Versuchen Sie die Variablen von unterschiedlichen Typen mit einander zu vergleichen. Ergebnisse sollen auf dem Bildschirm ausgegeben werden. Merken Sie sich, dass Vergleich der Zeichenketten fast in allen Sprachen besonders gemacht werden soll.

5. Konvertierung

Die Werte der Variablen eines Typen können, manchmal müssen, in einen anderen Typ umgewandelt werden. Es gibt zwei Möglichkeiten, dies zu tun: einfach durch direkte Zuweisung, oder mit Hilfe einiger Konstruktionen oder Funktionen. Selbstverständlich, ist es nicht immer möglich, oder möglich, aber mit Verlust der Genauigkeit.

```
VAR r : real = 4.2;
VAR i : integer = 7;
BEGIN
    r := i; // ist möglich, umgekehrt aber nicht
    WRITELN('r= ', r, ' i= ', i);
    r := 4.2;
    i := trunc(r); // explizite Umwandlung
    WRITELN('r= ', r:5:2, ' i= ', i);
END.
```

```
var i int = 42           // Versuchen Sie mit -42
var f float64 = float64(i)
var u uint = uint(f)     //unsigned
fmt.Println(i, f, u)
f = 4.2
i = int(f) // oder (int)(f), (int) heißt Casting
fmt.Println(i, f, u)
```

Aufgabe 8. Schreiben Sie Programme, wo Sie die Zahlentypen (ganze Zahlen und Fließkommazahlen) in die Zeichenketten und umgekehrt umwandeln.

6. Arithmetische Ausdrücke

Arithmetische Ausdrücke beinhalten nur die ganzzahligen und Fließkomma-Operationen wie +, -, , / , mod und Aufrufe der Funktionen, z.B.:

$$\text{Geldzulage} = (\text{Grundgehalt} * 0.7 + \text{AnzahlKinder} * 100 + \text{Arbeitsjahre} / 3) * 98.7$$

Aufgabe 9. Schreiben Sie Programme, wo Sie die Gaußsche Wochentagsformel für Berechnung des Wochentages des 1. Januar eines Jahres implementieren. Von der Tastatur wird eine ganze Zahlen eingelesen – das Jahr. Ausgegeben wird auch eine ganze Zahl – Nummer des Wochentages.

7. Logische Ausdrücke

Logische Ausdrücke kann man in einfache und zusammengesetzte unterteilen.

Einfache logische Ausdrücke sind so aufgebaut:

Arithmetischer Ausdruck Vergleichsoperator Arithmetischer Ausdruck

z.B. Alter > 36
 Gehalt <= 5000
 A*120 - 36 = 152*XYZ + 12

Zusammengesetzte logische Ausdrücke sind so aufgebaut:

Einfacher logischer Ausdruck Logischer Operator (AND OR) Einfacher logischer Ausdruck

NOT Einfacher logischer Ausdruck

Alter > 36 AND Gehalt <= 5000 OR NOT (A*120 - 36 = 152*XYZ + 12)

Aufgabe 10. Schreiben Sie Programme, in dem Sie folgende logische Formel berechnen und Ergebnisse ausgeben:

$P = X \text{ and } Y \text{ and } (\text{not } X \text{ or not } Y)$

$Q = (X \text{ or } Y) \text{ and } (\text{not } X \text{ or not } Z)$

$R = (Y \text{ and } Z) \text{ or not } (X \text{ and not } Y \text{ and not } Z)$

Die Variablen X, Y, Z werden von der Tastatur eingelesen (Go) oder im Programm festgelegt (Pascal).