2.2 Einfache Datenstrukturen

Konstante und Variable

Die Begriffe "Konstante" und "Variable" haben zunächst auch in der Informatik dieselbe grundsätzliche Bedeutung wie in der übrigen Mathematik.

Variablen sind Speicherplätze im Rechner, in denen Zahlen, Buchstaben, Wörter und andere Objekte gespeichert werden. Eine Variable hat einen Namen (Variablenname), einen so genannten Datentyp und einen Dateninhalt. Mit der Variablen kann genauso gearbeitet werden wie mit den Daten selbst.

In Pascal werden Variablennamen und Typ im Deklarationsteil gemäß folgender Syntax festgelegt:
VAR *Liste von Variablennamen: Variablentyp*
Der Dateninhalt wird der Variablen aber erst im Programm zugewiesen und kann jederzeit geändert werden.

Beispiel:
```
program variable;
var laenge, breite, umfang: real;
...
begin
read (laenge, breite);
umfang := 2 * (laenge + breite);
write (umfang)
end.
```

Eine Konstante ist ähnlich der Variablen ein Speicherplatz, der einen Namen und einen Dateninhalt hat. Jedoch lässt sich der Dateninhalt nicht mehr im Programm verändern. In Pascal wird der Inhalt im Deklarationsteil nach folgender Syntax festgelegt:
CONST *Konstantenname = Wert der Konstanten*

Beispiel:
```
CONST pi = 3.14159;
```

Einfache Datentypen in Pascal

INTEGER: Ganze Zahlen. Der Zahlenbereich ist beschränkt auf Zahlen von -32768 bis +32767.

REAL: Dezimalzahlen. Eine Dezimalzahl kann auf einem Computer ebenfalls nur mit einer beschränkten Genauigkeit angegeben werden. Sollten bei sehr kleinen Zahlen die Stellen hinter dem Dezimalpunkt oder bei sehr großen

Zahlen die Stellen davor nicht ausreichen, so werden die Zahlen in der Zehnerpotenz-Schreibweise (wie beim Taschenrechner) angegeben.
Beispiel: Statt 300 000 000 000 kann 3E11 (= $3 \cdot 10^{11}$) angegeben werden.

CHAR: Zeichen. Eine Variable dieses Typs kann ein beliebiges Zeichen aus dem Zeichensatz des Rechners enthalten.

BOOLEAN: Logische Variable. Dieser Variablentyp kennt nur zwei mögliche Werte: Wahr und Falsch. In Pascal heißen diese Werte **TRUE** (wahr) und **FALSE** (falsch).

Neben diesen einfachen Datentypen gibt es noch so genannte zusammengesetzte Datentypen, z. B.
STRING[n] bzw. **STRING(.n.)**: Zeichenkette. Eine Zeichenkette ist eine Folge von beliebigen Zeichen (z. B. auch Leerzeichen u. ä.). Ein String hat eine maximale Länge von 255 Zeichen, außer es wird durch [n] eine (kleinere) Länge vereinbart. Überzählige Zeichen werden einfach ignoriert.

1. Beispiel (Üben des Umgangs mit dem Datentyp **STRING**): "Spiegeln" einer Zeichenkette (z. B. Wort, Satz). Das Programm soll z. B. aus dem Wort "Neger" das Wort "Regen" erzeugen.
Dazu bietet sich folgender Algorithmus an:
a) Bestimme die Länge der Zeichenkette.
b) Wiederhole: Schreibe den letzten Buchstaben, hänge den vorletzten Buchstaben an usw., bis der erste Buchstabe geschrieben ist.

Das zugehörige Struktogramm kann so aussehen:

<table>
<tr><td rowspan="6">Wieder-
hole</td><td>Lies ein Wort ein und weise es der Variablen "WORT" zu.</td></tr>
<tr><td>laenge := Länge von wort</td></tr>
<tr><td>Schreibe "wort", wort</td></tr>
<tr><td>Wiederhole für i := laenge bis 1
<table><tr><td>Schreibe den i-ten Buchstaben von wort</td></tr></table></td></tr>
<tr><td>Drücke eine Taste ("j" oder belieb. andere Taste)</td></tr>
<tr><td colspan="2">bis die Taste nicht "j" ist</td></tr>
<tr><td colspan="2">Schreibe: "Programmende"</td></tr>
</table>

In Pascal werden für diesen Algorithmus folgende Funktionen bereitgestellt:

1. Die Funktion **LENGTH** ermittelt die Länge eines Strings gemäß folgender Syntax:
   LENGTH*(zeichenkette)*
   Das Ergebnis ist vom Typ **INTEGER**.
2. Die Funktion **COPY** entnimmt aus einer Zeichenkette ab einer bestimmten Position eine bestimmte Anzahl von Zeichen.
   Syntax: **COPY***(zeichenkette, position, teilstring)*
   Das Ergebnis ist vom Typ **STRING**.
3. Mit der Funktion **READKEY** aus der Unit **CRT** wird ein Zeichen von der Tastatur übernommen. Der Wert von **READKEY** ist vom Typ **CHAR**.
   Der Vorteil von **READKEY** (gegenüber einer Eingabe über **READ** oder **READLN**) ist, dass die Eingabe nicht mit "RETURN" bestätigt werden muss. Von Nachteil kann sein, dass nur ein Zeichen eingegeben werden kann, die Eingabe kein Bildschirmecho erzeugt und die Eingabe nicht mehr korrigierbar ist.
4. Kommt es nicht darauf an, welche Taste betätigt wird (z. B. wenn das Programm mit einem beliebigen Tastendruck fortgesetzt werden soll), sondern geht es nur um eine Wartefunktion, so kann auch eine andere Funktion aus der Unit **CRT** eingesetzt werden: Die Funktion **KEYPRESSED** liefert den Wert **FALSE**, solange keine Taste gedrückt wird, sonst **TRUE**.
   Beispiel: **REPEAT WRITE ('$') UNTIL KEYPRESSED**
   schreibt den Bildschirm mit Dollarzeichen voll, bis eine beliebige Taste gedrückt wird.

So kann das Spiegelungsprogramm in Pascal aussehen:

```
program programm_10_(stringmanipulation);
uses crt;
var wort  : string;
antwort   : char;
laenge, i : integer;

begin
     clrscr;
     writeln ('Wörterspiegel');
     writeln ('============'); writeln;
     repeat
          writeln ('Gib einen String (Wort, Satz o. ä.) ein');
          readln (wort);
          laenge := length(wort);
          for i := laenge downto 1 do
               write (copy(wort,i,1));
          writeln; writeln;
          writeln ('Nochmals? (j/bel. Taste)'); writeln;
          antwort := readkey
     until antwort <> 'j';
     writeln ('Programmende');
     readln
```

```
end.
```

Ein seltsamer Pascal-Datentyp: BOOLEAN

Aus der Aussagenlogik kennen wir sogenannte logische Aussagen. Sie können die Werte "wahr" oder "falsch" annehmen.

Beispiele:
1. "Paris ist die Hauptstadt von England" ist falsch.
2. "Paris ist die Hauptstadt von Frankreich" ist richtig.
3. "Boolesche Variablen sind leicht verständlich" ist objektiv nicht entscheidbar.

Alle drei Aussagen sind für unsere Arbeit mit dem Computer unbrauchbar. Insbesondere darf es niemals eine unentscheidbare Situation geben.

Vielmehr haben wir es mit Aussagen folgenden Typs zu tun:
1. 5 = 4 ist falsch
2. 5 > 4 ist richtig

In Pascal haben die Wahrheitswerte folgende Namen:
TRUE - richtig
FALSE - falsch
Außerdem kann man einen Wahrheitswert einer Variablen zuordnen, die vom Typ **BOOLEAN** ist.

Anmerkung:
Werden Daten durch die bekannten Operatoren <, <=, >, >=, =, <> verglichen, so ist das Ergebnis vom Typ **BOOLEAN** und könnte einer entsprechenden Variablen zugewiesen werden. Es ist aber auch die Verwendung der logischen Verknüpfungen **AND** (logisches **UND**), **OR** (logisches **ODER**), **XOR** (logisches **ENTWEDER - ODER**) und **NOT** (Negation) möglich.

Ein Beispiel:
Nach Eingabe zweier ganzer Zahlen sollen diese hinsichtlich ihrer Größe verglichen werden.

In Pascal kann dieses Problem so realisiert werden:

```
program boolesche_variable;

uses crt;
var x, y             : integer;
logische_bedingung   : boolean;
wort            : string [17];
antwort              : char;

begin
```

```
     repeat
          clrscr;
          write ('Gib zwei verschiedene ganze Zahlen ein. ');
          readln (x,y);
          logische_bedingung := x > y; {logische bedingung}
          if logische_bedingung
               then wort := ' ist größer als '
               else wort := ' ist kleiner als ';
          writeln (x, wort, y);
          if not logische_bedingung
          then writeln (y, ' als letzte und größte Zahl!');
          writeln;
          writeln ('Nochmals? (j/sonst. Taste)');
          antwort := readkey
     until antwort <> 'j'
end.
```