Einführung in Object Pascal mit DELPHI

Wenn Sie Delphi – Anwendungen erstellen, programmieren Sie in Object Pascal, einem objektorientierten Pascal – Dialekt.

1. Der Aufbau einer Prozedur

Eine Prozedur besteht aus einem Kopf und einem Körper. Der Körper wiederum besteht aus einem (optionalen) Deklarationsteil und einem Anweisungsteil.

Kopf
Körper
Deklarationsteil
Anweisungsteil

```
procedure name (parameterliste);

var   a, b, r: integer;
begin
        Anweisung1;
end;
```

Anmerkungen:
- Jede Programmanweisung wird mit einem Semikolon abgeschlossen.
- Eine Wertzuweisung erfolgt mit := Beispiel: a := 3;
- Kommentare: // einzeiliger Kommentar bzw. {mehrzeiliger Kommentar}

2. Variablen und Konstanten

Um eigene Elemente in ein Programm aufnehmen zu können, müssen Sie diese erst dem Compiler bekannt machen (deklarieren). Die Deklaration teilt dem Compiler mit, um was für ein Element es sich handelt (Konstante, Typ, Variable) und verbindet das neue Element mit einem Namen (Bezeichner). Prinzipiell sind diese Namen frei wählbar. Dies dient der besseren Lesbarkeit der Programme. Trotzdem sollten Sie bei der Namensgebung einige Regeln beachten:
- Bezeichner dürfen Buchstaben, Ziffern und Unterstriche enthalten. Leerzeichen, deutsche Umlaute und ß sind nicht zugelassen.
- Das erste Zeichen muss ein Buchstabe oder Unterstrich (_) sein.
- Es wird nicht zwischen Groß – und Kleinschreibung unterschieden.
- Die zum Sprachumfang von Object Pascal gehörenden Schlüsselwörter können nicht als Bezeichner verwendet werden. (siehe Delphi – Hilfe → Reservierte Wörter)

2.1 Konstanten

Eine Konstante ist ein Bezeichner, dessen Wert während des ganzen Programms gleich, d.h. konstant bleibt. Die Definition wird mit dem Schlüsselwort **const** eingeleitet.

Beispiel:
```
const Eulersche_Zahl = 2.718;
```

2.2 Variablen

Variablen sind Sprachelemente deren Wert sich verändern lässt. Variablendeklarationen beginnen mit dem Schlüsselwort **var**. Dann wird zunächst der Variablenname angegeben. Dem Bezeichner folgt ein Doppelpunkt, dem sich der Typ des Bezeichners anschließt.

Beispiel:
```
var Ergebnis: integer;
```

3 Einfache Datentypen

- **Ordinaltypen:** Ordinaltypen sind dadurch gekennzeichnet, dass sie sich durch eine Ordnungsrelation (<, >, = usw.) ordnen lassen.
- **Integer – Typen:** Integer – Typen stellen jeweils Teilmengen der ganzen Zahlen dar und unterscheiden sich in dem von ihnen benötigten Speicherplatz (siehe Tabelle 1).

Typ	Bereich	Format
Integer	-2147483648..2147483647	32 Bit mit Vorzeichen
Cardinal	0..4294967295	32 Bit ohne Vorzeichen
Shortint	-128..127	8 Bit, mit Vorzeichen
Smallint	-32768..32767	16 Bit, mit Vorzeichen
Longint	-2147483648..2147483647	32 Bit, mit Vorzeichen
Int64	-263..263 -1	64 Bit, mit Vorzeichen
Byte	0..255	8 Bit, ohne Vorzeichen
Word	0..65535	16 Bit, ohne Vorzeichen
Longword	0..4294967295	32 Bit, ohne Vorzeichen

- **Zeichentypen:** Variablen vom Typ `char` dienen dazu einzelne Zeichen (Character) aufzunehmen.
- **Boolesche Typen:** Variablen vom Typ `boolean` können nur einen der vordefinierten Werte **`false`** oder **`true`** annehmen.
- **Gleitkommatypen:** Gleitkommatypen stellen eine Teilmenge der reellen Zahlen dar.

Typ	Bereich	Größe in Byte
Single	1.5×10^{-45} .. 3.4×10^{38}	4
Double	5.0×10^{-324}.. 1.7×10^{308}	8
Extended	3.6×10^{-4951}.. 1.1×10^{4932}	10
Real	5.0×10^{-324} .. 1.7×10^{308}	6

- **String – Typen:** Unter einem String versteht man eine Folge von Zeichen, d.h. Zeichenketten.

  Beispiel:

```
var
  str: string;
begin
  str := 'Dies ist ein String';
end;
```

Ein – und Ausgaben sind jeweils Zeichenketten. Für die Konvertierung in Zahlen gibt es entsprechende Standardprozeduren:

```
a := StrToFloat(Edit1.Text)      // wandelt den Text in Edit1 in eine Gleitkommazahl um
b := StrToInt(Edit1.Text)        // wandelt den Text in Edit1 in eine Integerzahl um
Edit1.Text := FloatToStr(a)      // wandelt die Gleitkommazahl a in einen String um und gibt sie in
                                 Edit1 aus
Edit1.Text := IntToStr(b)        // wandelt die Integerzahl b in einen String um und gibt sie in Edit1 aus
```

- **Aufzählungstypen:** Aufzählungstypen definieren eine Menge von Werten mit eindeutiger Reihenfolge, indem einfach die einzelnen Bezeichner dieser Werte aneinander gereiht werden.
  `type` `Typname = (Wert1, ..., Wertn)`
  Beispiel: **`type`** `Wochentag = (Montag, Dienstag, Mittwoch, Donnerstag, Freitag, Samstag, Sonntag);`
- **Teilbereichstypen:** Ein Teilbereich ist eine Untermenge der Werte eines anderen Aufzählungstyps (des so genannten Basistyps). Im obigen Beispiel wäre **`type`** `Arbeitstage = Montag..Freitag` eine Teilmenge der Aufzählung der Wochentage.