



清华大学出版社
TSINGHUA UNIVERSITY PRESS

Delphi程序设计教程



第2章 Delphi语法基础

2.1 保存字与标记符

2.1.1 保存字

2.1.2 标记符

2.1.3 指令符

2.1.4 注释

2.2 数据类型

2.2.1 数值型数据

2.2.2 字符型数据

2.2.3 布尔型数据

2.3 常量与变量

2.3.1 常量

2.3.2 变量

2.4 运算符与体现式

2.4.1 算术运算符与算术体现式

2.4.2 逻辑运算符与布尔体现式

2.4.3 字符串运算符

2.4.4 运算符的优先级

2.5 惯用系统函数与过程

2.5.1 数值运算函数

2.5.2 字符解决函数

2.5.3 日期时间函数

2.5.4 次序类型函数

2.6 语句

2.6.1 语句的基本概念

2.6.2 常见声明语句

2.6.3 赋值语句和程序的次序构造

2.6.4 条件语句和程序的选择构造

2.6.5 循环语句和程序的循环构造

小结

习题2

2.1 保存字与标记符

2.1.1 保存字

保存字（Reserved Word）是由系统规定的、含有特定意义的单词，在编程时不能被重新定义或作其它用。

2.1.2 标记符

标记符是用来表达常量、变量、字段、类型、程序、过程、函数、对象等语法实体的名称的符号。

1. 原则标记符

2. 自定义标记符

3. 限定标记符

2.1.3 指令符

2.2 数据类型

2.2.1 数值型数据

1. 整型

2. 实型

2.2.2 字符数据

1. 字符型

2. 字符串型

2.2.3 布尔型数据

2.3 常量与变量

在程序的执行过程中其值保持不变，称为常量；
在程序的执行过程中其值能够被变化，称为变量。

2.3.1 常量

定义符号常量使用常量阐明语句，其语法格式为：

CONST

〈常量名1〉 = 〈常量值1〉 ；

...

〈常量名n〉 = 〈常量值n〉 ；

3. 类型常量

定义类型常量的语法格式为：

CONST

〈常量名〉 : 〈类型名〉 = 〈常量值〉 ；

2.3.2 变量

1. 变量的声明

变量声明的语法格式为：

VAR

〈变量名1〉 : 〈类型名1〉 ;

...

〈变量名n〉 : 〈类型名n〉 ;

当多个变量含有相似的类型时，能够使用以下的紧凑格式：

VAR 〈变量名1〉 , ..., 〈变量名n〉 : 〈类型名〉 ;

2.4 运算符与体现式

2.4.1 算术运算符与算术体现式

1. 算术运算符

2. 算术体现式

算术体现式的格式为：

〈数值1〉 〈算术运算符1〉 〈数值2〉 [〈算术运算符2〉 〈数值3〉]

2.4.2 逻辑运算符与逻辑体现式

逻辑运算符可分为布尔运算符、位运算符和关系运算符。

1.布尔运算符

布尔运算符只能对两个布尔型操作数进行运算，成果仍为布尔型，即只能为True或者False基本的布尔运算符有4个：NOT、AND、OR和XOR。

2.位运算符

位运算符是对Integer类型操作数的二进制形式的位执行操作。

3.关系运算符

关系运算符用于比较两个同类型量的值。

4.布尔体现式

布尔体现式由布尔运算符和布尔类型的操作数所构成，因此布尔体现式中的操作数能够是任何运算成果为布尔类型的体现式，涉及关系运算体现式和运算成果为布尔类型的函数如Odd(x)、FileExists(x)、等。

2.4.3 字符串运算符

字符串体现式的格式为：

〈字符串体现式〉 + { 〈字符串〉 | 〈字符〉 }

2.4.4 运算符的优先级

除了以上介绍的运算符外，尚有指针运算符、集合运算符、类运算符和取地址运算符等，全部的运算符和算术运算符同样，**都含有优先级的概念。**

优先级	运算符	分类描述
1	@（取地址），NOT，-	一元运算符
2	*, /, DIV, MOD, AND, SHL, SHR	乘除及类型强制转换运算符
3	+, -, OR, XOR	加减运算符
4	=, < >, <, >, < =, > =, in, is	关系、集合成员及类型比较运算符

2.5 惯用系统函数与过程

2.5.1 数学运算函数

1. 绝对值函数

平方根函数的定义：

function Sqrt(x: Extended): Extended;

3. 三角函数

function Sin(x: Extended): Extended;

function Cos(x: Extended): Extended;

function ArcTan(x: Extended): Extended;

4. 整数部分与小数部分

取整数函数的定义：

function Int(x: Extended): Extended;

取小数函数的定义：

function Frac(x: Extended): Extended;

6. 指数函数和对数函数

指数函数的定义：

function Exp(x: Real): Real;

对数函数的定义：

```
function Ln(x: Real): Real;
```

7. 随机函数

随机函数的定义：

```
function Random [ ( Range: Integer) ];
```

8. π 函数

函数定义为：

```
function Pi: Extended;
```

2.5.2 字符解决函数

1. 大小写转换函数

其函数定义分别为：

```
function LowerCase(const S: string): string;
```

```
function UpperCase(const S: string): string;
```

2. 比较先后次序函数

其函数定义分别为：

```
function CompareStr(const S1, S2: string): Integer;
```

```
function CompareText(const S1, S2: string): Integer;
```

3. 合并字符串

其过程定义为：

```
procedure AppendStr(var Dest: string; const S: string);
```

其函数定义为：

```
function Concat(s1 [, s2,..., sn]: string): string;
```

4. 查找字符串函数

其函数定义为：

```
function Pos(Substr: string; S: string): Integer;
```

5. 求字符串长度函数

其函数定义为： `function Length(S): Integer;`

6. 截取子字符串函数

其函数定义为：

```
function Copy(S; Index, Count: Integer): string;
```

7. 删除子字符串过程

其过程定义为：

```
procedure Delete(var S: string; Index, Count: Integer);
```

8. 插入子字符串过程

其过程定义为：

```
procedure Insert(Source: string; var S: string; Index: Integer);
```

9. 数值与字符串转换

(1) 过程Str用来将数值型数据转换成字符串，其过程定义为：

```
procedure Str(X [: Width [: Decimals ]]; var S);
```

(2) 过程Val用来将字符串转换成数值型数据，其过程定义为：

```
procedure Val(S; var V; var Code: Integer);
```

(3) 函数IntToStr用来将整型数据转换成字符串，其函数定义为：

function IntToStr(Value: Integer): string; overload;

(4) 函数FloatToStr用来将实型数据转换成字符串，其函数定义为：

function FloatToStr(Value: Extended): string;

(5) 函数StrToInt用来将数字字符串转换成整型数值，其函数定义为：

function StrToInt(const S: string): Integer;

(6) 函数StrToFloat用来将数字字符串转换成实型数值，其函数定义为：

function StrToFloat(const S: string): Extended;

(7) 函数IntToHex用来将10进制整数转换成或16进制格式，其函数定义为：

function IntToHex(Value: Integer; Digits: Integer): string;

10. ASCII转换为字符

函数Chr用来将ASCII码值转换成字符，其函数定义为：

function Chr(X: Byte): Char;

11. 格式化函数

函数Format用来将字符串按照指定的格式返回，故称格式化函数。其函数定义为：

```
function Format(const Format: string; const Args: array of const): string;
```

类型和格式信息由字符“%”开头，格式以下：

```
%[ <参数位置> :][-[ <所占宽度> ][. <小数位数> ] <类型>
```

2.5.3 日期时间函数

1. 日期与时间函数

函数Now用来返回系统现在的日期和时间，其函数定义为：

```
function Now: TDateTime;
```

2. 日期函数

函数Date用来返回系统现在的日期，其函数定义为：

```
function Date: TDateTime;
```

3. 时间函数

函数Time用来返回系统现在的时间，其函数定义为：

```
function Time: TDateTime;
```

4. 转换函数

(1) 函数DateToStr、TimeToStr、DateTimeToStr能够分别将日期和时间（TdateTime类型）转换成字符串，其函数定义分别为：

```
function DateToStr(Date: TDateTime): string;
```

```
function TimeToStr(Time: TDateTime): string;
```

```
function DateTimeToStr(DateTime: TDateTime): string;
```

(2) 函数EncodeDate与过程DecodeDate能够在TdateTime类型的日期值与整型的年、月、日之间互相转换，其定义分别为：

```
function EncodeDate(Year, Month, Day: Word): TDateTime;
```

```
procedure DecodeDate(Date: TdateTime, var Year, Month, Day: Word);
```


(3) 函数EncodeTime与过程DecodeTime能够在TdateTime类型的时间值与整型的小时、分、秒、毫秒之间互相转换，其定义分别为：

```
function EncodeTime(Hour, Min, Sec, MSec: Word): TDateTime;
```

```
procedure DecodeTime(Time: TDateTime, var Hour, Min, Sec, MSec: Word);
```

5. 星期函数

函数DayOfWeek用来返回日期所对应的星期数，其函数定义为：

```
function DayOfWeek(Date: TDateTime): Integer;
```

6. 日期与时间的格式

函数FormatDateTime用来解决日期与时间的格式。其函数定义为：

```
function FormatDateTime(const Format: String; DateTime: TDateTime): String;
```

2.5.4 次序类型函数

1. 次序类型

2. 次序函数

(1) 序数函数Ord返回数据的序数，其定义为：

```
function Ord(X): Longint;
```

(2) 前趋函数Pred返回数据的前趋值，其定义为：

```
function Pred(X);
```

(3) 后继函数Succ返回数据的后继值，其定义为：

```
function Succ(X);
```

(4) 首序数函数Low返回次序型数据取值集中的第一种值（序数最小），其定义为：

```
function Low(X);
```

(5) 末序数函数High返回次序型数据取值集中的最末一种值（序数最大），其定义为：

```
function High(X);
```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/895110020342011322>