C++ Builder 编程基础

孙永林 编

广东交通职业技术学院

前　言

C++是一门最为流行的编程语言，是计算机专业学生必学的一种专业语言之一。本教材主要以引导初学者从C语言程序设计到融入C++ Builder提供的常用组件去设计Windows应用程序。本教材注重培养C++ Builder初学者，主要介绍程序设计的基本技巧。《C++ Builder编程技术基础教程》是一本工具与程序设计相结合的入门教材，由于内容简单直观，也可作为初学者自学教材。本教材重点介绍C++程序设计基本方法和技巧，每部分内容都有相应的实例，最适合做高职类计算机专业“C++编程技术”课程的教材。

编者

2002年6月修订

目　录

第1章 C++数据类型

开始学习任何一种高级程序语言，要必须知道各种数据类型在该程序语言中如何声明，以及各数据使用的有效范围。这样设计出来的程序执行时，才不会发生因数据太大而发生溢出现象，本章最主要是介绍 C++Builder 的基本数据类型，以及数据在计算机中的存储方式。

1.1 内存单元与变量

1.1.1 内存单元

人类有十个手指头，可以用来做计数。如，由零到十遇到十便进位为 10，共有十种不同的状态，我们称为十进制数。而计算机是利用电子元件来存储数据的，由于电子元件可以显示两种状态，不是开（ON）便是关（OFF），利用 ON/OFF 来表示数据，这种描述太麻烦了，于是改用 0 和 1 来分别表示 OFF 和 ON。仿照十进制数，在计数时，由 0 到 2 遇到 2 便进位以 10 表示，一共有两种不同的状态，我们称为二进制数，简称为位（Bit）。

从上可知，Bit 便成为计算机内存上的最小的存储单元。我们可以利用一连串的 0 或者 1 来代表一个数值或符号。一般我们是以 8Bits 为单位来表示一个数据的大小，我们称为字节（Byte）。Byte 是目前计算机内存中用来定址的最小单位。若要用到内存中的内容便要以 Byte 为单位来传递信息，一个 Byte 可以代表一个数值或符号。它可以表示 256 种符号，如‘A’，‘a’，‘+’，‘0’，‘1’等，若表示数值的话，一个 Byte 可以表示的数值范围为 0~2^8-1，即 0~255。

一部计算机的内存是由许多个连续的内存单元所组成的，这些单元是用来存放程序代码或数据的。为了能够用到每个单元的程序或数据，每个单元都必须指定一个地址，就如同邮局里面一列列的邮政信箱。每个信箱都有不同的编号，用来存放不同的信件。但是内存的容量和邮局的信箱都是有限的，而辅助内存（如硬盘或光盘），容量却可以无限扩充。

通常计算机每次存取数据都是以字（Word）为单位，一个 Word 到底包含多少个 Bit 要视机器的不同而定，有 8 Byte、16 Byte、32 Byte、64 Byte 等。有些机器允许有不同长度的 Word，而有些机器却只能用一种长度。由此可知一部计算机所用一个 Word 的长度愈长，存取数据的速度也就愈快。

Byte 是一个内存地址存储数据大小的单位，我们在描述内存容量大小时，以 Byte 为单位似乎太小了，所以一般常采用千字节（K Bytes），百万字节（M Bytes）以及十亿字节（G Bytes）来表示。其换算公式如下：

1K Bytes（KB）=2^{10} Bytes=1024 Bytes （约 1 千）

1M Bytes（MB）=2^{20} Bytes=1,048,576 Bytes=1024KB （约 100 万）

1G Bytes（GB）=2^{30} Bytes=1,073,741,824 Bytes=1024MB （约 10 亿）

1T Bytes（TB）=2^{40} Bytes=1,099,511,627,776 Bytes=1024GB （约 1 兆）

1.1.2 变量

当执行程序时，必须先将程序和数据加载到计算机的内存（RAM）中才能执行，但是程序中所要的数据是如何放入内存的呢？大多数的高级语言都是使用变量，数据存放在变量名称所对应内存中，在设计程序时将每个数据赋一个变量名称，也就是说在程序中用到该数据的地方，以对应的变量名称替换该数据即可。当程序执行时，计算机便自动在内存中分配一个空的位置来存放该变量值。首先，若要顺利的使用变量，必须知道：

（1）　变量要如何命名；

（2）　使用哪种类型的变量。

（3）　如何赋初值给变量。

1、变量的命名规则

（1）变量名称可以由英文字母大小字（A~Z，a~z）、下划线（_）及数字（0~9）所组成，但第一个符号不可使用数字。

（2）变量不可使用保留字及符号常数。

（3）所用的英文字母大小写其变量是不同的。

（4）英文版不可使用中文变量名称。

2、下列变量名称写法错误

（1）Home Work　（中间不能使用空格）

（2）3_1_ Home　(第一个符号不可用数字)

（3）if　　（if 为 C++系统保留字）

注意：

（1）变量名称中的英文字母大小写是不同的。

例如：SCORE、Score、score 代表三个不同变量。

（2）变量的命名最好具有意义，名称最好和数据有关系，这样不但能提高程序的可读性而且易于分辨。

例如：变量名称 name 与 total 分别代表姓名及总数。

（3）变量名称有多个单字时，中间可以加下划线“_”，以增加变量名称的可读性。

例如：tel_no 代表电话号码，id_no 代表身份证号码。

习惯上，变量名称应与要存储数据的意义相关，以避免产生混淆，例如，存成绩可用 score，存总分可用 total 等。另外，C++Builder 沿用 C 语言的特性，变量必须事先声明后才能使用，否则就算符合变量命名规则，编译程序时仍会出现错误。变量可在程序中赋不同数据值，这与常数是不同的。

另外，按照声明变量的位置，可以划分为全局变量与局部变量，主要判别在于变量存在程序中的可存取的有效范围（scope）。

1.1.3 全局变量

声明在头文件（.h）或不是在函数内部，即属于全局变量。这种变量可供整个单元存取。若想在另外一个单元中共用，则需在另外一个单元最前面加 extern 来声明。

例如： extern int year。

以上说明的整型变量 year 是在另一个单元中说明的，现在引用到本单元中使用，并仍为一个全局变量。

1.1.4 局部变量

若声明在函数内部或某个程序块中（如循环中），即属于局部变量，这类型变量只能在该函数的范围（或块）中存取，在范围以外则无法存取该局部变量。

1.2 C++Builder 数据类型

1.2.1 C++Builder 的基本数据类型

基本数据类型决定了内存中可以存储的数据种类及数据范围。表 1-1 列出 C++基本数据类型、所分配的内存单元数目及许可的数据范围。

表 1-1

数 据 类 型	分配空间（bytes）	范 围 值
char	1	-128~127(字符型)
unsigned char	1	0~255（字符型）
byte	1	0~255（整型）
short	2	-32768~32767（短整型）
unsigned short	2	0~65535（无符号短整型）
long	4	-2,147,483,648~2,147,483,647（长整型）
unsigned long	4	0~4,294,967,295（无符号长整型）
int	4	与 long 相同
unsigned int	4	与 unsigned long 相同
float	4	1.2E-38~3.4E38(单精度实型)
double	8	2.2E-308~1.8E308(双精度实型)
bool	1	true 或 false(逻辑值)

C++Builder 把数值分为带正负号的(signed)与无负号的(unsigned)两种。其中无正负号类型的变量只能存储正值数据，而带正负号者则可以存储正负值数据。表 1-1 中 byte，short，int 与 long 为整型（integer）数据，float 与 double 为浮点型数据（带有小数点的数字）。另外，

在整数前面若加上数字零“0”表示该数值是八进制；若在数值前面加上 0x 表示该数值是十六进制，如下列三者数据值是一样的。

int i= 65　　（十进制）

int j= 0101　（八进制）

int k= 0x41　（十六进制）

至于 char 是用来存放字符数据的，在程序中书写字符符号时必须用单引号将数据括起来，如：

char ch=‘a’（声明 ch 是字符变量并将‘a’放到 ch 字符变量中）

char ch=‘A’

1.2.2　AnsiString 数据类型

在 C++Builder 中，字符数据在程序中是用单引号括起来的，如‘A’、‘B’、‘1’、‘2’等。若一个以上的字符合并在一起就形成字符串（string）数据，字符串数据在程序中是用双引号将字符串括起来，如“C++Builder5.0”、“一切 OK！”、“公元二千年”等都属于字符串。C++Builder 为提高字符串的处理效率，提供了许多对字符串处理的方法。AnsiString 字符串类型就是 C++Builder 所提供的字符串类型，在此我们仅举简单例子说明。

1、AnsiString 字符串类型变量声明

AnsiString strl；// 声明一个字符串变量 strl

2、AnsiString 字符串类型的运算符

字符串变量的数据可以做一些简单的运算，如表 1-2 所示。

表 1-2

运 算 符	说　明	例　子
=	把字符串赋给字符串变量	strl= “C++Builder”;
+	两个字符串的合并	str2=“公元”+“二千年”;

例：AnsiString strl，str2，str3；// 声明三个字符串变量

str1=“公元二千年是”;　str2=“△Y2K”;　//△代表一个空格

str3=str1+str2;　//str3=“公元二千年是△Y2K”

此处的“+”（加号）是将 strl 与 str2 两个字符串数据合并在一起。至于“=”（赋值运算符）会将等号右侧的结果赋给左侧 str3 字符串变量。表 1-3 即是 C++Builder 所提供的特殊字符控制符号。

表 1-3

控 制 符	意 义
'\<数字>'	表示一个 ASCII 值对应的字符。如'\0'表示空字符；'\13'表示回车符
'\n'	换行字符,即光标移到下一行开头处
'\''	表示打出单引号字符
'\"'	表示打出双引号字符
'\\'	表示打出\(反斜线字符)

1.2.3 类型转换

若将一个浮点数赋给一个整数变量，结果将会如何呢？例如：

```
int i;
float j=10.5;
i=j;    //将 10.5 赋给变量 i，结果 i 为 10。
```

由于 i 声明为整数型变量，故 10.5 存入 i 时仍只会存入整数值，小数点部分将被去掉，即 i=10。

上述数据转换是在程序执行时自动处理的，我们也可用强迫转换处理，例如：

```
int i;
float j=10.5;
i=(int)j;
```

强制转换的做法是，在变量前的一对括号内（注意必须用半角）写上欲转换数据的类型名称，C++Builder 会将接在此类型转换后面的值自动转换成整数后再赋给前面的变量。

1.2.4 变量的初值

当使用下列语句声明一个变量时，C++Builder 会自动在内存中保留一个位置来存放变量的数据值，下面语句只是声明变量名但并未分配数据值给该变量，

```
int i;
```

上述语句只声明了一个整型变量 i，但未赋初值给变量 i，因此，i 值可能是任意值（并不是 0，可能是正值，负值或任意数）。一般而言，在声明时也可以同时赋变量的初值，其写法如下：

```
int i=100;    （声明 i 为整型变量，并设置其初值为 100）
float j=200.5;（声明 j 为浮点型变量，并设置其初值为 200.5）
char ch='a' ;（声明 ch 为字符型变量，并设置其初值为 a）
```

1.3 C++Builder 运算

程序的表达式用来指定数据做什么运算，如+(加)、-（减）、*（乘）、/（除）等表达式。

表 1-4 到表 1-8 列出 C++常用的表达式运算符号：

1.3.1 运算符

1、算术运算符

表 1-4

运 算 符	意 义	实 例
+	相 加	i=j+k
-	相 减	i=j-k
*	相 乘	i=j*k
/	相 除	i=j/k
%	取余数	i=j%k

2、赋值运算符

表 1-5

运 算 符	意 义	实 例
=	赋值	i=5
+=	相加后再赋值	i+=5
-=	相减后再赋值	i-=5
*=	相乘后再赋值	i*=5
/=	相除后再赋值	i/=5
%=	余数除法后再赋值	i%=5
&=	作 AND 运算后再赋值	i&=5
|=	作 OR 运算后再赋值	i|=5

3、逻辑运算符

表 1-6

运 算 符	意 义	实 例
&&	逻辑 AND 运算	if(i&&j)
::	逻辑 OR 运算	if(I::j)
!	逻辑 NOT 运算	If(!i)

4、关系(比较)运算符

表 1-7

运 算 符	意 义	实 例
==	相等于	if(i==5)
!=	不等于	if(i!=5)
<	小于	if(i<5)

>	大于	if(i>5)
<=	小于等于	if(i<=5)
>=	大于等于	if(i>=5)

5、单目运算符

表 1-8

运 算 符	意 义	实 例
*	指针表达式	int i=*j
&	地址表达式	int* I=&j
~	NOT 运算	i&=~0x00;
!	逻辑 NOT 运算	if(!true)
++	递增表达式	i++;(与 i=i+1 同)
--	递减表达式	i--;(与 i=i-1 同)

6、类别及结构体成员运算符

表 1-9

运 算 符	意 义	实 例
::	范围表达式	MyClass::Getdata( );
->	间接成员表达式	myClass->Getdata( );
.	直接成员表达式	MyClass.Getdata( );

1.3.2 表达式

1、算术运算符表达式

算术运算符用来执行一般的数学运算，包括+(加)、-（减）、×（乘）、/（除）和%（取余数），其用法举例如下：

```
int i，j，k;
i=16;
j=i/3;    //j=5
k=i%3; //k=1
```

j 与 k 均为整数变量，故 i/3 运算结果仍为整型，其结果为 5（j=5）。另外，%为求余，故 k=1。上例中使用“//”为注释符号，是给程序设计者阅读用，与程序中的语句无直接关连，C++Builder 编译执行时，也不会执行这些注释。

2、复合赋值运算符表达式

程序中 i=i+5 可以用较简洁方式表示如下：

```
i+=5;  //比原式少了一个符号 i
```

若等号两侧有相同名称的变量，做运算时可采用**复合赋值表达式来表示。**

3、比较运算符表达式

关系(比较)表达式用于数据间的比较，比较的结果可为逻辑值 true 或 false，在 C++Builder 中是以 0 来代表 false（不成立），而以非零值（0 以外其他值）代表 true。

```
int i=10, j=20;
(i > j )              //false
(i <= j )             // true
(i +10>= j )          // true，同（(i +10）>= j）
(i +20>= j*2 )        // true，同((i +20)>=(j*2))
```

4、递增与递减表达式

i=i+1 可表示为 i++，i=i-1 可表示为 i--，++与--分别表示递增与递减表达式，此表达式为变量值加 1 或减 1。递增与递减可按照++在变量之前或之后分为前递增与后递增，--也可分前递减或后递减。其主要差别在于变量加 1 或减 1 的顺序，例如：

```
int i，j, k, m;
i= j=10;              //与 j=10；  i= 10 语句相同
k=++i;                // i 值先加 1，再将结果赋值给 k
m=j++;                //j 值先赋给 m，然后 j 再加 1
k=++i 语句相当于 { i=i+1;    // i=11
                   k=i;      // k=11
m=j++语句相当于 { m=j;       // j=10；m=10
                  j=j+1;     // j=11
```

再看另一个例子，

```
int i，j, k, m;
i= j=10;
k=--i;                // i=9，k=9
m=j--;                // m=10，j=9
k=--i 语句相当于 { i=i-1;    // i=9
                   k=i;      // k=9
m=j--语句相当于 { m=j;       // j=10；m=10
                  j=j-1;     // j=9
```

1.4　常用系统函数介绍

当我们在写程序时，将一段具有某种特定功能的语句块或是重复出现的程序区段单独抽出来，编写成一个独立的程序单元，并给予特定名称，以方便其他程序调用，我们将这类的程序单元称为“函数”。函数具有下面特点：

• 函数是应用程序的一部分，一般函数是不能单独执行的。

• 函数拥有自己的名称，在一个单元文件中不能同时拥有两个相同名称的函数。

• 函数内的变量，除非有特别声明，否则都是局部变量，也就是说 C++Builder 在不同函数内允许使用相同的变量名称。

• 函数具有特定功能。

• 函数具有模块化。

按照其函数的特点，可分成下面三大类型：

（1）系统函数：是 C++Builder 系统所提供的，使用时只要在程序前面加入正确头文件就可以调用。

（2）事件函数：是配合对象使用，函数内的程序代码是由设计者视对象执行情况而写入的。

（3）一般函数：是由程序设计者自己编写，自己命名。

C++Builder 提供的系统函数很多，通常可以从集成环境中的帮助菜单中查寻。为了便于程序设计，下面列出一些常用的系统函数。

1.4.1 数值函数

数值函数的头文件是：

#include <math.h>

常用数值函数如表 1-10。

表 1-10

函数名称	功能说明
fabs	语法：float fabs(float x); 说明：返回 x 绝对值
sprt	语法：float sqrt(float x); 说明：返回 x 的开平方值
hypot	语法：float hypot(float x,float y); 说明：返回 x^2+y^2 的开平方值

sin
cos
tan
	语法：float sin(float x);
	float cos(float x);
	float tan(float x);
	说明：返回三角函数值，x 以弧度为单位。

asin
acos
atan
	语法：float asin(float x);
	float acos(float x);
	float atan(float x);
	说明：返回反三角函数值，asin 与 acos 的 x 范围为-1 到 1。

exp	语法：**float exp(float x);** **说明：返回 e^x 值**
log	语法：**float log(float x);** 说明：**返回 ln(x)值**
log10	语法：float log10(float x); 说明：返回以 10 为底 $\log_{10}(x)$值
ceil	语法：float ceil(float x); 说明：返回不小于 x 的最小整数
Floor	语法：float floor(float x); 说明：返回不大于 x 的最大整数

1.4.2 字符串常用函数

1、传统格式字符串函数

字符串常用函数的头文件如下：

#include <string.h>

声明字符串的方法有：char str[20];

char *str1,*str2;

字符串常用函数见表 1-11。

表 1-11

函数名称	功能说明
Strlen	语法：size_t strlen(const char *s); 说明：返回字符串的长度，不包括 Null 字符
Strcpy	语法：char *strcpy(char *dest,const char *src); 说明：将 src 字符串拷贝到 dest 数组
strncpy	语法：char *strncpy(char *dest,const char *src,size_t maxlen); 说明：将 src 前面 maxlen 个字符拷贝到 dest 字符串
Strcat	语法：char *strcat(char *dest,const char *src); 说明：将 src 字符串连接到 dest 后面
Strncat	语法：char *strcat(char *dest,const char *src,size_t maxlen); 说明：将 src 前面 maxlen 个字符连接到 dest 字符串后面
Strcmp	语法：char *strcmp(const char *s1,const char *s2); 说明：字符串 s1 和字符串 s2 由左到右逐字比较
strncmp	语法：char *strcmp(const char *s1,const char *s2,size_t maxlen); 说明：比较字符串 s1 和 s2 最前面的 maxlen 个字符

例 1-1：练习求字符串长度并将该字符串颠倒显示。

```
#include <stdio.h>
#include <string.h>
//---------------------------------------------------------------------------
void __fastcall TForm1::FormPaint(TObject *Sender)
{
  int i,n,k;
  char *s;
  char buff[20];
  char str1[]="Happy";
  Canvas->TextOut(10,20,"    ===    字符串颠倒    ===");
  // 利用数组名称显示字符串
  sprintf(buff,"原字符串：%s" ,str1);
  Canvas->TextOut(20,60,buff);
  //求字符串长度
  n=strlen(str1);
  Canvas->TextOut(20,80,"该字符串长度="+IntToStr(n));
  //利用指针显示字符串
  s=str1;  //将指针指向数组第一个元素
  for (k= 0 ; k<=n ; k++)
  {
    Canvas->TextOut(20+12*k,120,*(s+k));
  }
  //颠倒字符串
  for (k=n-1,i=0;k>=0; k--,i++)
  {
    sprintf(buff ,"%c",*(s+k));
    Canvas->TextOut(20+12*i,140,buff);
  }
}
//---------------------------------------------------------------------------
```

2、AnsiString 格式字符串函数

在 C++Builder 标准类别库中提供了一个 AnsiString 字符串类型，这种类型可以简化传统 C 语言使用字符来处理字符串的方式。为了让程序书写方便，C++Builder 在 SYSDEFS.H 头文件中定义为：

typedef AnsiString String;

在书写程序时，声明 AnsiString 的对象时，可以直接使用 String 来声明。AnsiString 所声明的变量，可以直接接收 char、char*、int、float 或 double 的数据。

例如：

```
AnsiString s1='w';
AnsiString s1=52;              //s1="52"
AnsiString s1=12.54;           //s1="12.54"
AnsiString s1="Good";
```

AnsiString 类型包含有比传统 C 使用更为方便的成员函数。成员函数的使用格式为：

< AnsiString 字符串变量>.<成员函数名>（参数表）

例如：String s1;

```
char *str;
str=s1.c_str()    // c_str()是一个把 AnsiString 格式的串转换成传统格式的成员函数
```

下面表 1-12 是 AnsiString 的常用成员函数。

表 1-12

成员函数名	语法与功能说明
c_str	语法：char* fastcall c_str() const 功能：返回字符串数据的指针。
Delete	语法：void _fastcall Delete(int index,int const); 功能：由字符串 index（起始位为 1）处开始删除 const 个字符。
Insert	语法：void _fastcall Insert(const AnsiString& str,int index); 功能：由 index（起始位为 1）处开始插入字符串 str 到原字符串中。
IsEmpty	语法：void _fastcall IsEmpty() const; 功能：返回字符串是否为空字符串。
Length	语法：void _fastcall Length() const; 功能：返回 AnsiString 的长度。
LowerCase	语法：AnsiString _fastcall LowerCase() const; 功能：将字符串中的大写字母改成小写。
UpperCase	语法：AnsiString _fastcall UpperCase() const; 功能：将字符串中的小写字母改成大写。