

第2章 基本数据类型、运算符和体现式（课时4节）

2.1 数据类型

2.2 C++的字符集

2.3 变量和常量

2.4 运算符和体现式

2.5 类型的转换

2.6 数制

2.7 位运算

2.8 sizeof运算

2.1 数据类型

程序处理的对象是数据，数据分为变量和常量。

数据类型是指一定的数据在计算机内部的表达方式。

定义数据类型的作用：

- (1) 决定数据分配存储空间的大小。
- (2) 要求数据所能进行的操作。

C++中的数据类型有内部数据类型和外部数据类型两大类，内部数据类型主要有：整数类型(int)、单精度浮点(float)、双精度浮点(double)、字符类型(char)、布尔类型(bool)和空类型(void)；

外部数据类型是在内部类型以及其他外部数据类型的基础上定义的数据类型。

一、整数类型

整数类型（整型）用于描述整数。整数的存储空间和数据范围见表1。

表1 整数的存储空间和数据范围

类型	字节数	范围	备注
短整型short int	2	$-32768(2^{15})\sim 32767(2^{15}-1)$	简称short
整型int	4	$-2^{31}\sim(2^{31}-1)$	
无符号整型unsigned int	4	$0\sim(2^{32}-1)$	简称unsigned
长整型long int	4	$-2^{31}\sim(2^{31}-1)$	简称long

二、实数类型（浮点型）

实数类型（实型）用于描述实数，根据存储空间的大小，实型数据分为单精度浮点和双精度浮点。

在计算机内部，实数用科学记数法表达，即把数据表达成 $m \times 2^e$ ，其中 m 称为尾数， e 称为阶码。

实型数据在计算机中存储的是尾数和阶码两部分。实数的存储空间和数据范围见表2。

表2 实数的存储空间和数据范围

类型	字节数	范围	备注
float	4	$-3.4 \times 10^{-38} \sim 3.4 \times 10^{38}$	7位有效位
double	8	$-1.7 \times 10^{-308} \sim 1.7 \times 10^{38}$	15位有效位

三、字符类型

字符类型用于描述单个字符数据。字符采用ASCII编码（国际原则码）表达，在存储时占用一种字节。字符数据的存储空间和数据范围见表2。

表2 字符数据的存储空间和数据范围

类型	字节数	范围
char	1	-128~127
有符号字符型signed char	1	-128~127
无符号字符型unsigned char	1	0~255

四、布尔类型

布尔类型用于描述整数1和0的两个值。整数表达逻辑的真（true）和假（false）。

五、空类型

空类型用于描述函数没有返回值、函数没有参数及无类型指针。

2.3 C++的字符集

1、字符集

(1) 大小写字母: A~Z a~z

(2) 数字: 0~9

(3) 运算符: + - * / % = ! & | ~ ^ < >

(4) 标点符号: ; : , .

(5) 特殊符号: () [] { } ‘ “ * ? 下划线_ 空格

2、标识符

由字母、数字和下划线三种字符构成，且必须以字母或下划线开头。

例：下列是正当标识符：

A2、 student、 area_of_circle、 num、 _dd、 Int

下列是不正当标识符：

2A、 A-B、 area of circle、 M.D、 int

3、关键字

关键字是有特定含义的、专用的标识符。

如int、char、break、for、define等。

注意：

关键字是不能作变量名、符号常量名、函数名、类名等。

2.2 常量与常量符号定义

常量是在程序运营过程中其值不能变化的量。

常量有值常量和符号常量两种。

值常量：以字面值的形式直接出现在程序中。

符号常量：以标识符的形式表达。

(一) 值常量 (字面常量)

1. 整型常量

在C++中，整型常量可用十进制、八进制和十六进制表示

进制	形式	实例	错例
十进制	以数字1~9开始	-3, +5, 12	045
八进制	以数字0开始 (由0~7之间数字构成)	023, -010	078
十六进制	以0x或0X开始 (由0~9及A~F构成)	0XAF, -0X51	0XFG

2. 实型常量

实型常量的表达形式：小数形式和指数形式。

小数形式：由正负号、数字和小数点构成，

如： 1.25、-1.25、17. 和 .1

指数形式：由尾数、指数符号e(或E)及指数构成，其中指数必须是十进制整数。

如： 1.25e-5、+1e10、-1.25e5。

注意：

e前面不能没有数字，e背面的数字必须为整数，也不能加园括号。

如： 1E(-3)、E-5、1e2.1都是错误的。

3. 字符常量

字符常量:用单引号括起来的一种字符。

如: 'A'、'0'、'?'。

转义字符常量:以“\” (反斜杠) 开头的用单引号括起来的字符序列。

如: '\n'、'\t'、'\x61'等。常用的转义字符见p27页。

注意:

'\0'表达ASCII码为0的字符,它是一种空字符,

'\0'与'0'的区别,后者表达的是ASCII码为48的数字字符。

字符型数据在内存中以ASCII码存储,故整型数据和字符型数据在一定范围内能够通用。

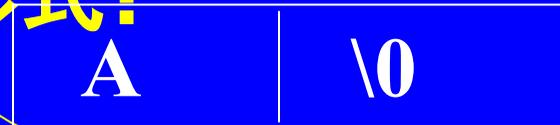
如: `x = 'a' + 2;` //x的值是99

4. 字符串常量

字符串常量是用双引号括起来的字符序列，
系统在字符串常量最终自动加字符串结束符号'\0'。

例如：

字符串“A”的存储形式：



字符'A'的存储形式：



内存中实际存储的是这些字符的ASCII码

(二). 符号常量

符号常量是以标识符形式出现的常量，

符号常量的作用：便于程序阅读和修改。

C++中用define命令或关键字const定义符号常量,形式如下：

(1) #define 标识符 常量值

(2) const 数据类型 标识符 = 常量值;

注意：

(1) 习惯上符号常量名大写，而变量名小写，以示区别。

(2) 符号常量虽然用标识符来标识，但本质上是常量，具有常量值不能变化的性质。

2.2 变量

1.变量的定义

变量是在程序运营过程中其值可变的量。

定义方式如下：**数据类型 变量名表；**

C++要求：

(1) 程序中全部变量必须先定义后使用。

(2) 变量名是顾客书写的标识符。

定义变量的作用：可及时发觉录入错误，可限制非法运算。

C++中变量可随使用，随时定义。

例如：

```
int j,k;      //定义j和k都为整型变量，内存中变量j,k各占4个字节
float f1 ;    //定义f1为单精度实型变量，内存中变量f1占4个字节
char c1;     //定义c1为字符型变量，内存中变量c1占1个字节
```

2.变量的初始化

变量必须先有拟定的值后才干参加多种相应的操作。

变量获取值的途径有三：

- (1) 经过输入语句输入 (如: `cin>>a;`)
- (2) 经过赋值语句赋值 (如: `a=3;`)
- (3) 经过初始化方式赋初值 (如: `int k=3;`)

所谓初始化是指在定义变量的同步指定变量的初值。

初始化的形式有二：

- (1) 数据类型 变量名=体现式；
- (2) 数据类型 变量名 (体现式) ；

例如：

```
int k=3,m=3, n(3);
```

(×) `int k=m=n=3;` //不允许为多种变量同步初始化

3.变量引用

所谓**引用**即给一种已知变量起个别名，系统不为引用另外开辟内存，而是与原变量共享同一段内存，引用好比是一种虚拟的变量，

申明形式为：**数据类型 &引用名=已定义的变量**

如：

```
int a=3, &b=a; //定义b为a的引用
```

```
b=5;
```

```
//a=5
```

2.3 运算符和体现式

运算符具有三个属性：

目数：指运算符要求的操作数的个数，涉及单目、双目和三目，一般目数越少，优先级越高。

优先级：指运算的先后顺序，共分15个级别。(见p28表1.2.4)

结合性：指相同优先级运算符连续出现时的计算顺序，分从左往右算（**左结合**）和从右往左算（**右结合**）两种，除尤其申明外，皆指左结合。

体现式：由常量、变量、函数调用和运算符按一定规律组合在一起构成的式子。

如： $a+14/(3+2)-\text{sqrt}(17)$

体现式的值：按照运算规则进行运算而获取的值。

2.3.1 运算符

1. 算术运算

+ (加法或取正)

- (减法或取负)

* (乘法)

/ (除法)

% (求余数)

++ (自增)

-- (自减)

其中： ++和--的结合性则是自右向左

优先级：(从高到低) (++、--、+ (取正)、- (取负))

(*、/、%)

(+ (加法)、- (减法))

阐明:

(1) 在/(除法)运算中, 当两个操作数都是整数时, 商也为整数, 小数部分一律舍去。

如: $1/2$ 的值是 0 ; $1.0/2$ 的值是 0.5

(2) % (求余运算) 要求两个操作数都是整型, 成果为整除后余下的整余数。

如: $1\%2$ 的值是1; $4\%2$ 的值是0

(3) 自增、自减运算符(++、--)

`++i` , `--i` (在使用 `i` 之前,先使`i`的值加1、减1)

`i++` , `i--` (在使用 `i` 之后,使`i`的值加1、减1)

如: `int j=5,k=5,m,n;`

`m=j++;` // `m=5, j=6`

`n=++k;` // `n=6,k=6`

注意: 自增、自减运算符只能用于变量,而不能用于常量和体现式。

如: 下列表达是错误的 `5++`、`(a+b)++`

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/975323140241011341>