
第二章 数据类型、运算符与表达式

2.1 数据类型

前言：数据类型的作用：

☐ 指出系统应为数据分配多大的存储空间；

☐ 规定了数据所能进行的操作；

数据类型的分类（P43）：

☐ 基本类型：用户可直接使用，如整型、浮点型、字符型等；

☐ 派生类型（构造类型）：在基本类型的基础上，由系统或用户自行定义的如：数组、结构体、指针型等；

一. 整型数据（P44）

1. 基本整型

(1) 类型名称：int

(2) 存储形式：二进制，2（或 4）个字节，原码、反码与补码（P44）

(3) 取值范围：P45 $-32768 \sim +32767$ ($-2^{15} \sim 2^{15}-1$)

(4) 应用实例：求两个整数的积

```
# include "iostream.h"
```

```
void main( )
```

```
{
```

```
    int a,b;
```

```
    cout<<"请输入两个整数:";    print f ("请输入两个整数:");
```

学习实践部祝大家考试顺利(*^o^*)

```
cin>>a>>b;————scanf ("%d %d ", &a, &b);

cout<<"积="<< a*b <<endl;————printf ("积=%d \n", a*b );

}
```

2. 拓展整型（拓展数据的处理范围）： P44-P46

long: 扩大数值所占的字节数;

(short: 缩短数值所占的字节数;)

unsigned: 无符号位;

(signed: 有符号位, 缺省方式;)

二. 浮点型（实型）数据（P49）

1. 浮点数的两种表示形式：十进制小数、指数形式

2. 单精度型

(1) 类型名称: float

(2) 表示方法: P50

(3) 存储形式: P50, 4 个字节, (规范化的) 指数形式

(4) 取值范围: P50 $-3.4 \times 10^{-38} \sim 3.4 \times 10^{38}$, 7 位有效数字

(5) 应用实例: 输入圆的半径, 求面积

```
# include "iostream.h"
```

```
void main( )
```

```
{
```

```
float r, s;
```

```
cout<<"输入圆半径:";————printf ("输入圆半径:");
```

```
cin>>r;————scanf ("%f ", &r);
```

```
s=3.14*r*r;
```

学习实践部祝大家考试顺利(*^o^*)

```
cout<<"面积="<<s<<endl;—————printf("面积=%f\n", s);  
}
```

3. 双精度型

- (1) 类型名称: double
- (2) 存储形式: 8 个字节
- (3) 取值范围: P50

4. 浮点型数据的舍入误差: P50

三. 字符型数据 (P47)

1. 字符数据

- (1) 类型名称: char
- (2) 取值范围: ASCII 码字符集中的字符 (无中文)
- (3) 表示方法: P39

☒ 用单引号作定界符, 例: 'a'、'+'、'1'、'\$'

☒ 转义字符: 将双引号中的反斜杠 (\) 后面的字符转换成另外的含义 (P40)

例 1: 八进制与十六进制数所代表的字符 (P40 表 3.1)

```
# include "stdio.h"  
  
void main( )  
{  
  
    printf("\061\x41\n");—————cout<<"\061\x41\n";  
  
}
```

运行结果: 1 A (加回车)

- (4) 存储形式: (P39), 1 个字节, 用于存储该字符的相应的 ASCII 代码

例: 字符 'a' 的存储方式如下:

学习实践部祝大家考试顺利(*^o^*)

011000001 (ASCII 代码为 97)

(5) 运算操作：字符型数据和整型数据之间可以通用

☐ 算术运算：'a'+'b' 97+98

☐ 关系运算：'a'<'b' 97<98

(6) 应用实例：P55 例 3.3

2. 字符串数据

(1) 字符串的表示：用双引号作定界符，例："aBc"、"567"、"\$1.234"、"\$"

(2) 字符串的存储：P40，附加字符串结束标志"\0"（空操作字符） 例：字符串 "string"、"a" 及字符 'a' 的存储方式如下：

"string"	"a"	'a'
s t r i n g \0	a \0	a

(3) 存储形式：存储一个字串的字节数等于字符个数加 1，即字符串长度再加 1

(4) 字符串的数据类型：需要通过字符数组来存放字符串

2.2 常量与变量

例： X=5 x—变量名

变量 常量 5—变量值

X=X+3

一. 常量 (P39)

1. 整型常量：189（十进制）、0177（八进制）、0x1FF（十六进制）

2. 实型常量：-1.25（小数形式）、1.25E-5（指数形式）

3. 字符常量：'a'（普通字符）、'\n'（转义字符）

4. 字符串常量："123"、"a"

学习实践部祝大家考试顺利(*^o^*)

-
1. 运算符及运算规则: P91
 2. 运算结果: 为逻辑型数据, 其中: 真 (true) 为 1, 假 (false) 为 0
 3. 例:

63<54	0
5==3	0
3>=3	1
'a'>'A'	1
'男'>= '女'	0
true!=false	1

四. 逻辑运算符和表达式

1. 运算符及运算规则: P93
2. 运算结果: 真 (true) 为 1, 假 (false) 为 0
3. 逻辑表达式: 可包含其它多种运算 (如关系运算、算术运算), 优先级的规则: P93
4. 例: 用逻辑表达式描述下列条件

☒ 1)x 是 3 的倍数

`x%3==0`

☒ 2)x 是偶数

`x%2==0`

逻辑表达式: 用作程序语句中的<条件>

例: 判断奇偶数

```
cin>>x;
```

```
if(x%2==0)
```

```
    cout<<x<<"为偶数";
```

```
else
```

```
cout<<x<<"为奇数";
```

3)x 是 3 的倍数且 x 是偶数

```
(x%3==0) && (x%2==0)
```

4) $100 \leq x < 200$

```
(x>=100) && (x<200)
```

5)x 等于 2 或 8

```
(x==2) | (x==8)    ✓
```

```
(x=2) | (x=8)       ✗
```

```
(x=2|8)             ✗
```

五. 条件运算符和表达式

1. 运算格式: 条件? 表达式 1:表达式 2 (P97)

2. 运算规则及注释: P98

3. 例: 利用条件表达式求任意 3 个数中最大的一个数。

```
#include "iostream.h"
```

```
void main( )
```

```
{
```

```
    float  x,y,z,max;
```

```
    cout<<"输入 3 个数: "<<endl;
```

```
    cin>>x>>y>>z;
```

```
    max=x>y?x:y;
```

```
    max=max>z?max:z;
```

```
    cout<<"最大的数据="<<max<<endl;
```

学习实践部祝大家考试顺利(*^o^*)

}

六. 标准库函数 (P384)

1. 常用数学函数

例: `cos(0)` 1

`fabs(-10.7)` 10.7

`pow(2,3)` 8.0

`sqrt(9)` 3.0

函数的原型 (定义、声明) 与函数的调用: P385、P8 例 1.3

说明 (P384): `# include <math.h>` 或 `# include "math.h"`

2. 常用输入输出函数

例: `scanf`、`printf`、`getchar`、`putchar`

说明 (P387): `# include <stdio.h>` 或 `# include "stdio.h"`

第三章 C/C++程序的流程控制

3.1 程序中的数据输入输出

前言

☐ 数据输入输出的概念: P67

☐ (头文件) 调用的方式 (P72):

☐ 1) C 的库函数: `# include "stdio.h"` ;

学习实践部祝大家考试顺利(*^o^*)

包括: `printf`、`scanf`、`putchar`、`getchar`

图 2)C++的指令: `# include "iostream.h"` ;

包括: `cout`、`cin`

一. 格式输出函数 P68

1. 格式: `printf` (格式字符串, 输出参数表) (P69)
2. 功能: 向终端 (如屏幕) 输出若干个任意类型的数据
3. 说明: P69
4. 例 1: P69

例 2. `int a=3;`

`char b='y';`

`printf("a=%d, b=%c\n", a, b);`

结果: `a=3, b=y`

5. 格式字符: P70-P75 (可参考查阅)

(1) `%d` : P70 (`%md` 域宽)

(2) `%c` : P70 (`%mc`)

(3) `%s` : P71 (`%ms`)

(4) `%f` : P71 (`%m.nf`)

(5) `%e` : P72

二. 格式输入函数 P75

1. 格式: `scanf` (格式字符串, 输入参数表)
2. 功能: 从终端 (如键盘) 输入若干个任意类型的数据
3. 注意: P76-P78
4. 格式字符: P76 (类同 `printf` 函数)

学习实践部祝大家考试顺利(*^o^*)

例 1: `scanf ("%2d %3d" , &a , &b);`

输入: 12 3456

结果: a 的值 12, b 的值 345

三. 字符输出函数 P78

1. 格式: `putchar (表达式)`
2. 功能: 向终端 (如屏幕) 输出一个字符
3. 说明: 表达式可以是字符型 (或整型) 数据, 可包括常量、变量、转义字符等
4. 例 1: `putchar ('a');`

`char a='$';`

`putchar (a);` 等价于 `printf ("%c", a);`

`putchar ('\n');`

四. 字符输入函数 P79

1. 格式: `getchar ()`
2. 功能: 从终端 (如键盘) 输入一个字符
3. 说明: 函数无需参数, 接收的字符可以赋给字符型 (或整型) 变量
4. 例 1: `# include "iostream.h"`

`# include "stdio.h"`

`void main()`

`{`

`char a;`

`... ..`

`printf ("是否要继续(Y/N):");`

学习实践部祝大家考试顺利(*^o^*)

```

a=getchar( );    等价于    scanf ("%c", &a);

if(a=='y' | |a=='Y' )

    cout<<"继续"<<endl;

else

    cout<<"退出"<<endl;

}

```

五. 格式输出指令

1. 格式: `cout<<格式控制符<<数据项`
2. 例 1: `cout<<'a'<<setw(4)<<'b';` 结果: a b
3. 功能: `setw()`用于设置（其后的一个）数据项的输出宽度
4. 说明: `# include <iomanip.h>` 或 `# include "iomanip.h"`

3.2 流程控制与程序结构

一. 结构化程序设计的三种控制结构

1. 顺序结构: 按语句先后顺序, 依次执行一次.
2. 分支结构: 根据条件成立与否, 选择其中的部分执行.

两分支: `if` 语句

多分支: `switch` 语句

3. 循环结构: 某一段程序可以多次重复执行.

二. 顺序结构程序

例 1: `/*从键盘输入一个大写字母, 将其改用小写字母输出。*/ (P82)`

```
# include "stdio.h"
```

```
void main( )
```

```

{

char a;

printf ("请输入大写字母:");

a=getchar( );

printf ("%c\n",a+32);————cout<<(char)(a+32)<<endl;\\ 数据类型的强制转换

}

```

3.3 分支（选择）结构程序设计

一. 两分支语句 P89

1. 格 式：if（条件）

语句组 1;

else

语句组 2;

3.例 1: /*判断奇偶*/

```
# include <iostream.h>
```

```
void main( )
```

```
{
```

```
int x;
```

```
cout<<"请输入数据: ";
```

```
cin>>x;
```

```
if (x%2==0)
```

```
    cout<<x<<"为偶数."<<endl    ;
```

```
else
```

```
    cout<<x<<"为奇数."<<endl    ;
```

```
}
```

学习实践部祝大家考试顺利(*^o^*)

例 2: /*从键盘输入一个大写字母, 将其改用小写字母输出。*/

```
# include "iostream.h"

# include "stdio.h"

void main(    )

{

char a;

printf("请输入大写字母:");

a=getchar(    );

if(a>='A'  && a<='Z')

    cout<<char(a+32)<<endl    ;

else

    cout<<"输入错误! "<<endl    ;

}
```

4. 复合语句

图 P90 说明(4)

图 P87 例 4.2: 二值互换

二. 简单分支

1. 格 式: if (条件) 语句组;
2. 流程图: (P89)
3. 例 1: /*求数据的绝对值*/

```
# include "stdio.h"

void main(    )

{

float x    ;
```

学习实践部祝大家考试顺利(*^o^*)

```
printf("请输入数据:");  
  
scanf("%f", &x);  
  
if (x<0)  
  
    x=-x;  
  
printf("绝对值: %8.3f \n", x );  
  
}
```

三. 分支的嵌套 (P99)

1. 格式: 可多种 (P99-P100)

2. 规则:

- ☒ 1) 在每一嵌套层中必须使 **if--else** 对应, 互相匹配, 成对使用;
- ☒ 2) 可以多层嵌套, 但其层次必须清楚, 外层对内层完全嵌套, 不得交叉;
- ☒ 3) 为使嵌套层次清晰, 易于查错、修改, 建议采取缩进式的书写形式;
- ☒ 4) 任何嵌套层, 如果包含多个操作语句, 均必须使用“{ }”将它们括起来, 使之成为一个复合语句。

3. 注意事项及错误分析: P100-102 例 4.5 的内容及分析

4. 例 1: 输入三个数, 输出其中最大的.

四. 多分支语句 P103

1. 格式: switch (表达式)

```
{  
  
    case 常量表达式 1: 语句 1;  
  
    case 常量表达式 2: 语句 2;  
  
    ... ..  
  
    case 常量表达式 n: 语句 n;
```

default: 语句 n+1;

}

2. 例 1: P102

3. 说明: P103 (1)~(5)

4. 例 2: /*根据输入的数字来输出对应的星期几的英文单词。*/

```
# include <iostream.h>
```

```
void main( )
```

```
{
```

```
    int x;
```

```
    cout<<"输入数据(1~7): ";
```

```
    cin>>x;
```

```
    switch (x)
```

```
{
```

```
    case 1: cout<<"Mon"<<endl ;
```

```
    case 2: cout<<"Tue"<<endl ;
```

```
    case 3: cout<<"Wed"<<endl ;
```

```
    case 4: cout<<"Thur"<<endl ;
```

```
    case 5: cout<<"Fri"<<endl ;
```

```
    case 6: cout<<"Sat"<<endl ;
```

```
    case 7: cout<<"Sun"<<endl ;
```

```
    default: cout<<"Error number"<<endl ;
```

```
}
```

```
}
```

运行结果 ×

5. 说明: P104 (6)

学习实践部祝大家考试顺利(*^o^*)

break 语句：通常加在每一个 **case** 分支后，使流程在执行了任何一个 **case** 分支后，即刻跳出 **switch** 结构。

6. 说明：在写 **case** 后面的常量表达式的值时，应注意数据类型。

如：例 1 中为字符型，则 **case 'A':**

例 2 中为整型，则 **case 1:**

7. 扩展格式：

switch (表达式)

{

case 常量表达式 1: 语句组 1;

[**break**;

case 常量表达式 2: 语句组 2;

[**break**;

... ..

case 常量表达式 n: 语句组 n;

[**break**;

[**default** t: 语句组 n+1;

}

8. 说 明：P104 (8)

9. 注 释：

☐ P104 说明 (7) ：每个语句组可以不必用花括号括起来；

☐ 每一个 **case** 分支的入口只能是一个常量，而不能是<条件>；

例 3： P109 例 4.10

3.4 循环结构程序设计

一. **while** 语句（当型循环）

1. 格 式：**while** (条件) 语句组;

学习实践部祝大家考试顺利(*^o^*)

2. 流程图：（P116）

3. 功能及特点：P115

4. 例 1：/*输出 1-10 的立方值*/

```
# include "ios tream.h"
```

```
# include "math.h"
```

```
void main( )
```

```
{
```

```
    int i,s;
```

```
    i=1; //循环变量赋初值
```

```
    while ( i<=10 ) //循环条件判断
```

```
    {
```

```
        s=pow(i,3);
```

```
        cout<<i<<"的立方="<<s<<endl;
```

```
        i++; //循环变量改变步长
```

```
    }
```

```
}
```

例 2：/*求 1-100 的累加和*/

```
# include "ios tream.h"
```

```
void main( )
```

```
{
```

```
    int n,s;
```

```
    s=0; // 累加器
```

```
    n=1; // 计数器
```

```
    while ( n<=100 )
```

```
    {
```

学习实践部祝大家考试顺利(*^o^*)

```
s=s+n;

n++;

}

cout<<"s="<<s<<endl;

}
```

二. do - while 语句

1. 格 式: do

循环体语句组

while (条件) ;

2. 流程图: (P118)

3. 功能及特点: P117

4. 例 1: 改写程序 (输出 1-10 的立方值)

```
# include "iostream.h"

# include "math.h"

void main( )

{

    int i;

    i=1;

    do

    {

        cout<<i<<"的立方="<< pow( i ,3) <<endl;

        i++;

    } while ( i<=10 );

}
```

学习实践部祝大家考试顺利(*^o^*)