C++程序的基本要素

赵英良

1.标识符

- ▶程序中变量、类型、函数和标号的名称统称标识符.
 - a,b,name,int, char,main,void等
- 系统已有的标识符称为关键字。
- ▶常见关键字
 - using, namespace, void, return
 - int, float, double, char, bool, signed, unsigned, long, short, const, true, false, sizeof
 - if, else, for, do, while, switch, case, continue, break
 - class, struct, private, public, this, new, delete

- 自己给变量、函数等起的名字称为自定义标识符
- 自定义表示符的命名规则
 - 。由字母、数字和下划线组成
 - 。第1个符号不能是数字
 - 标识符的长度:不同编译器规定不同,但至少识别前6个字符,有的长度可达32个字符,甚至更长。
 - 。 区分大小写(大小写敏感)

标识符举例

- int a,b,c;
- int A,B,C;
- double a1,b2,c3;
- double table_width, table_height;
- char _name[40], _address[100];
- ▶ int 1a,2b,3c; //不合法的标识符
- ▶ 好习惯:标识符的命名应有意义,或与数学公式中 习惯使用的符号一致



2.变量和常量

(1)表示的数据可以改变的符号,称为变量。 int a; char name[40]; a = 5;cout << a << endl; a=a+5; cout << a << endl; cin>>name; cout << name << endl;

(2)直接写出的数据以及表示的数据不能改变的符号是常量。

cout << "Hello world" << endl; cout << 5+3 << endl;</pre>

3.运算符和表达式

(1)运算符

表示运算的符号

表达式中使用圆括号可以改变 运算的优先级。括号中的运算 比括号外的优先计算。

(2) 表示运算的式子就是表达式

可以是一个变量、一个常量

a, b, 5

也可以是用运算符将变量、常量连接起来

3+5, a+b, a-b, a*b, a>b, a<b, a>=b, a<=b

还可以是用运算符将表达式连接起来的式子

(a+b)*(a-b)

4.语句

- ▶ 语句是一个表达完整的命令
 - int a,b,c;// 这是一个说明语句,说明用a,b表示整数
 - 。a+b //这是一个表达式,虽然能表达求和运算,但运算 结果并没有说明如何处理
 - 。c=a+b; //语句, 计算a和b的和, 将结果存入c中
- ▶ C++中, 语句的标志就是分号";", 而不是等号
- ▶ 一个式子后面写上分号,就是一条语句
 - 。a+b; //在C++中也是语句, 但这个语句没什么作用
 - 。a; //也是语句,也没什么作用
 - 。;;; //这是3条语句,什么也没有,什么也不做,空语句
 - 。c=a+b__//虽然有等号,但没有分号,也不是语句____

- ▶ 等号也是运算符
- 》等号连接起来的式子是表达式,如

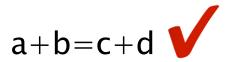
$$c=a+b$$

它有计算结果

$$a=5;b=6;$$

- 只有加上分号,才是语句 c=a+b;
- 等号的左边必须是变量

$$> 5=3$$
, $(a+b)=c+d$



5.输入和输出



不写长语句

▶ (1)输入cin

```
cin>>v1>>V2>>V3>>V4;//cin输入流对象>>提取运算符 int a,b,c;
```

cin>>a;

cin>>b;

cin>>c;



▶ (2)输出cout

cout << v1 << v2 << v4; // cout 输出流对象 << 插入运算符 cout << a;

cout <<" " < < b < < " " < < c < < " < sum: " < < a + b + c < < endl;

6.函数

- 能完成一定功能的有名的程序序列。
- main函数
 int main()
 { int a,b;
 cin>>a>>b; cout<<a+b<<endl; return 0;
 }</pre>
- > 系统函数 sin(x), cos(x), exp(x), sqrt(x)
- ▶自定义函数
- ▶ 函数是组成C++程序的基本单位

内容总结

- ▶ 标识符
- > 变量和常量
- 运算符和表达式
- > 语句
- ▶ 输入输出
- ▶ 函数
-