简单信息的表达和运算

常量和变量



赵英良

1.常量

- 具体把数据写出来
- 2,3,4; 1.2,1.3; "Hello world!","C++"
- cout << 2015;</p>
- ▶ 常量:不能改变的量
- >字面常量(字面量、直接常量):直接写出的数据。
-)符号常量:用符号表示数据,但它一旦确定了表示哪个数据,以后就不再改变。

符号常量①

- ▶ 说明一个符号表示常量
 - 。const <类型说明符> <符号常量名> =<数据>;
- <类型说明符>是表示数据类型的符号,

类型说明符	表示的数据类型	类型说明符	表示的数据类型
int	整型	float	单精度浮点型
short int 或short	短整型	double	双精度浮点型
long int 或long	长整型	long double	长双精度
unsigned int	无符号整型	char	字符型
long long int		bool	逻辑型/布尔型

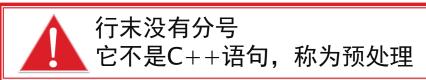
▶ 例如:

- const int N=100;
- const double pai=3.1415926;
- const char A='E';
- const char CPP[20]="C++ Programming";
- const int M=100, K=100;
- ∘ N=200;
- pai=3.14;
- ∘ A='F';



符号常量②

- ▶ 在main函数前
 - #define <符号名称> <数据>
- ▶ 例如:
 - #define PAI 3.14
- 程序编译时,先将<符号名称>全部替换为<数据>,然后才正式编译,所以这样的说明称为编译预处理



2.变量

> 变量

- 。值可以变的量。
- 更通俗地说,变量是用符号表示数据,它表示的数据是可以变的。

> 变量的说明

- 。<类型说明符> <变量名表>;
- <类型说明符>,就是前面介绍常量时的类型说明符,象 int, float, double, char, bool等。
- 。<变量名表>是一个或多个标识符,多个标识符之间用逗号隔开.

变量说明举例

int a; double d1,d2; char c1,c2,c3;



变量的说明也叫<mark>变量的声明</mark> 变量意味着编译系统会给这个变量在内存中分配一个存储单元。 所以变量的声明也叫变量的定义

变量的初始化

- 在声明变量的同时说明它代表的数据是什么, 称为变量的初始化
- ▶ 格式是

```
<类型说明符> <变量名1>=<值或表达式1>[, <变量名2>=<值或表达式2>];
```

> 例如

```
int a=5;
double pai=3.14, e=2.71828;
char A='?', B='%';
cout<<2*pai*a<<endl;
```

变量的赋值

- 设定变量值<变量名>=<值或表达式>;
- 其中的等号称为赋值运算符,其作用之一是将右边 表达式的值放到左边变量的存储单元中去。
- **M**例如

```
const double PAI=3.14; //声明常量PAI double r, area; //声明变量r, area分别表示半径和面积 r=5.0; //为r赋值 area=PAI*r*r; //计算圆的面积并赋值给area cout<<area<<endl; //显示结果
```

变量通过输入语句赋值

```
#include<iostream>
using namespace std;
int main(){
 const double PAI=3.14; //声明常量PAI
 double r, area; //声明变量r, area分别表示半径和面积
 cin>>r; //通过输入为r赋值
 area=PAI*r*r; //计算圆的面积并赋值给area
 cout<<area<<endl;//显示结果
 return 0;
```

变量使用中应注意的问题

- ▶ 1.先定义,再使用
- ▶ 2.只能定义一次,不能重复定义
- ▶ 3. 先赋值,再参与计算
- ▶ 4.变量可以多次赋值,而常量是不能赋值的
- 5.变量的定义可以出现在使用前的任何地方,但建 议在程序的开始定义变量
- ▶ 6.少用或不用字面常量

内容要点

▶常量

- 。字面常量、符号常量
 - · const说明,是语句,末尾有分号
 - · #define说明,预处理,行末没有分号

▶变量

- 。 先定义,后使用
- 先赋值,后运算
- 。 定义一次,赋值多次