C++程序结构

YouTube频道： hwdong

博客： hwdong-net.github.io

C++程序结构

- main()函数
- 语句
- 注释（多行或单行）
- 预处理指令(包含、宏定义)
- 函数（程序入口主函数main）
  - 变量和常量（数据）
  - 表达式语句（数据处理）
  - 控制语句、返回语句等
- 名字空间
- 标识符

函数

- C++程序是由一些**函数**构成的，每个C++程序都执行唯一的叫作main的**主函数**。

```
main()
{
}
```

语句

- C++程序的最小的完整执行指令都是以分号结尾的**语句**。

```
int main()
{
    return 0;
}
```

语句

- C++程序的最小的完整执行指令都是以分号结尾的**语句**。
- return语句结束函数执行，并可返回结果给调用者

```
int main()
{
    return 0;
}
```

注释

- 用以对程序代码进行说明，方便他人或今后阅读理解程序。

```
/*
    这是我的第一个C++程序
    This is my first C++ program
    作者:hwdong
    2020-02-24
*/

#include <iostream>    //输入输出流头文件
int main() { // main函数是程序的主入口，程序总是从这里执行
    return 0; //程序结束，返回结果0
}
```

/*和 */包围起来的多行注释

//开头的单行注释

- 多行注释不能嵌套

```
/*
    这是我的第一个C++程序
    This is my first C++ program
    作者:hwdong
    //修改日期：2020-02-24
*/

#include <iostream>    //输入输出流头文件
int main() { // main函数是程序的主入口，程序总是从这里执行
    return 0; //程序结束，返回结果0
}
```

块注释中不能包含/*或*/

但可以包含//

预处理语句

- 预处理器对这些#开头的预处理指令进行处理

```
#include <iostream>
int main() {
    std::cout << "hello world" << std::endl;
    return 0;
}
```

endl是std中的换行符

包含指令： #include

- 预处理器用文件内容替换掉该语句

```
#include <iostream>
int main() {
    std::cout << "hello world" << std::endl;
    return 0;
}
```

#include <iostream>

预处理指令

在系统路径下寻找库文件

包含指令： #include

- 预处理器用文件内容替换掉该语句

```
#include <iostream>
int main() {
    std::cout << "hello world" << std::endl;
    return 0;
}
```

#include “iostream”

预处理指令

先在当前目录，后在系统路径下寻找库文件

条件编译指令

- #if、#else、#elif 、#endif

```
#if 0
main(){

}
#else
int main(){
   return 0;
}
#endif
```

条件编译指令

- #ifdef、#else、#elif 、#endif

```
#ifdef UNIX
...
#endif

int main(){
   /*... */
}
```

宏定义指令： #define

```
#include <iostream>
int main() {
    std::cout << 3.14 * 2.5 * 2.5 << std::endl;
    // ...
    std::cout << 2 * 3.14 * 2.5 << std::endl;
    return 0;
}
```

宏定义指令： #define

```
#include <iostream>
int main() {
    std::cout << 3.1415926 * 2.5 * 2.5 << std::endl;
    // ...
    std::cout << 2 * 3.14 * 2.5 << std::endl;
    return 0;
}
```

宏定义指令： #define

- 给一个常量或文字起一个名字。

```
#include <iostream>
#define PI 3.1415926 //PI就是宏常量，#define就是宏定义指令
int main() {
    std::cout << PI * 2.5 * 2.5 << std::endl;
    // ...
    std::cout << 2 * PI * 2.5 << std::endl;
    return 0;
}
```

宏定义指令： #define

- 预处理器会将程序中所有PI用该常量的值3.14159替换掉

```
#include <iostream>
#define PI 3.1415926 //PI就是宏常量，#define就是宏定义指令
int main() {
    std::cout << PI * 2.5 * 2.5 << std::endl;
    // ...
    std::cout << 2 * PI * 2.5 << std::endl;
    return 0;
}
```

变量、类型

- 数据：常量和变量， 变量就是一块内存，每个变量都有确定的类型。
- 类型：决定了对变量能进行说明运算、变量占内存大小、变量值范围

i: 整型
ch:字符型
radius:双精度浮点实数
ok:布尔类型

```
#include <iostream>
int main(){
    int i = 3;
    char ch = 'A';
    double radius = 2.56;
    bool ok = false;

    std::cout << i << std::endl;
    std::cout << ch << std::endl;
    std::cout << radius << std::endl;
    std::cout << ok << std::endl;
}
```

3, ‘A’, ,2.56, false都是文字常量

```
3
A
2.56
0
```

运算符、表达式

- 表达式：用运算符对数据(变量/常量）进行运算

```
#include <iostream>
int main() {
    std::cout << "长宽为" << 5.8 << ',' << 3.4 << "的矩形\n";
    std::cout << "其周长是：" << 2 * (5.8 + 3.4) << '\n';
    std::cout << "其面积是：" << 5.8*3.4 << std::endl;
    return 0;
}
```

运算符、表达式

- 表达式：用运算符对数据(变量/常量）进行运算

```
#include <iostream>
int main() {
    int x, y = 4;
    x = 3;
    int z = 2 * (x + y);
    std::cout << z << std::endl;
    return 0;
}
```

同类型变量可在1个语句定义，可初始化

赋值表达式

算术表达式

赋值运算符：=

给变量赋值，即将数据存储到变量对应内存中

算术运算

```
x + y // plus
+x //unary plus
x-y //minus
-x //unar y minus
x*y //multiply
x / y // divide
x%y // remainder (modulus) for integers
```

比较运算

```
x == y // equal
x != y // not equal
x<y // less than
x>y // greater than
x <= y // less than or equal
x >= y // greater than or equal
```

修改一个变量的组合运算

```
x += y // x = x+y
++x // increment: x = x+1
x-= y // x = x-y
--x //decrement: x = x-1
x*= y // scaling: x = x*y
x /= y // scaling: x = x/y
x %= y // x = x%y
```

输入输出（io）

- cout是代表控制台窗口的**输出流对象**，可用 输出运算符<<向它输出

```
std::cout << "hello world!";
```

输出流对象　输出运算符　数据

- cin是代表键盘的**输入流对象**，可用 输入运算符>>从它输入

```
double r;
std::cin >> r;
```

输入流对象　输出运算符　变量

```cpp
#include <iostream>
int main() {
    double radius;
    std::cout << "请输入圆的半径：";
    std::cin >> radius;
    std::cout <<"圆的面积是："<< 3.14* radius * radius << std::endl;
    return 0;
}
```

```
请输入圆的半径：2.5
圆的面积是：19.625
```

格式化输出

\n 换行符　　\t 制表符　　设置宽度：setw(10)

setw函数需要的头文件

```
#include <iomanip>
#include <iostream>
int main() {
    std::cout << "\n\n未格式化输出\n";
    std::cout << "Ints" << "Floats" << "Doubles" << "\n";

    std::cout << "\nsetw(15) 设置输出文本宽度\n";
    std::cout << "Ints" << std::setw(15) <<"Floats"
        <<std::setw(15) <<"Doubles"<<"\n";

    std::cout << "\n\n制表符 \n";
    std::cout << "Ints\t" << "Floats\t" << "Doubles" << "\n";
    return 0;
}
```

```cpp
#include <iomanip> //说明了setw()函数
#include <iostream>
int main() {
    double radius;
    std::cout << "未格式化输出\n";
    std::cout << "Ints"<<"Floats"<<"Doubles";
    std::cout << "\n\n格式化输出\n";
    std::cout << "Ints" <<std::setw(10)<< "Floats"
        << std::setw(10) << "Doubles";

    std::cout << "\n\n制表符\n";
    std::cout << "Ints\t" << "Floats\t"<< "Doubles";
    return 0;
}
```

```
未格式化输出
IntsFloatsDoubles

格式化输出
Ints    Floats   Doubles

制表符
Ints    Floats  Doubles
```

定义函数

- 定义函数的格式：

返回类型 函数名(参数列表)

{

多个语句构成的程序块

}

```
#include <iostream>
int main() {
    std::cout << "hello, " ;
    std::cout << "wang " << std::endl;
    std::cout << 3+4<< std::endl;
    return 0;
}
```

调用函数

- 通过函数名调用执行函数体中的语句

```
#include <iostream>
int main() {
    std::cout << "hello, " ;
    std::cout << "wang " << std::endl;
    std::cout << 3+4<< std::endl;
    return 0;
}
```

```
//返回类型void表示函数不返回结果
void hello() {
    std::cout << "hello, ";
    std::cout << "wang " << std::endl;
}
void add3_4() {
    std::cout << 3 + 4 << std::endl;
}
int main() {
    hello();
    add3_4();
    return 0;
}
```

调用hello()函数

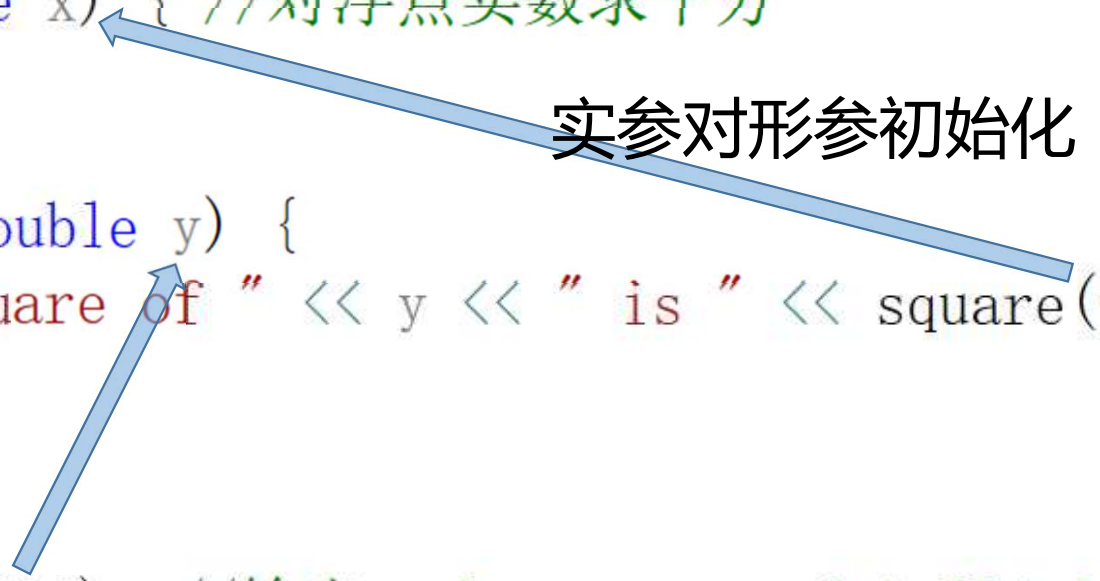
调用add3_4()函数

函数的参数

- 函数定义时可以有参数，调用时要提供对应的数据

```
#include <iostream>
using std::cout;
double square(double x) { //对浮点实数求平方
    return x*x;
}
void print_square(double y) {
    cout << "the square of " << y << " is " << square(y) << '\n';
}

int main() {
    print_square(1.234); //输出：the square of 1.234 is 1.52276
}
```



- 练习： 完成下面的函数f.

```
#include <iostream>
? f(?) {
    ?
}
int main() {
    std::cout << "请输入2个整数" << std::endl;
    int x,y;
    while (std::cin >> x) {
        if (std::cin >> y) {
            std::cout<<f(x, y)<<endl;
        }
        else break; //break跳出循环语句
        std::cout << "请输入2个整数" << std::endl;
    }
    return 0;
}
```

标识符

- 标识符用来给“类型”、“变量(对象)”、“函数” 等起名字
- 标识符由字母、数字、下划线_构成，但不能以数字开头

  A　3a　i2　radius　_name　#width

- 标识符是大小写敏感的. *Arr* 和 *arr* 是不同的标识符
- 标识符是不能和C++关键字相同
- 标识符的作用域(scope)是这个名字有效的范围。作用域分为全局、函数和块作用域

保留的关键字

alignas	default	noexcept	this
alignof	delete	not	thread_local
and	do	not_eq	throw
and_eq	double	nullptr	true
asm	dynamic_cast	operator	try
auto	else	or	typedef
bitand	enum	or_eq	typeid
bitor	explicit	private	typename
bool	export	protected	union
break	extern	public	unsigned
case	false	register	using
catch	float	reinterpret_cast	virtual
char	for	return	void
char16_t	friend	short	volatile
char32_t	goto	signed	wchar_t
class	if	sizeof	while
compl	inline	static	xor
const	int	static_assert	xor_eq
constexpr	long	static_cast	override*
const_cast	mutable	struct	final*
continue	namespace	switch	
decltype	new	template	

下面哪些是非法的名字（标识符）？

(a) `int double = 3.14;`

(b) `int _;`

(c) `int catch-22;`

(d) `int 1_or_2 = 1;`

(e) `double Double = 3.14;`

名字空间namespace

- **名字冲突**：不同的库可能出现同名，如同生活中出现同名。

```
信控 {
    张伟
    李平
    王强
}
```

```
计算机{
    李伟
    张伟
    王兵
}
```

```
张伟

信控的张伟
计算机的张伟
```

名字限定：在具体对象名字前加上其所属的单位(名字空间)

信控 的 张伟

名字空间namespace

- 解决不同的库开发者的“名字冲突”问题。
- 信控的张伟 、计算机的张伟

名字限定::

A :: name

```
namespace  A {
    int a;
    double f() {/*…*/ }
    int g(int x) {/*…*/ }
    //...
}
namespace  B {
    double a;
    int f() {/*…*/ return 0; }
    char g;
    //...
}
```

定义名字空间

```
int main() {
    int i = 2;
    a = B::f();   //错
    B::a = f(); //错
    A::a = B::f(); +A::g(2);
}
```

使用namespace中的名字

- **名字限定:** **名字空间名::** 名字　　如 A::f　B::b
- **引入整个名字空间**：　using namespace A;

```
namespace  A {
    int a;
    double f() {/*…*/ }
    int g(int x) {/*…*/ }
    //...
}
namespace  B {
    double a;
    int f() {/*…*/ return 0; }
    char g;
    //...
}
```

```
using namespace A;
int main_9() {
    int i = 2;
    a = B::f();
    B::a = f();
    A::a = B::f(); +A::g(2);
}
```

使用namespace中的名字

- **名字限定**: **名字空间名::** 名字　　如 A::f　B::b
- **引入整个名字空间**：　using namespace A;
- **using 声明**:　**using 名字空间名::** 名字;

```
namespace  A {
    int a;
    double f() {/*…*/ }
    int g(int x) {/*…*/ }
    //...
}
namespace  B {
    double a;
    int f() {/*…*/ return 0; }
    char g;
    //...
}
```

```
using A::a;
using A::f;
int main_9() {
    int i = 2;
    a = B::f();
    B::a = f();
    A::a = B::f(); +A::g(2);
}
```

控制语句：条件语句

- 满足条件时就执行其程序块。
- if条件语句：if …else…

```
#include <iostream>
int main() {
    double radius;
    std::cin >> radius;
    if (radius <= 0)
        std::cout << "错误：半径<=0" << std::endl;
    else
        std::cout << "圆的面积是：" << 3.1415*radius*radius;
}
```

控制语句：条件语句

- 满足条件时就执行其程序块。
- if条件语句：

```
if（条件表达式）
    程序块1
else
    程序块2
```

控制语句：条件语句

```
if（条件表达式）
    程序块1
else
    程序块2
```

```
#include <iostream>
int main() {
    double radius;
    std::cin >> radius;
    if (radius <= 0)
        std::cout << "错误：半径<=0" << std::endl;
    else
        std::cout << "圆的面积是：" << 3.1415*radius*radius;
}
```

控制语句：循环语句

- 满足条件时就一直执行其程序块。

  while（条件表达式）

  　　程序块

```
#include <iostream>
int main() {
    int i = 1,sum = 0;
    while (i <= 100) {
        sum = sum + i;  //sum+=i;
        i = i + 1; //  i++;
    }
    std::cout << "1到100整数之和是："
        << sum << std::endl;
}
```

C++ 程序结构

一个程序文件通常包含下列一些：

- 注释（多行或单行）
- 预处理语句(包含、宏定义)
- 函数（程序入口主函数main）
  - 变量和常量（数据）
  - 表达式语句（数据处理）
  - 控制语句、返回语句等

```
/*
计算圆的面积：
Area = PI*r*r
*/
#include <iostream>
using namespace std;
#define PI 3.1415926
int main() {
    //变量名可以含多个字母、下划线、数字
    double radius;
    std::cin >> radius;
    if (radius <= 0)
        cout << "Error\n";
    else {
        double area = PI*radius*radius;
        std::cout << area << std::endl;
    }
    return 0;
}
```

练习

- 找出程序中的错误

```
#include <iostream>
using namespace std
void main () {
  int FTemp = 0
  int CTemp = 0;
  cout >> "Enter a Farenheit temperature: ";
  cin << FTemp;   CTemp = FTemp - 32 / (9/5);
  cout >> "\n <<FTemp >> " in Farenheit = "  >> CTemp >> in Celsius\n";
  return 0;
}
```

```
/* 目标：写一个程序计算体积： cube(立方体), sphere(球), cone(圆锥).
**Cube Volume = side^3
**Sphere Volume = (4/3) * pi * radius^3
**Cone Volume = pi * radius^2 * (height/3)
**Write the values to the console.
*/

#include<iostream>
int main(){
   float cubeSide = 5.4;    //Dimension of the cube
   float sphereRadius = 2.33;   //Dimension of sphere
   float coneRadius = 7.65;   //Dimensions of cone
   float coneHeight = 14;
   float volCube, volSphere, volCone = 0;
   return 0;
}
```

- 从键盘输入3个实数，然后按从小到大的程序输出
- 从键盘输入一系列整数，求它们的最大值并输出

```
#include <iostream>
int main() {
    ? a; //定义接受键盘输入的整数变量
    ? max = -10000000;//初始化最大值
    while(?::cin>>a){
        if(?){
            max = ;
        }
    }
    // \n是转义字符表示换行，等同于std::endl
    std::cout<<"输入的最大整数是"<< ? << '\n';
    return 0;
}
```

- 假设行数从键盘输入，打印输出如下的金字塔。

```
#include <iostream>
int main(){
    printf("Enter number of rows: ");
    std::cin>>rows;

    for(int i=1; i<=rows; ++i) {
        //输出一定数量的空白
        for(int space=1; space <= ? ; ++space){
            std::cout<<"  ";
        }
        //输出2*i-1个星号*
        for(int k = 0 ; k< ? ;k++)  {
            ?
        }
        //输出换行
        std::cout<<std::endl
    }
    return 0;
}
```

- 假设行数从键盘输入，打印输出如下的金字塔。

- 写一个程序，从键盘输入一个正整数，判断它是否是质数。

关注我

博客： hwdong-net.github.io

Youtube频道

hwdong

@hwdong · 5.01K subscribers · 558 videos

博客：https://hwdong-net.github.io >

youtube.com/c/4kRealSound and 4 more links