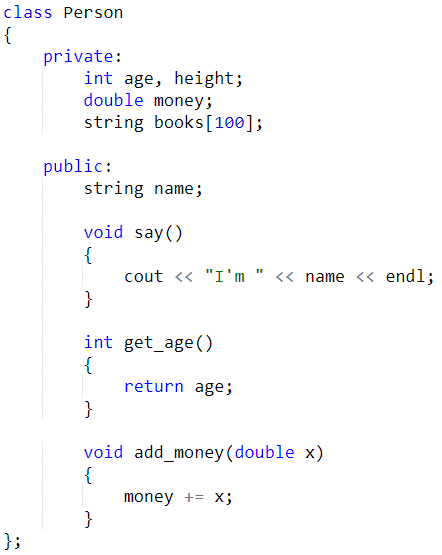
第七章 类、结构体、指针、引用

类可以将变量、数组和函数完美地打包在一起。

——闫学灿

1. 类与结构体

类的定义：



类中的变量和函数被统一称为类的成员变量。

private后面的内容是私有成员变量，在类的外部不能访问；public后面的内容是公有成员变量，在类的外部可以访问。

类的使用：

#include <iostream>

using namespace std;

const int N = 1000010;

class Person

{

private:

int age, height;

double money;

string books[100];

public:

string name;

void say()

{

cout << "I'm " << name << endl;

}

int set\_age(int a)

{

age = a;

}

int get\_age()

{

return age;

}

void add\_money(double x)

{

money += x;

}

} person\_a, person\_b, persons[100];

int main()

{

Person c;

c.name = "yxc"; // 正确！访问公有变量

c.age = 18; // 错误！访问私有变量

c.set\_age(18); // 正确！set\_age()是共有成员变量

c.add\_money(100);

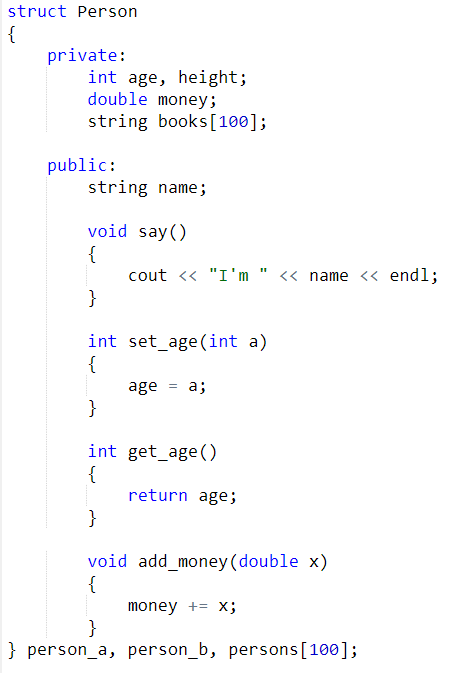
c.say();

cout << c.get\_age() << endl;

return 0;

}

结构体和类的作用是一样的。不同点在于类默认是private，结构体默认是public。



1. 指针和引用

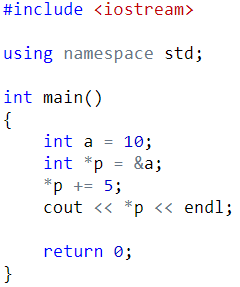
指针指向存放变量的值的地址。因此我们可以通过指针来修改变量的值。

Int \*p 表示 = & a 可以看成 int\* p

p 是一个int类型遍历的指针

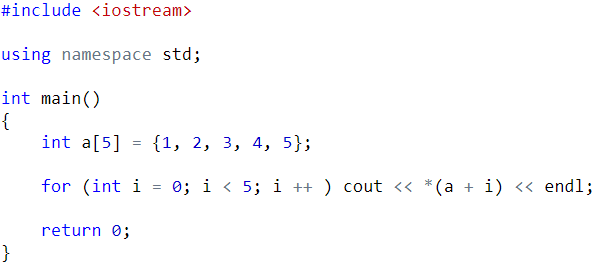
&a 表示取a的地址

\*p 表示取p的值



数组名是一种特殊的指针。指针可以做运算：

数组 a 本身就是一个指针，存的是数组的开始地址



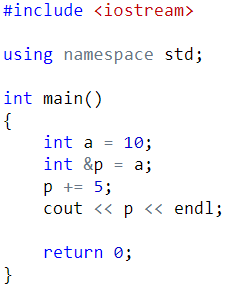
\*（a + 1） 会根据 a 指针的类型，来决定加几个字节，因为 a 是int型，所以会加4个字节

引用和指针类似，相当于给变量起了个别名。

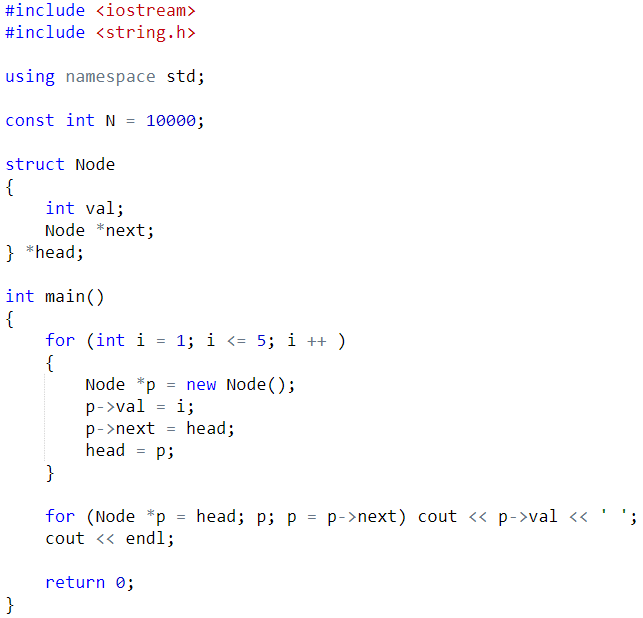
Int &p = a; // 表示 p是a的一个别名，存在一个地址上

是对以下操作的一种简化

Int\* p = &a;



1. 链表



Node\* p = new Node(1);

此时的 p 是一个指针，调用next就需要使用p->next