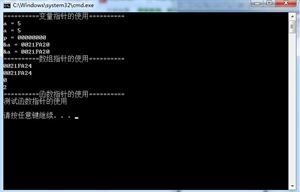
详解C++中的指针、数组指针与函数指针

转藏

[http://www.jb51.net/article/70241.htm](http://www.jb51.net/article/70241.htm" \t "http://www.360doc.cn/article/_blank)2015C++中一个重要的特性就是指针，指针不仅具有获得地址的能力，还具有操作地址的能力。指针可以用于数组、或作为函数的参数，用来访问内存和对内存的操作，指针的使用使得C++很高效，但是指针也非常危险，使用不当会带来比较严重的问题。1、指针程序中所有的变量和常量都存在一个内存地址中，当然，函数也有对应的内存地址，内存地址的不同会导致程序执行时有所不同。指针就是用来控制和存储内存地址的变量，它指向单个对象的地址，除了void之外，指针的数据类型与所指向地址的变量数据类型保持一致。2、如何定义指针、数组指针、函数指针常见的指针定义有3种：变量指针、数组指针和函数指针的定义。（1）、变量指针的定义复制代码 代码如下:int\* p=0;   //定义指针p，并初始化指针为0，即指向的地址为0000 0000或int a=0;    //定义初始化常量aint\* p; //定义指针pp=&a;   //指针p指向a的地址，即指针获取地址（2）、数组指针的定义复制代码 代码如下:int a[]={0,1,2,3,4,5,6,7,8,9};  //定义数组int\* p=a;   //定义并赋值数组指针，即获得数组的首地址（3）、函数指针的定义复制代码 代码如下:int f();    //定义函数int (\*p)(); //定义函数指针p=f;    //赋值函数指针，即获得函数代码的首地址区别变量指针、数组指针和函数指针的定义，示例代码如下。?123456789101112131415161718192021222324252627282930313233343536#include<iostream>using namespace std;int f() //定义一个函数{cout<<"测试函数指针的使用"<<endl<<endl;return 0;}void main(){cout<<"==========变量指针的使用=========="<<endl;int a=5;int\* p=0;  //初始化指针为0int\* q;   //定义指针q=&a;    //赋值指针cout<<"a = "<<a<<endl;    //变量a的值cout<<"a = "<<\*q<<endl;    //变量a的值cout<<"p = "<<p<<endl;    //指针p的地址为0000 0000cout<<"&a = "<<&a<<endl; //获取a的地址cout<<"&a = "<<q<<endl;    //获取a的地址cout<<"==========数组指针的使用=========="<<endl;int b[]={0,1,2,3,4,5,6,7,8,9};int\* pb=b;   //直接指向第一个元素的地址cout<<pb<<endl //第1个元素的地址，即数组的首地址<<b<<endl  //第1个元素的地址，即数组的首地址<<\*pb<<endl //第1个元素的值<<(\*pb+2)<<endl;  //第3个元素的值cout<<"==========函数指针的使用=========="<<endl;int f();    //定义函数int (\*pf)();  //定义函数指针pf=f;      //赋值函数指针，即将函数的首地址赋值给指针pf(\*pf)();    //通过函数指针调用函数}结果如下图：

[](http://www.360doc.cn/article/javascript:void(0)" \t "http://www.360doc.cn/article/_blank)

[3、数组指针与指针数组数组指针是一个指针变量，它指向一个数组，即指向一个数组的指针就是数组指针；而指针数组是一个只包含指针元素的数组，它的元素可以指向相同类型的不同对象。4、函数指针与指针函数函数指针就是指向函数的存储空间地址的指针，可以对函数指针进行赋值并且通过函数指针来调用函数，它的本质是一个指针。而指针函数只是说明它是一个返回值为指针的函数，其本质是一个函数。5、引用传递、值传递和指针传递C++语言中，函数的参数和返回值的传递方式有3种：即值传递、引用传递和指针传递。（1）、值传递形参是实参的拷贝，改变形参的值并不会影响外部实参的值。从被调用函数的角度来说，值传递是单向的（实参->形参），参数的值只能传入，不能传出。当函数内部需要修改参数，并且不希望这个改变影响调用者时，采用值传递。（2）、指针传递形参为指向实参地址的指针，当对形参的指向操作时，就相当于对实参本身进行的操作。（3）、引用传递形参相当于是实参的“别名”，对形参的操作其实就是对实参的操作，在引用传递过程中，被调函数的形式参数虽然也作为局部变量在栈中开辟了内存空间，但是这时存放的是由主调函数放进来的实参变量的地址。被调函数对形参的任何操作都被处理成间接寻址，即通过栈中存放的地址访问主调函数中的实参变量。正因为如此，被调函数对形参做的任何操作都影响了主调函数中的实参变量。最后，总结一下指针和引用的相同点和不同点：相同点：都是地址的概念，指针指向一块内存，它的内容是所指内存的地址；而引用则是某块内存的别名。不同点：指针是一个实体，而引用仅是个别名；引用只能在定义时被初始化一次，之后不可变；指针可变；引用“从一而终”，指针可以“见异思迁”；引用没有const，指针有const，const的指针不可变；（具体指没有int& const a这种形式，而const int& a是有的，前者指引用本身即别名不可以改变，这是当然的，所以不需要这种形式，后者指引用所指的值不可以改变）引用不能为空，指针可以为空；“sizeof 引用”得到的是所指向的变量（对象）的大小，而“sizeof 指针”得到的是指针本身的大小；指针和引用的自增(++)运算意义不一样；引用是类型安全的，而指针不是（引用比指针多了类型检查）。](http://image97.360doc.com/DownloadImg/2016/05/1611/71841848_1" \t "http://www.360doc.cn/article/_blank)