

Ch.3 深入的话题

2017 / 10

An Important Tip

- 学会读代码

- 构造函数
- 拷贝构造函数
- 析构函数
- 运算符重载
- 函数对象，临时对象

构造函数

- 什么时候运行？
 - 产生对象的时候（诞生即执行）

析构函数

- 什么时候运行？
 - 对象销毁的时候

产生时机

- 什么时候产生?
 - Object a;
 - Object b(a); // 通过a产生了b, 拷贝构造
 - Object c = a; // 调用了拷贝构造和 =
 - Object * p = new Object();

产生时机

- 什么时候产生?
 - Object a;
 - Object b(a);
 - Object c = a;
 - Object * p = new Object();

销毁的时机

- 分配在栈上的对象，当栈销毁
- 堆上对象，调用`delete`主动销毁

限定对象的分配

- 让对象只能分配在栈上
 - `Object obj;`
 - `Object obj [5];`
 - `Object * pObj = new Object();` // 不能这么使用
- 让对象只能分配在堆上
 - `Object * pObj = new Object();` // 只能这么使用
 - `Object obj;` // 不能这么使用

只分配在栈上

```
class StackObject
{
public:
    StackObject(std::string n) : name(n) {}
private:    // 把动态分配的函数私有化
    void * operator new(size_t size) {}
    void operator delete(void * p) {}
    void * operator new[](size_t size) {}
    void operator delete[](void * p) {}

    std::string name;
};
```


只在堆上分配

```
class HeapObject
{
public:
    static HeapObject * produceObject(std::string name)
    {
        return new HeapObject(name);
    }
private:
    // 把所有的构造函数都私有化
    HeapObject() {}
    HeapObject(std::string n) : name(n) {}

    std::string name;
};
```

函数对象

- 给定一个数组
- 进行各种各样的操作
- 循环怎么处理？