

new和delete表达式

一、new表达式工作步骤

使用new表达式时发生的三个步骤：

1. 调用名为operator new的标准库函数，分配足够大的原始的未类型化的内存，以保存指定类型的一个对象
2. 运行该类型的一个构造函数初始化对象
3. 返回指向新分配并构造的构造函数对象的指针

二、delete表达式工作步骤

使用delete表达式时发生的两个步骤：

1. 调用析构函数，回收对象中数据成员所申请的资源
2. 调用名为operator delete的标准库函数释放该对象所用的内存

三、operator new和operator delete函数的重载版本

```
1 //operator new库函数
2 void *operator new(size_t);
3 void *operator new[](size_t);
4
5 //operator delete库函数
6 void operator delete(void *);
7 void operator delete[](void *);
```

下面通过例子来说明这两个函数的用法

```
1 class Student
2 {
3 public:
4     Student(int id, const char *name)
5         : _id(id)
6         , _name(new char[strlen(name) + 1]())
7     {
8         cout << "Student()" << endl;
9         strcpy(_name, name);
10    }
11
12    ~Student()
13    {
14        cout << "~Student()" << endl;
15        delete [] _name;
16    }
17
18    void *operator new(size_t sz)
19    {
20        void *ret = malloc(sz);
21        return ret;
22    }
23
```

```

24     void operator delete(void *pointer)
25     {
26         free(pointer);
27     }
28
29     void print() const
30     {
31         cout << "id:" << _id << endl
32              << "name:" << _name << endl;
33     }
34
35 private:
36     int _id;
37     char _name;
38 };
39
40 int main(void)
41 {
42     Student *pstu = new Student(100, "Jackie");
43     pstu->print();
44
45     delete pstu; //思考一下：对象的销毁与执行析构函数是不是等价的？
46     return 0;
47 }
48 //注意：1、可以试试成员函数operator new/delete里面包含this没有？
49 //2、将operator new/delete函数放在类外且不加类名与作用域限定符，又会产生什么现象，该如何解释？

```

四、要求一个类只能创建栈对象

如果要求一个类只能创建栈对象，其意思是在创建出栈对象的同时，不能创建堆对象。以Student类型而言，

```

1 void test0()
2 {
3     //思考：栈对象的创建需要哪些条件？是不是只要构造函数不为私有就ok？
4     Student s1(101, "John");//ok
5     Student *pstu = new Student(101, "Mark");//error
6 }

```

要达到以上的效果，咱们只需要将operator new放入Student类的private区域。

五、要求一个类只能创建堆对象

如果要求一个类只能创建堆对象，其意思是在创建出堆对象的同时，不能创建栈对象。以Student类型而言，

```

1 void test1()
2 {
3     Student s1(101, "John");//error
4     Student *pstu = new Student(101, "Mark");//ok
5 }

```

要达到以上的效果，咱们只需要将Student类的构造函数放入private区域。

