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

```
//operator new库函数
void *operator new(size_t);
void *operator new[](size_t);

//operator delete库函数
void operator delete(void *);
void operator delete[](void *);
```

下面通过例子来说明这两个函数的用法

```
class Student
{
public:
    Student(int id, const char *name)
    : _id(id)
    , _name(new char[strlen(name) + 1]())
    {
        cout << "Student()" << endl;
        strcpy(_name, name);
    }

    ~Student()
    {
        cout << "~Student()" << endl;
        delete [] _name;
    }

    void *operator new(size_t sz)
    {
        void *ret = malloc(sz);
        return ret;
    }

```

```
    void operator delete(void *pointer)
    {
        free(pointer);
    }

    void print() const
    {
        cout << "id:" << _id << endl
             << "name:" << _name << endl;
    }

private:
    int _id;
    char  _name;
};

int main(void)
{
    Student *pstu = new Student(100, "Jackie");
    pstu->print();

    delete pstu;//思考一下：对象的销毁与执行析构函数是不是等价的?
    return 0;
}
//注意：1、可以试试成员函数operator new/delete里面包含this没有?
//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，

```
void test0()
{
    //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?
    Student s1(101, "John");//ok
    Student *pstu = new Student(101, "Mark");//error
}
```

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，

```
void test1()
{
    Student s1(101, "John");//error
    Student *pstu = new Student(101, "Mark");//ok
}
```

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