

满分25分。

将源码打包提交到网络学堂，命名格式为：job3-姓名-学号.zip

### 题目 3.0 [5 分]

本次作业的所有题目实现在“THU”命名空间下。

### 题目 3.1 [8 分]

完善题目2.1中的SafePtr类，将其改为模板类，管理任意对象的指针。声明框架如下：

```
template<class T>
class SafePtr {
    ...
};
```

ComicBook改成T即可

实现一个模板函数，用于交换两个SafePtr的值，声明如下：

```
template<class T>
void swap(SafePtr<T> & a, SafePtr<T> & b);
```

如果掉了private申明友元函数

### 题目 3.2 [9 分]

STL是C++泛型编程的经典之作。让我们实现一个iVector模板类，山寨STL中vector的功能。STL中使用分配器动态分配空间，本题简化为由构造函数确定vector大小。定义框架如下：

```
template<class T>
class iVector {
protected:
    int _size;
    T * _vector;
    ...
public:
    typedef T * iterator; //迭代器
    iVector(int n) : _size(n) {
        ...
    }
    ...
};
```

3.2.1 [3分] 为iVector类实现下列方法，功能对应STL vector的相关函数。

- iterator begin();
- iterator end();
- void clear();
- bool empty() const;
- void pop\_back();
- void push\_back(const T& v);
- T& operator[](int n);

3.2.2 [3 分] 编写友元函数实现冒泡排序（升序），声明如下：

```
template<class iterator>
void isort(iterator begin, iterator end);
```

3.2.3 [3 分] pop\_back、push\_back 和 operator[]中抛出数组越界异常，并编写至少一处测试代码捕获异常并输出提示信息。

### 题目 3.3 [3 分]

实现一个迭代器类iterator，定义框架如下：

```
template<class T>
class iterator {
protected:
    T * _iterator;
    ...
}
```

将题目3.2中typedef T \* iterator;替换成typedef iterator<T> iterator;，使得iVector类正常工作。