

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

3.0 [5 分] 将本次作业 3.1 和 3.3 题的实现放到 “THU”命名空间下。

3.1 [8 分] 完善 1.2 中 SafePtr 类，将其改为模板类。

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

实现一个模板函数，用于交换两个 SafePtr 的值。

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

3.2 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。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] 实现 iVector 的成员函数，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] 实现冒泡排序算法 sort（升序）。

```
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，至少写一处测试代码 catch 异常。

3.3 [3 分] 实现一个迭代器类 iIterator 替换 3.2 中的迭代器。

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

将 3.2 中 typedef T* iterator 替换成 typedef iIterator<T> iterator 。