嵌套类

```
class A {
    ...
    class B {
        ...
    };
    ...
};
```

- 嵌套类中，内部类和外部类都是各自独立的对象，各不包含对方的成员
- 内部类可以通过对象访问外部类的私有（保护）成员，但是外部类不能访问内部类的私有（保护）成员

```
class A {
    ...
    class B {
        ...
        void foo(A &obj)
        {
            obj._a = 10;     // 内部类可以通过对象访问外部类的私有（保护）成员
        }
    };
    ...
private:
    int _a;
};
```

迭代器

迭代器是一种检查容器内元素并遍历元素的数据类型。可以替代下标访问容器对象的元素。

每种容器类型都定义了自己的迭代器类型，如 `vector`：

```
vector<int>::iterator iter;
```

迭代器模拟了C++中的指针，可以有`++`运算，用`*`（解引用算符，deference）或`->`算符来访问容器中的元素。容器中元素如果改变了所用内存，也不影响绑定的迭代器指向正确的位置。

```
// 遍历vector<int> v;
for(auto iter = v.begin(); iter != v.end(); ++iter) {
    /* TODO: */
}

for(auto &item : v) {
    /* TODO: */
}

for_each(v.begin(), v.end(), [](int value) { /* TODO: */ });

// 排序
sort(v.begin(), v.end(), [](int a, int b) { /* TODO: */});
```

实现迭代器

详见作业