2.向量

（c4）无序向量

- 遍历

可是我们惊愕地发现，“发动起来的群众”，就像通了电的机器人，都随着按钮统一行动，都不是个人了

邓俊辉

deng@tsinghua.edu.cn

遍历

- 对向量中的每一元素，统一实施visit操作

  如何指定visit操作？如何将其传递到向量内部？

- template <typename T> //利用函数指针机制，只读或局部性修改

```
void Vector<T>::traverse( void ( * visit )( T & ) )
   { for ( int i = 0; i < _size; i++ ) visit( _elem[i] ); }
```

- template <typename T> template <typename VST> //利用函数对象机制，可全局性修改

```
void Vector<T>::traverse( VST & visit )
   { for ( int i = 0; i < _size; i++ ) visit( _elem[i] ); }
```

- 体会两种方法的优劣

实例

❖ 比如，为统一将向量中所有元素分别加一，只需...

❖ 首先，实现一个可使单个T类型元素加一的类

```
template <typename T> //假设T可直接递增或已重载操作符“++”
struct Increase //函数对象：通过重载操作符“()”实现
   {   virtual void operator()( T & e ) { e++; }   }; //加一
```

❖ 此后...

```
template <typename T> void increase( Vector<T> & V )
   {   V.traverse( Increase<T>() );   } //即可以之为基本操作遍历向量
```

❖ 作为练习，可模仿此例，实现统一减一、加倍，甚至求和等遍历功能