



UNIVERSITY OF SCIENCE AND TECHNOLOGY OF CHINA

地址: 中国安徽省合肥市 电话: 0551-63602184 传真: 0551-63631760 网址: <http://www.ustc.edu.cn>

前面两节笔记已上传了

3.

non-explicit-one-argument ctor
argument. 实参

意味着只需要一个实参. 有默认实参
explicit 是关键字. 可以加到类构造函数前面
可以把别的东西转化为这种东西.

Conversion function vs. non-explicit-one-argument ctor
可能导致 ambiguous 歧义

explicit (明白的)

explicit Fraction(int num, int den=1)

: m_numerator(num), m_denominator(den) { }

explicit 很少用到, 基本都是用到构造函数前面

conversion function 转换函数

4. 关于智能指针 point-like classes

为什么设计一个类让它像一个指针?

template <class T>

class shared_ptr

public:

shared_ptr(T* p): px(p) { } 借一般的指针包装产生智能指针

T& operator*() const

{ return *px; }

注意到操作符重载的重要性.

T* operator->() const

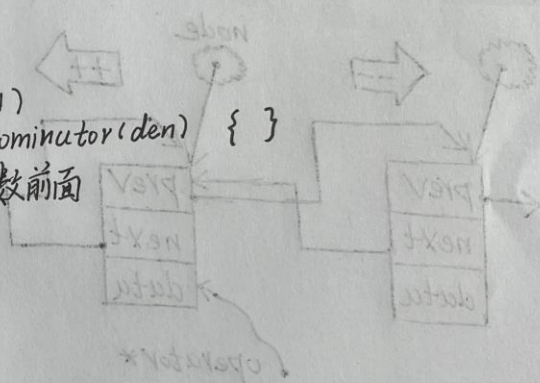
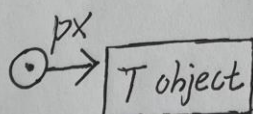
{ return px; }

shared_ptr<Foo> sp(new Foo);

private:

T* px;

long* pn;



sp → method();

px → method();

→ 操作符的特殊之处 操作符作用后的结果会被 → 符号继续作用

pointer-like class 一定带着普通指针

除了智能指针, 还有一种 pointer-like, 即迭代器

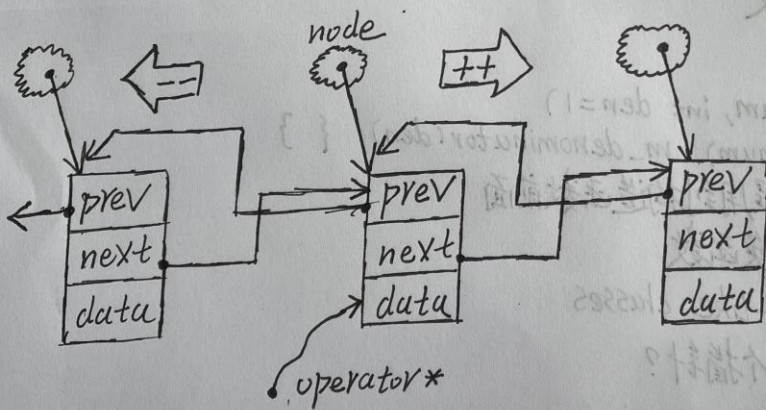
对于迭代器, 不但要处理 →, *, 还需要处理 ++, --

这是因为迭代器需要遍历容器

```
reference operator*() const  
{ return (*node).data; }
```

```
pointer operator->() const  
{ return &(operator*()); }
```

node 即迭代器



Object T