

一起来写 **vector**

标准 C++ 编程

- 薛浩
- xuehao0618@outlook.com



目录

CONTENTS

1

类

class

2

模板类

template class

3

关键字

const

4

运算符重载

operator overloading

5

特殊成员函数

copy-assign

6

移动语义

move semantics




复习

RECAP



复习 Recap

- 类的定义
 - 声明与实现分离
 - 构造函数
 - 析构函数
 - 动态内存管理
- 

复习 Recap

构造函数

- `vector<int> vec;`
- `vector<int> vec2(20);`
- `vector<int> vec3(3, 10);`

复习 Recap

动态内存管理

- `vector<int> intVec;` // 默认 10 个元素空间
- `intVec.reserve(5);` // 不调整
- `intVec.reserve(30);` // 默认 10 个元素空间

复习 Recap

插入元素

```
vector<int> intVec;  
auto pos = intVec.insert(intVec.begin() + 5, 8);  
int offset = pos - intVec.begin();  
EXPECT_EQUAL(offset, 5);           // 检测返回值  
EXPECT_EQUAL(intVec.size(), 1);    // 检测计数
```



CLASS



类 Class

命名空间

```
using namespace cs101;
```



类 Class

迭代器

```
vector<int> intVec(3, 10);  
for (auto itr = intVec.begin(); itr < intVec.end(); ++itr) {  
    std::cout << *itr << std::endl;  
}
```



模板类

TEMPLATE CLASS

模板类 Template Class

参数化类型

```
vector<double> doubleVec(3, 3.14);  
vector<std::string> strVec(3, "abc");
```



关键字

CONST

关键字 Const

Const-Correctness

```
vector<int> intVec(3, 10);  
if(intVec.at(1) = 1){  
    // do something  
}
```

关键字 Const

public
——
——
——
(const)

Const Interface

```
const vector<int> intVec(3, 10);  
intVec.size(); intVec.at(1);  
for (auto itr = intVec.begin(); itr < intVec.end(); ++itr) {  
    // do something  
}
```

<< → toString
() + () 无 &

() + = ()
↖ 有 &



运算符重载

OPERATOR OVERLOADING

方法

自由

friend

运算符重载 Operator Overloading

Operator 类型 {
}

转换函数

```
vector<int> intVec;  
if (intVec) {  
    // do something  
}
```

vector
↓
bool

运算符重载 Operator Overloading

成员函数

```
vector<int> intVec(3, 10);  
intVec[0] = 12;  
std::cout << intVec[0] << std::endl;
```

运算符重载 Operator Overloading

自由函数

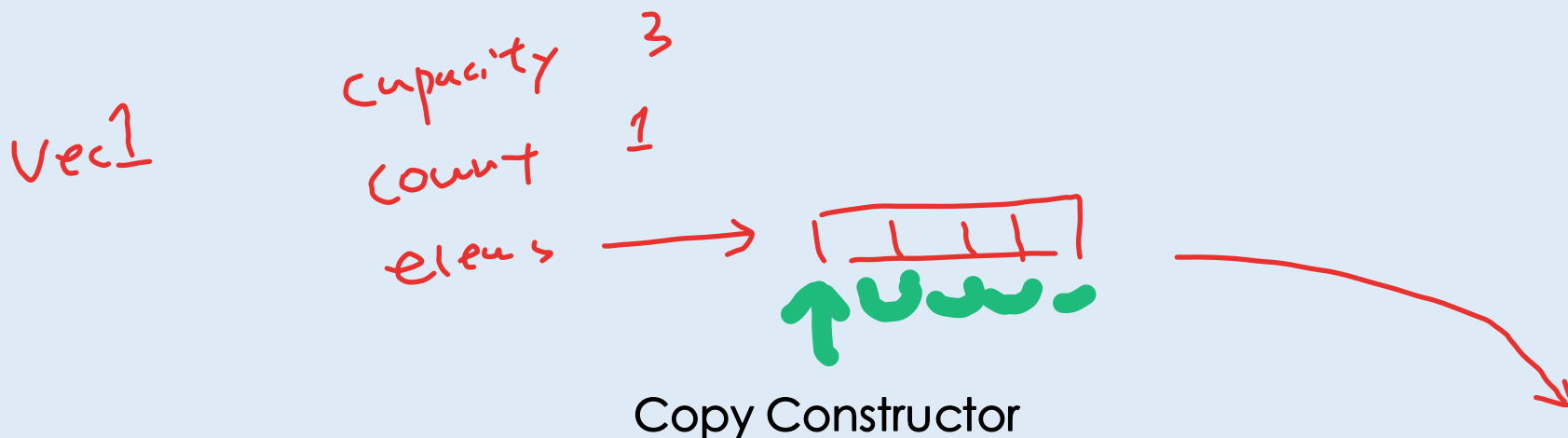
```
vector<int> intVec(3, 10);  
std::cout << intVec << std::endl;  
vector<std::string> strVec(3, "abc");  
std::cout << intVec.toString() << std::endl;
```



特殊成员函数

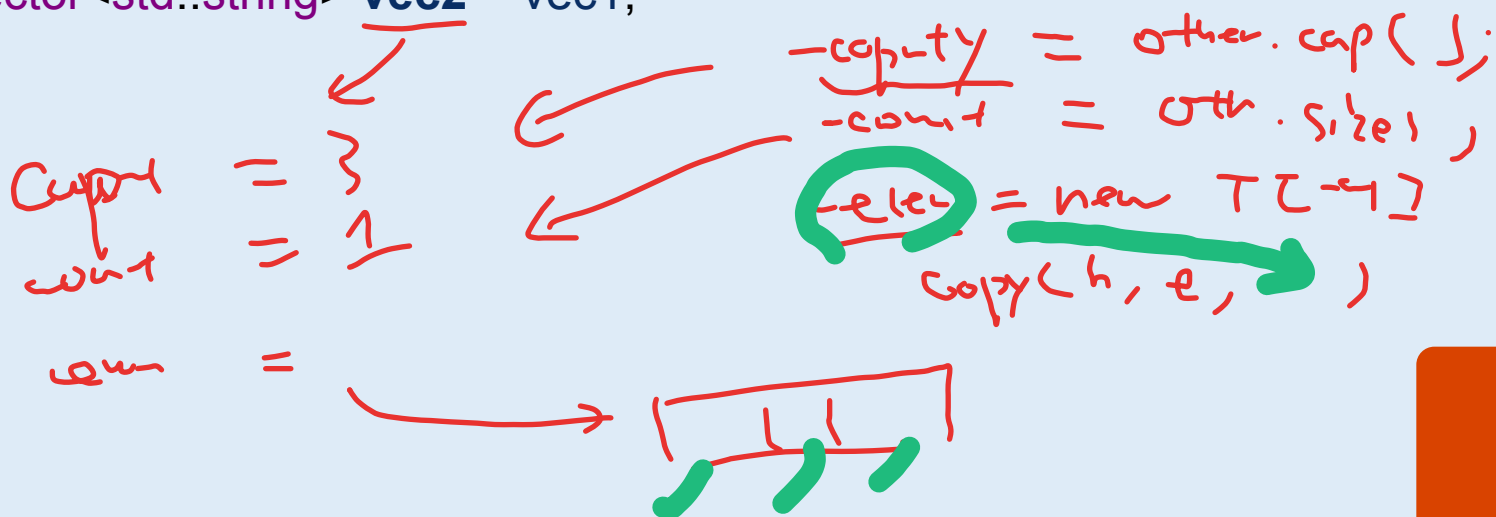
COPY-ASSIGN

特殊成员函数 Special Member Function



```
vector<std::string> vec1(3, "A");  
vector<std::string> vec2 = vec1;
```

vector(con T & other)



特殊成员函数 Special Member Function

Copy Assignment

```
vector<std::string> vec1(3, "A");  
vector<std::string> vec2(3, "B");  
vec2 = vec1;
```



移动语义

MOVE SEMANTICS

移动语义 Move Semantics

右值引用 vs 左值引用

```
vector<ValueType> &  
vector<ValueType> &&
```


移动语义 Move Semantics

- Move Constructor

```
vector<std::string> vec1(3, "A");  
vector<std::string> vec;  
vec = returnByVal(vec1);
```

移动语义 Move Semantics

Move Assignment

```
vector<std::string> vec1(3, "A");  
vector<std::string> vec2(3, "B");  
vector<std::string> vec;  
vec = vec1 + vec2;
```



THANKS

CONGRATULATIONS