

第8章 非静态数据成员 默认初始化

《现代C++语言核心特性解析》 谢丙堃

使用默认初始化

- 基本语法

```
class X {  
public:  
    X() {}  
    X(int a) : a_(a) {}  
    X(double b) : b_(b) {}  
    X(const std::string &c) : c_(c) {}  
private:  
    int a_ = 0;  
    double b_{ 0. };  
    std::string c_{ "hello world" };  
};
```

- 传统方法

```
class X {  
public:  
    X() : a_(0), b_(0.), c_("hello world") {}  
    X(int a) : a_(a), b_(0.), c_("hello world") {}  
    X(double b) : a_(0), b_(b), c_("hello world") {}  
    X(const std::string &c) : a_(0), b_(0.), c_(c) {}  
private:  
    int a_;  
    double b_;  
    std::string c_;  
};
```

初始化优先级

- 初始化列表高于默认初始化

```
class X {  
public:  
    X() : a_(1) {  
        std::cout << a_;  
    }  
private:  
    int a_ = 0;  
};  
  
int main()  
{  
    X x; // 输出为1  
}
```

位域的默认初始化

- 基本语法

```
struct S {  
    int y : 8 = 11;  
    int z : 4 {7};  
};
```



感谢聆听
欢迎关注