



C++primer学习笔记（7-8章）

第七章 类

类的基本思想：数据抽象 + 封装

- 数据抽象：依赖于接口和实现的分离式编程技术
- 封装：实现了类的接口和实现的分离

this 指针：1.每个对象都拥有的指针，用于指向本对象的地址；

2.默认情况下，this 的类型是指向类类型的非常量版本的常量指针；

3.只能在成员函数中使用，不能再静态函数或者全局函数中使用。

- 若存在一个 MyClass 类，则 this 指针类型为 MyClass *const，故不能再一个常量对象上调用成员函数，只能调用常量成员函数

```
//常量成员函数事例
class MyClass{
    .....
public:
    //常量成员函数
    int getX() const
    {
        return x;
    }
};
```

C++

构造函数：用于初始化对象；无返回类型，不能声明为 const

- 默认构造函数初始化规则：如果存在类内初始值，用其初始化成员，否则默认初始化该成员。
- C++11 中需要默认行为，可通过在参数列表后加=default();
- C++11 支持委托构造函数
- explicit 关键字用于抑制构造函数隐式转换
- 隐式的调用默认构造函数时不要使用圆括号

析构函数：对象过期时，程序自动调用的一个特殊的成员函数

- 当创建的是自动存储类对象，析构函数在程序执行完代码块时被调用
- 通过 new 创建出来的对象，当用 delete 释放内存时析构函数被调用

```
//初始化事例
class MyClass{
    int x;
    int y;
public:
    MyClass()=default;
    MyClass(int a,int b):x(a),y(b){}
    explicit MyClass(int a):x(a){}
    ~MyClass(){}
};
```

C++

内联函数：inline 关键字标识，定义位于类声明中的函数都将自动成为内联函数

对象初始化方式：显式初始化和隐式初始化

C++

```
MyClass myClass=MyClass(1,2);//显式初始化  
MyClass myClass(1,2);//隐式初始化
```

访问控制符

- public: 整个程序内可访问
- private: 可被类的成员访问
- protected: 基类的保护成员可以在派生类的成员函数中被使用

友元：允许其他类访问或者函数访问其非公有成员，友元声明只能出现在类内部

- 即使在类的内部定义友元函数，也需要在函数外部提供相应的声明使得函数可见

可变数据成员：mutable 关键字修饰，永远不会是 const

聚合类：用户能够直接访问其成员

- 聚合类满足：
 1. 所有成员都是 public;
 2. 没有定义任何构造函数;
 3. 没有类内初始值;
 4. 没有基类，没有 virtual 函数

类的静态成员

- 通过声明之前加上 static 关键字，使用 :: 访问
- 可以使用静态成员作为默认实参

第八章 IO 库

IO 类

- iostream 用于读写流的基本类型；fstream 用于读写命名文件的基本类型；sstream 用于读写 string 对象的类型；
- io 对象无拷贝和赋值

iostream 的 8 个自动创建的流对象

cin(wcin)	标准输入流
cout (wcout)	标准输出流
cerr (wcerr)	标准错误流
clog(wclog)	标准错误流

刷新输出缓冲区

- endl 换行然后刷新缓冲区
- ends 插入空字符然后刷新缓冲区
- flush 刷新缓冲区
- unitbuf 所有输出操作都会立即刷新缓冲区 nunitbuf 回到正常缓冲模式
- 任何试图从输入流读取数据的操作都会先刷新关联的输出流

文件流

- ifstream 读取文件数据 ostream 写入数据 fstream 读写给定文件

文件模式

in	读方式
out	写方式
app	写操作定义到文件末尾
ate	打开文件后定义到末尾
trunc	截断文件
binary	以二进制打开

```
//文件流事例  
ofstream file('file',ofstream::out|ofstream::app);
```

C++

string 流

·istringstream 从 string 读取数据 ostringstream 向 string 写入数据 stringstream 读写 string 数据

流状态：由三个 ios_base 组成—eofbit、badbit、failbit