石家庄铁道大学

**软件工程上机实验报告**

**C++编码规范**

**学 院 信息科学与技术学院**

**年 级 2017 级**

**班 级 信1706**

**姓 名 李杨悦**

**学 号 20173660**

**2019年06月15日**

目录

**[一、文件排版方面](#_Toc6610_WPSOffice_Level1)** **[1](#_Toc6610_WPSOffice_Level1)**

[1. 包含头文件](#_Toc6925_WPSOffice_Level2) [1](#_Toc6925_WPSOffice_Level2)

[2. h和cpp文件](#_Toc10272_WPSOffice_Level2) [1](#_Toc10272_WPSOffice_Level2)

[3. 文件结构](#_Toc26469_WPSOffice_Level2) [1](#_Toc26469_WPSOffice_Level2)

[4. 空行](#_Toc28173_WPSOffice_Level2) [2](#_Toc28173_WPSOffice_Level2)

**[二、注释方面](#_Toc6925_WPSOffice_Level1)** **[2](#_Toc6925_WPSOffice_Level1)**

[1. 文件头注释](#_Toc25819_WPSOffice_Level2) [2](#_Toc25819_WPSOffice_Level2)

[2. 函数注释](#_Toc6028_WPSOffice_Level2) [2](#_Toc6028_WPSOffice_Level2)

**[三、命名方面](#_Toc10272_WPSOffice_Level1)** **[2](#_Toc10272_WPSOffice_Level1)**

[１. 原则](#_Toc21465_WPSOffice_Level2) [2](#_Toc21465_WPSOffice_Level2)

[２. T,C,M,R类](#_Toc5316_WPSOffice_Level2) [3](#_Toc5316_WPSOffice_Level2)

[３. 函数名](#_Toc22466_WPSOffice_Level2) [3](#_Toc22466_WPSOffice_Level2)

[５. 成员变量](#_Toc2041_WPSOffice_Level2) [3](#_Toc2041_WPSOffice_Level2)

[６. 局部变量](#_Toc11379_WPSOffice_Level2) [4](#_Toc11379_WPSOffice_Level2)

[７. 全局变量](#_Toc32152_WPSOffice_Level2) [4](#_Toc32152_WPSOffice_Level2)

[８. 类名](#_Toc29852_WPSOffice_Level2) [4](#_Toc29852_WPSOffice_Level2)

[９. 风格兼容性](#_Toc7586_WPSOffice_Level2) [4](#_Toc7586_WPSOffice_Level2)

**[四、代码风格方面](#_Toc26469_WPSOffice_Level1)** **[4](#_Toc26469_WPSOffice_Level1)**

[1. Tab和空格](#_Toc20490_WPSOffice_Level2) [4](#_Toc20490_WPSOffice_Level2)

[2. 类型定义和大括号](#_Toc26965_WPSOffice_Level2) [5](#_Toc26965_WPSOffice_Level2)

[3. 函数](#_Toc8475_WPSOffice_Level2) [5](#_Toc8475_WPSOffice_Level2)

[4. 代码块](#_Toc28490_WPSOffice_Level2) [5](#_Toc28490_WPSOffice_Level2)

[5. else](#_Toc25325_WPSOffice_Level2) [5](#_Toc25325_WPSOffice_Level2)

[6. 代码行](#_Toc29026_WPSOffice_Level2) [5](#_Toc29026_WPSOffice_Level2)

[7. switch语句](#_Toc15893_WPSOffice_Level2) [6](#_Toc15893_WPSOffice_Level2)

[8. 循环](#_Toc4919_WPSOffice_Level2) [6](#_Toc4919_WPSOffice_Level2)

[9. 类](#_Toc14290_WPSOffice_Level2) [6](#_Toc14290_WPSOffice_Level2)

[10. 宏](#_Toc7401_WPSOffice_Level2) [7](#_Toc7401_WPSOffice_Level2)

[11. goto](#_Toc30750_WPSOffice_Level2) [7](#_Toc30750_WPSOffice_Level2)

**[五、类型](#_Toc28173_WPSOffice_Level1)** **[7](#_Toc28173_WPSOffice_Level1)**

**[六、表达式](#_Toc25819_WPSOffice_Level1)** **[7](#_Toc25819_WPSOffice_Level1)**

**[七、函数](#_Toc6028_WPSOffice_Level1)** **[8](#_Toc6028_WPSOffice_Level1)**

[１. 引用](#_Toc32127_WPSOffice_Level2) [8](#_Toc32127_WPSOffice_Level2)

[２. 常量成员函数](#_Toc17679_WPSOffice_Level2) [8](#_Toc17679_WPSOffice_Level2)

[３. 返回值](#_Toc20792_WPSOffice_Level2) [8](#_Toc20792_WPSOffice_Level2)

[４. 内联函数](#_Toc1628_WPSOffice_Level2) [8](#_Toc1628_WPSOffice_Level2)

[５. 函数参数](#_Toc4994_WPSOffice_Level2) [9](#_Toc4994_WPSOffice_Level2)

**[八、类](#_Toc21465_WPSOffice_Level1)** **[9](#_Toc21465_WPSOffice_Level1)**

[1. 构造函数](#_Toc31245_WPSOffice_Level2) [9](#_Toc31245_WPSOffice_Level2)

[2. 纯虚函数](#_Toc8552_WPSOffice_Level2) [9](#_Toc8552_WPSOffice_Level2)

[3. 构造和析构函数](#_Toc19351_WPSOffice_Level2) [9](#_Toc19351_WPSOffice_Level2)

[4. 成员变量](#_Toc6328_WPSOffice_Level2) [10](#_Toc6328_WPSOffice_Level2)

[5. 成员函数](#_Toc2297_WPSOffice_Level2) [10](#_Toc2297_WPSOffice_Level2)

[6. 继承](#_Toc16700_WPSOffice_Level2) [10](#_Toc16700_WPSOffice_Level2)

[7. 友元](#_Toc3472_WPSOffice_Level2) [10](#_Toc3472_WPSOffice_Level2)

**[九、错误处理](#_Toc5316_WPSOffice_Level1)** **[11](#_Toc5316_WPSOffice_Level1)**

**[十、性能](#_Toc22466_WPSOffice_Level1)** **[11](#_Toc22466_WPSOffice_Level1)**

**[十一、兼容性](#_Toc2041_WPSOffice_Level1)** **[12](#_Toc2041_WPSOffice_Level1)**

**一、文件排版方面**

**1. 包含头文件**

　• 先系统头文件，后用户头文件。  
　• 系统头文件，稳定的目录结构，应采用包含子路径方式。  
　• 自定义头文件，不稳定目录结构，应在dsp中指定包含路径。  
　• 系统头文件应用：#include <xxx.h>  
　• 只引用需要的头文件。

**2. h和cpp文件**

　• 头文件命名为.h，；C++文件命名为.cpp  
　• 文件名用大小写混合，或者小写混合。例如infoview.cpp。不要用无意义的名称：例 如XImage；  
　• 头文件，首先是全局变量，全局常量，其次是宏定义代码块，类型定义，类定义，内联部分然后是包含代码块，。  
　• CPP文件，包含全局变量，函数定义。

**3. 文件结构**

　• 文件应包含文件头注释和内容。  
　• 函数体类体之间原则上用2个空行，特殊情况下可用一个或者不需要空行。

**4. 空行**

　• 文件头、控制块,#include部分、宏定义部分、class部分、全局常量部分、全局变量部分、函数和函数之间，用两个空行。

**二、注释方面**

**1. 文件头注释**

　• 作者，文件名称，文件说明，生成日期(可选)

**2. 函数注释**

　• 关键函数必须写上注释，说明函数的用途。  
　• 除了特别情况，注释写在代码之前，不要放到代码行之后。  
　• 善未实现完整的代码，或者需要进一步优化的代码，应加上 // TODO …

**三、命名方面**

**１. 原则**

　• 同一性：在编写一个子模块或派生类的时候，要遵循其基类或整体模块的命名风格，保持命名风格在整个模块中的同一性。  
　• 标识符组成：标识符采用英文单词或其组合，应当直观且可以拼读，可望文知意，用词应当准确，避免用拼音命名。  
　• 最小化长度 && 最大化信息量原则：在保持一个标识符意思明确的同时，应当尽量缩短其长度。  
　• 避免过于相似：不要出现仅靠大小写区分的相似的标识符，例如”Function”和”function”等等。  
　• 避免在不同级别的作用域中重名：程序中不要出现名字完全相同的局部变量和全局变量，尽管两者的作用域不同而不会发生语法错误，但容易使人误解。  
　• 正确命名具有互斥意义的标识符：用正确的反义词组命名具有互斥意义的标识符，如:”setname”和”getname”  
　• 避免名字中出现数字编号：尽量避免名字中出现数字编号，如Value1,Value2等，除非逻辑上的确需要编号。

**２. T,C,M,R类**

　• T类表示简单数据类型，不对资源拥有控制权，在析构过程中没有释放资源动作。  
　• C表示从CBase继承的类。该类不能从栈上定义变量，只能从堆上创建。  
　• M表示接口类。  
　• R是资源类，通常是系统固有类型。除了特殊情况，不应在开发代码中出现R类型。

**３. 函数名**

　• M类的函数名称应采用handlexxx名，不推荐采用java风格，除了标准c风格代码，不推荐用下划线.。  
　• Leave函数，用后缀L。  
　• Leave函数，且进清除栈，用后缀LC。  
　• Leave函数，且删除对象，用后缀LD。

**５. 成员变量**

　• 成员变量用m最为前缀。

**６. 局部变量**

　• 循环变量和简单变量采用简单小写字符串即可。例如，int i  
　• 指针变量用p打头，例如void \*p1；

**７. 全局变量**

　• 全局变量用g最为前缀。

**８. 类名**

　• 类和对象名应是名词。  
　• 实现行为的类成员函数名应是动词。  
　• 类的存取和查询成员函数名应是名词或形容词。

**９. 风格兼容性**

　• 对于移植的或者开源的代码，可以沿用原有风格，不用C++的命名规范。

**四、代码风格方面**

**1. Tab和空格**

　• 每一行开始处的缩进只能用Tab，不能用空格，输入内容之后统一用空格。除了最开始的缩进控制用Tab，其他部分为了对齐，需要使用空格进行缩进。这样可以避免在不同的编辑器下显示不对齐的情况。  
　• 在代码行的结尾部分不能出现多余的空格。  
　• 不要在“：：”，“->”，”.”前后加空格。  
　• 不要在“，”，“；”之前加空格。

**2. 类型定义和大括号**

　• 类，结构，枚举，联合：大括号另起一行

**3. 函数**

　• 函数体的{需要新起一行，在{之前不能有缩进。  
　• 除了特别情况，函数体内不能出现两个空行。  
　• 除了特别情况，函数体内不能宏定义指令。  
　• 在一个函数体内，逻揖上密切相关的语句之间不加空行，其它地方应加空行分隔。

**4. 代码块**

　•等语句自占一行，执行语句不得紧跟其后。不论执行语句有多少都要加 “{ }” 。这样可以防止书写和修改代码时出现失误。  
　• if、for、while、do、try、catch的括号和表达式，括号可紧挨关键字，这样强调的是表达式。

**5. else**

• if语句如果有else语句，if语句结束后另起一行写else

**6. 代码行**

　• 一行代码只做一件事情，如只定义一个变量，或只写一条语句。这样的代码容易阅读，并且方便于写注释。  
　• 多行变量定义，将变量竖向对齐。  
　• 代码行最大长度宜控制在一定个字符以内，能在当前屏幕内全部可见为宜。

**7. switch语句**

　• case关键字应和switch对齐。  
　• case子语句如果有变量，应用{}包含起来。  
　• 如果有并列的类似的简单case语句，可考虑将case代码块写为一行代码。  
　• 简单的case之间可不用空行，复杂的case之间应考虑用空行分割开。  
　• case字语句的大括号另起一行，不要和case写到一行。  
　• 为所有switch语句提供default分支。  
　• 若某个case不需要break一定要加注释声明。

**8. 循环**

　• 空循环可用 for( ;; ) 或者 while( 1 ) 或者 while( true )

**9. 类**

　• 类继承应采用每个基类占据一行的方式。  
　• 单继承可将基类放在类定义的同一行。如果用多行，则应用Tab缩进。  
　• 多继承在基类比较多的情况下，应将基类分行，并采用Tab缩进对齐。  
　• 重载基类虚函数，应在该组虚函数前写注释 // implement XXX  
　• 友元声明放到类的末尾。

**10. 宏**

　• 不要用分号结束宏定义。  
　• 函数宏的每个参数都要括起来。  
　• 不带参数的宏函数也要定义成函数形式。

**11. goto**

　• 尽量不要用goto。

**五、类型**

• 定义指针和引用时\*和&紧跟类型。  
• 尽量避免使用浮点数，除非必须。  
• 用typedef简化程序中的复杂语法。  
• 避免定义无名称的类型。例如：typedef enum { EIdle, EActive } TState;  
• 少用union，如果一定要用，则采用简单数据类型成员。  
• 用enum取代(一组相关的)常量。  
• 不要使用魔鬼数字。  
• 尽量用引用取代指针。  
• 定义变量完成后立即初始化，勿等到使用时才进行。  
• 如果有更优雅的解决方案，不要使用强制类型转换。

**六、表达式**

• 避免在表达式中用赋值语句。  
• 避免对浮点类型做等于或不等于判断。  
• 不能将枚举类型进行运算后再赋给枚举变量。  
• 在循环过程中不要修改循环计数器。  
• 检测空指针，用 if( p )  
• 检测非空指针，用 if( ! p )

**七、函数**

**１. 引用**

　• 引用类型作为返回值：函数必须返回一个存在的对象。  
　• 引用类型作为参数：调用者必须传递一个存在的对象。

**２. 常量成员函数**

　• 表示该函数只读取对象的内容，不会对对象进行修改。

**３. 返回值**

　• 除开void函数，构造函数，析构函数，其它函数必须要有返回值。  
　• 当函数返回引用或指针时，用文字描述其有效期。

**４. 内联函数**

　• 内联函数应将函数体放到类体外。  
　• 只有简单的函数才有必要设计为内联函数，复杂业务逻辑的函数不要这么做。  
　• 虚函数不要设计为内联函数。

**５. 函数参数**

　• 只读取该参数的内容，不对其内容做修改，用常量引用。  
　• 修改参数内容，或需要通过参数返回，用非常量应用。  
　• 简单数据类型用传值方式。  
　• 复杂数据类型用引用或指针方式。

**八、类**

**1. 构造函数**

　• 构造函数的初始化列表，应和类的顺序一致。  
　• 初始化列表中的每个项，应独占一行。  
　• 避免出现用一个成员初始化另一个成员。  
　• 构造函数应初始化所有成员，尤其是指针。  
　• 不要在构造函数和析构函数中抛出异常。

**2. 纯虚函数**

　• M类的虚函数应设计为纯虚函数。

**3. 构造和析构函数**

　• 如果类可以继承，则应将类析构函数设计为虚函数。  
　• 如果类不允许继承，则应将类析构函数设计为非虚函数。  
　• 如果类不能被复制，则应将拷贝构造函数和赋值运算符设计为私有的。  
　• 如果为类设计了构造函数，则应有析构函数。

**4. 成员变量**

　• 尽量避免使用mutable和Volatile。  
　• 尽量避免使用公有成员变量。

**5. 成员函数**

　• 努力使类的接口少而完备。  
　• 尽量使用常成员函数代替非常成员函数，const函数  
　• 除非特别理由，绝不要重新定义(继承来的)非虚函数。（这样是覆盖，基类的某些属性无初始化）

**6. 继承**

　• 继承必须满足IS-A的关系，HAS-A应采用包含。  
　• 虚函数不要采用默认参数。  
　• 除非特别需要，应避免设计大而全的虚函数，虚函数功能要单一。  
　• 除非特别需要，避免将基类强制转换成派生类。

**7. 友元**

　• 尽量避免使用友元函数和友元类。

**九、错误处理**

• 申请内存用new操作符。  
• 释放内存用delete操作符。  
• new和delete，new[]和delete[]成对使用。  
• 申请内存完成之后，要检测指针是否申请成功，处理申请失败的情况。  
• 谁申请谁释放。优先级：函数层面，类层面，模块层面。  
• 释放内存完成后将指针赋空，避免出现野指针。  
• 使用指针前进行判断合法性，应考虑到为空的情况的处理。  
• 使用数组时，应先判断索引的有效性，处理无效的索引的情况。  
• 代码不能出现编译警告。  
• 使用错误传递的错误处理思想。  
• 卫句风格：先处理所有可能发生错误的情况，再处理正常情况。  
• 嵌套do-while(0)宏：目的是将一组语句变成一个语句，避免被其他if等中断。

**十、性能**

• 使用前向声明代替#include指令。Class M;  
• 尽量用++i代替i++。即用前缀代替后缀运算。  
• 尽量在for循环之前，先写计算估值表达式。  
• 尽量避免在循环体内部定义对象。  
• 避免对象拷贝，尤其是代价很高的对象拷贝。  
• 避免生成临时对象，尤其是大的临时对象。  
• 注意大尺寸对象数组。  
• 80-20原则。

**十一、兼容性**

• 遵守ANSI C和ISO C++国际标准。  
• 确保类型转换不会丢失信息。  
• 注意双字节字符的兼容性。  
• 注意运算溢出问题。  
• 不要假设类型的存储尺寸。  
• 不要假设表达式的运算顺序。  
• 不要假设函数参数的计算顺序。  
• 不要假设不同源文件中静态或全局变量的初始化顺序。  
• 不要依赖编译器基于实现、未明确或未定义的功能。  
• 将所有#include的文件名视为大小写敏感。  
• 避免使用全局变量、静态变量、函数静态变量、类静态变量。在使用静态库，动态库，多线程环境时，会导致兼容性问题。  
• 不要重新实现标准库函数,如STL已经存在的。