C++头文件包含内容及顺序

1 全面了解 C++头文件中可以包含的内容，列举出相应的代码

1.1 版权声明和版本信息

版权和版本的声明位于头文件和定义文件的开头，主要内容有：版权信息，文件名称，标识符，摘要，当前版本号，作者/修改者，完成日期以及版本历史信息。

【示例】/*

```
* Copyright (c) 2018 ，北京*******有限公司
* All rights reserved.
*
* 文件名称：
* 文件标识：
* 摘 要：
*
* 当前版本：
* 作 者：
* 完成日期：
*
* 取代版本：
* 原作者 ：
* 完成日期：
*/
```

1.2 编译预处理指令

①文件包含指令：可以包含系统标准库的头文件、用户自己定义的头文件以及引用的第三方提供的标准件的头文件

【示例】#include <iostream>

```
#include “HeaderName.h”
```

②宏定义指令

【示例】#define MaxSize 100

③宏定义取消指令

【示例】#undef MaxSize

④条件编译指令

【示例】#ifdef / #ifndef

```
/*Code*/
#endif
```

【示例】#if <表达式>

```
/*Code*/
#elif
/*Code*/
#else
/*Code*/
```

1.3 定义和声明

①结构和枚举类型定义

【示例】struct POINT{
        int px;
        int py;
    };

【示例】enum DAY{
        MON=1, TUE, WED, THU, FRI, SAT, SUN
    };

②typedef 定义

【示例】typedef struct Node *PtrToNode

③具名常量定义

【示例】const int g_int_a = 1000;

④类的声明

【示例】class Person
    {
    private:
        int age;
    public:
        Person();
        Person(int age);
        int NextyearAge();
    };

⑤外部变量声明

【示例】extern int variable;

⑥函数声明

【示例】int FunctionName(int param);

⑦内联函数定义

【示例】inline double square(double x){
        return x*x;
    }

⑧模板定义

【示例】

```
template <class T>
class Temp
{
public:
    void set_value(const T& rT);
protected:
private:
    T m_value;
};

template <class T>
```

```
void Temp<T>::set_value(const T& rT){
    m_value = rT;
}
```

⑨命名空间

【示例】

```
/*compare.h*/
namespace compare{
    double max(const double* data,int size);
    double min(const double* data,int size);
}
/*compare.cpp*/
#include "compare.h"
double compare::max(const double* data,int size){
    double result=data[0];
    for(int i=1;i<size;i++)
        if(result<data[i])
            result=data[i];
        return result;
}
double compare::min(const double* data,int size){
    double result=data[0];
    for(int i=1;i<size;i++)
        if(result>data[i])
            result=data[i];
        return result;
}
```

2 C++头文件的包含顺序

2.1《Google C++ 编程风格指南》中的观点：主要解决隐藏依赖问题

所谓的隐藏依赖就是一个头文件依赖其他文件，使用标准的头文件包含顺序可增强可读性，避免隐藏依赖，次序如下：C 库、C++库、其他库的.h、项目内的.h。项目内头文件应按照项目源代码目录树结构排列，并且避免使用 UNIX 文件路径.（当前目录）和..（父目录）

举例来说，xxx.cpp 文件中的头文件的包含顺序：

- xxx.h
- C 系统文件
- C++ 系统文件
- 其他库的 .h 文件
- 本项目内 .h 文件

2.2《C++编程思想》中的观点：能够很清楚地知道自己定义的接口是否和系统库以及第三方库发生冲突

头文件被包含的顺序是从“最特殊到最一般”。这就是，在本地目录的任何头文件首先被包含。然后是我们自己的所有“工具”头文件，随后是第三方库头文件，接着是标准 C++ 库头文件和 C 库头文件。

这样能够保证.h 文件的组成部分不被它自身解析，可以避免潜在的使用错误。因为被自身解析缺乏明确提供的声明或定义。在.c 文件的第一行包含.h 文件能确保所有对于构件的物理界面重要的内部信息块都在.h 中（如果的确是缺少了某些信息块，一旦编译这个.c 文件时就可以发现这个问题）。

如果包含头文件的顺序是“从最特殊到最一般”，如果我们的头文件不被它自己解析。我们将马上找到它，防止麻烦事情发生。

举例来说，xxx.cpp 文件中的头文件的包含顺序：

- xxx.h
- 本项目内 .h 文件
- 其他库的 .h 文件
- C++ 系统文件
- C 系统文件