# 一：使用驼峰拼写法

驼峰拼写法分两种：

大驼峰拼写法：第一个词的首字母，以及后面每个词的首字母都大写

如ActivityRankView

小驼峰拼写法：第一个词的首字母小写，后面每个词的首字母大写

如 activityRankView

# 二：命名规则

1. 根据变量、类、接口、命名空间等具体作用或者用途，使用描述它意义的单词或词组拼接而成

# 三：命名空间 NameSpace 命名规范

1.使用大驼峰拼写法

# 四：类Class 命名规范

1.使用大驼峰拼写法

2.文件名应和类名相同

3.不使用前缀，例如C表示Class，应该使用 FileStream 而不是 CFileStream

4.不使用下划线

# 五：接口 Interface 命名规范

1. 使用大驼峰拼写法
2. 在名称前加上Interface 首字母大写I 来表示类型是 Interface 如 ISprite
3. 不要使用下划线
4. 文件名应该和接口名相同

# 六：参数 Attribute 命名规范

1. 使用小驼峰写法
2. 可以在首字母前加下划线如：\_number
3. 可以在字母中间加下划线如：m\_number、number\_of\_car

# 七：枚举 Enumeration Type 命名规范

1. 使用大驼峰写法
2. 可以在Enum 类型名称后加上 Type 后缀如：GoodsPriceType

**八：不要使用单字母变量**

1. 像 i，j，k 这样单字母变量只在 for 循环中使用

**九：不要写行数太多的函数**

1. 函数逻辑最好简短，简单明了，清晰易读，过渡复杂难度的逻辑，是坏代码

不要为了写高性能的代码而将代码写的高深难以理解

1. 要尽量做到单一职责，不要在一个函数内做过多处理
2. 如果行数过多，考虑拆分函数
3. 函数形参不要太多，最好4个以内，多了不容易记忆、使用以及容易出错，如果函数参数需求过多，则考虑创建 Struct 或者 Class 添加多个变量，最后传递一个 Struct 或者一个 Class 对象
4. 相同的逻辑不要在多个函数中重复的实现，如有必要，考虑将函数抽取出来，在一个地方实现，因为当需求修改的时候，不能保证把所有的函数都做相同的修改，容易引出bug