**编码规范**

**一 命名空间**

　　<公司名称>.(<产品名称>|<相关技术>)[.<用途>] [.<子命名空间>]

**二 代码风格**

* 花括号“{}”不允许省略，即使只有一段代码。
* 不允许省略访问修饰符。
* 类型默认是密封的。
* 不允许公开字段。
* 使用括号“()”来强调运算符优先级。

**三 命名规范**

（一） 类、结构和接口的命名

* 使用名词或名词短语。
* 使用Pascal方式。
* 在接口名称前加上前缀“I”。
* 考虑在派生类末尾使用基类的名字。
* 如果该类仅仅为了实现某个接口，那么请保持其与接口命名的统一。
* 如果从.NET 框架中存在的类型派生的类型，应该遵循以下规范：

|  |  |
| --- | --- |
| 基类 | 派生类 |
| System.Attribute | 要给自定义的特性添加“Attribute”后缀 |
| System.Delegate | 要给用于事件处理的委托添加“EventHandler”后缀   要给用于事件处理之外的那些委托添加“Callback”后缀   不要给委托添加“Delegate”后缀 |
| System.EventArgs | 要添加“EventArgs”后缀 |
| System.Exception | 要添加“Exception”后缀 |
| IDictionary，IDictionary<T,V> | 要添加“Dictionary”后缀 |
| IEnumerable，ICollection，IList，   IEnumerable<T>，ICollection<T>，IList<T> | 添加“Collection”后缀 |
| System.IO.Stream | 添加“Stream”后缀 |
| CodeAccessPermission，IPermission | 添加“Permission”后缀 |

（二） 成员的命名

|  |  |  |
| --- | --- | --- |
| 成员 | 大小写 | 规范 |
| 方法 | Pascal（公开）、Camel（私有） | 用动词或动词短语命名 |
| 属性 | Pascal | 用名词、名词短语或形容词来命名   集合属性应该使用复数形式，而不是添加后缀   用“Is”、“Can”、“Has”等表示布尔属性   可以用属性的类型名来命名属性 |
| 事件 | Pascal | 使用动词或动词短语来命名事件   用现在时和过去时来区分前置和后置事件 |
| 字段 | Camel（私有） | 要用名词、名词短语或形容词来命名   不要加任何前缀 |

（三） 参数的命名

* Camel风格。
* 要使用left和right来命名重载的二元操作符的参数——如果参数没有具体的含义。
* 要使用value来命名重载的一元操作符的参数——如果参数没有具体的含义。
* 不要在参数中使用数字编号。
* 尽量使用描述性的名字命名泛型类型参数，并在前面使用“T”前缀。

（四） 常量、变量的命名

* 常量——所有单词大写并用“\_”分隔。
* 局部变量——Camel风格。

（五） 枚举的命名

* Pascal风格。
* 使用名词的复数形式来命名标记枚举。
* 不要添加“Enum”或“Flag”后缀。
* 不要给枚举类型值的名字加前缀。

（六） 资源的命名

* Pascal风格。
* 仅使用字母、数字和下划线。
* 在命名异常消息的资源时，资源标识符应该是异常类型名加上简短的异常标识符。

（七） 数据库命名

* 表——“模块名\_表名”。
* 字段——bool类型用“Is”、“Can”、“Has”等表示；日期类型命名必须包含“Date”；时间类型必须包含“Time”。
* 存储过程——使用“proc\_”前缀。
* 视图——使用“view\_”前缀。
* 触发器——使用“trig\_”前缀。

（八） XML命名

　　节点名称使用Pascal风格，属性名称使用Camel风格。

**四 注释**

* 对接口和复杂代码块必须进行注释。
* 修改代码时保持注释同步。
* 未完成的功能使用TODO标记。
* 修改他人代码时要先注释对方代码，并写明修改原因，不允许随便删除他人代码。
* 发布前移除无用注释。

**五 异常处理**

* 原则上只允许显示抛出InvalidOperationException、ArgumentException、ArgumentNullException和ArgumentOutOfRangeException四种异常类型。
* 在自定义异常时，必须使用VS提供的代码模板来创建自定义异常。