

程序编码规范

制作:

审核:

批准:

发布日期:

江苏京海博源信息技术有限公司

修订历史记录

日期	版本	说明	作者
2011-7-21	<1.0>	编写	徐新

一、 目的

为了保证编写出的程序都符合相同的规范,保证一致性、统一性而建立的程序编码规范。

二、 范围

适用于研发部的软件开发工作。

三、 定义

定义: 无

四、 职责

- ◆ 所有开发人员必须按照此规范进行系统的编码工作,项目经理职责对项目中各程序编码进行监督、检查。
- ◆ 测试人员在进行程序验收时,必须按照此规范进行程序的验收工作,对不按照此规范进行程序编码的相应的编码人员,将进行过失单处罚,并通知相应的项目经理。
- ◆ 所有人员如对编码规范有异议,可提请项目组讨论同意后修改此规范,以按照新规范进行开发工作。

五、 规范内容

1、 代码格式规范

- ◆ 所有的缩进为 4 个空格。
- ◆ 在代码中垂直对齐左括号和右括号。
- ◆ 代码行的最大长度。为了防止在阅读代码时不得不滚动源代码编辑器,每行代码在 1024*800 的显示频率下不得超过一显示屏(136 个英文字符)。如果包括注释在内不得超过 1.5 显示屏的长度(176 个英文字符)。
- ◆ 当一行被分为几行时,通过将串联运算符放在每一行的末尾而不是开头,清楚地表示没有后面的行是不完整的。例如:

```
string strSql  ="INSERT INTO PAS_CM_RewardPunlish(RPLevelDetail,RPType, "+
                "UserGUID,RPID,Memo,RPReason,RPDate,Status,AddIP, "+
                "RPType,UserGUID,RPID,Memo,RPReason,RPDate,Status,AddIP";
```
- ◆ 编写 SQL 语句时,对于关键字使用全部大写;对于数据库元素(如表、列和视图)使用大小写混合。
- ◆ 将大的复杂代码节分为较小的、易于理解的模块。为了便于代码的阅读及维护,规定每个方法、函数的代码最长不得超过 100 行。

2、 接口规范

- ◆ 接口的命名规范如下:
 - 给接口名称加上字母 I 前缀,以指示该类型为接口。
 - 用名词或名词短语,或者描述行为的形容词命名接口。例如,接口名称

IComponent 使用描述性名词。接口名称 **ICustomAttributeProvider** 使用名词短语。名称 **IPersistable** 使用形容词。各英文单词或缩写单词的首字母必须大写。接口中使用的英文单词数不得超过三个英文单词。

- 使用 Pascal 大小写。
- 少用缩写。
- 在定义类/接口对（其中类是接口的标准实现）时使用相似的名称。两个名称的区别应该只是接口名称上有字母 I 前缀。使用要编写 Employee 类的接口，则其接口为 IEmployee。这样在编码时就能很容易的辨认出某类的接口及某接口对应的类。
- 不要使用下划线字符 _。

◆ 接口方法的命名：

- 接口方法的名称 = 代表接口功能的动词 或 由代表接口功能的动词+代表接口操作对象的名称组成。
- 其中，动词由一个英文单词组成。
- 一般不允许使用英文的缩写。
- 操作对象的名称由一个或三个以下（包括三个）英文单词或英文单词缩写组成。
- 每个英文单词或缩写单词的首字母应大写。

例如：

得到用户名的接口方法：GetUserName

更改用户口令的接口方法：ChangePassword

用户管理接口 IEmployee 增加用户的接口方法：Add

得到用户资料的接口方法：GetUserInfor

系统默认的一些接口方法的定义如下：

接口方法功能	接口方法名称
新增数据	Add
更新数据	Update
删除数据	Delete
查询数据	GetMyInfo

3、类的命名规范

◆ 模型层类（实体类）的命名：

- 类名：数据库表名（不含前后缀，且是单数） + Infor
- 其成员只有属性

◆ 数据访问层类的命名：实体类（不含 Infor）+ Service

◆ 业务逻辑层的命名：实体类（不含 Infor）+ Manager

4、类的方法命名规范

- ◆ 方法名应该能够标识事物的特性，使用名词或名词短语命名类。
- ◆ 方法名尽量不使用缩写，除非它是众所周知的。
- ◆ 方法名可以由三个以下的单词或单词缩写组成，但通常不应多于三个。每个英文单词或缩写单词的首字母必须大写。
- ◆ 使用 Pascal 大小写。

- ◆ 不要使用类型前缀，如在类名称上对类使用 C 前缀。例如，使用类名称 FileStream，而不是 CFileStream。
- ◆ 不要使用下划线字符 (_)。
- ◆ 有时候需要提供以字母 I 开始的类名称，虽然该类不是接口。只要 I 是作为类名称组成部分的整个单词的第一个字母，这便是适当的。例如，类名称 IdentityStore 是适当的。
- ◆ 参数命名使用 Camel 命名规则，首字母小写。

5、类属性的命名规范

- ◆ 所有的类属性必须使用 get、set 语句定义。不允许 public 类型的字段。
- ◆ 对于实体表的属性名称与对应的数据库中的表字段名称完全一致。
- ◆ 属性的名称应该能够标识属性所表达事物的特性，使用名词或名词短语命名属性。
- ◆ 属性的名称尽量不使用缩写，除非它是众所周知的。
- ◆ 属性的名称可以由三个以下的单词或单词缩写组成，但通常不应多于三个。每个英文单词或缩写单词的首字母必须大写。
- ◆ 使用 Pascal 大小写。
- ◆ 不要使用下划线字符 (_)。

6、注释规范

- ◆ 文件注释规范

文件注释必须放在文件的顶部

```

/*
 * 作用: <对类的描述>
 * 作者: <编写人>
 * 日期: <为编写日期, 格式如: 2011-07-12>
 *
 * 修改 1
 * 内容: <第一次修改内容说明>
 * 修改人:
 * 日期: <修改日期>
 *
 */

```

- ◆ 类的注释

```

/// <summary>
/// 类的描述
/// </summary>

```

- ◆ 类的属性

```

/// <summary>
/// 属性的描述
/// </summary>

```

- ◆ 方法注释规范

```

/// <summary>
/// 方法的说明

```

```
/// </summary>  
/// <param name="参数名">参数说明</param>  
/// <returns>对返回值的说明，要尽可能的具体说明返回类型的值代表什么意思  
</returns>
```

◆ 代码间注释

➤ 单行注释

```
//<这是单行注释>
```

➤ 多行注释

```
/*这是多行注释 1
```

```
这是多行注释 2
```

```
这是多行注释 3*/
```

代码中遇到语句块时必须添加注释（if,for,foreach,.....）,添加的注释必须能够说明此语句块的作用和实现手段（所用算法等等）。

7、变量的命名规范

◆ 类模块级变量

变量名为能够代表变量含义的英文单词，单词的首字母必须大写

◆ 过程级变量

使用 Camel 命名规则。对于 string 类型，要求使用 str + Pascal 命名的方式。