**使用前注意事项：**1、由于Java面向对象编程的特性, 在命名时应尽量选择名词

2、驼峰命名法（Camel-Case）: 当变量名或函式名是由一个或多个单字连结在一起，而构成的唯一识别字时，首字母以小写开头，每个单词首字母大写（第一个单词除外）。

如：myFirstName---------->

3 不规范的命名方式 ,尽量避免以下命名方式

如: iCard,cId

**一 包名的书写规范 （Package）**推荐使用公司或机构的顶级域名为包名的前缀，目的是保证各公司/机构内所使用的包名的唯一性。包名全部为小写字母，且具有实际的区分意义。

1.1 一般要求  
1、选择有意义的名字，能快速地传达该类的用途。

2、所有包的命名必须采用小写英文字母。

1.2 实际应用  
应用系统中经常应用分层，Dao层（数据库访问）、Service层（业务处理）、Web层（页面控制action类）。

1、包名的前几个为固定名称, 如果是网站的话，采用网站的域名的反写，如果域名还没有确定的话，采用公司固定的几个名称。如：net.vschool

2、在包名的接下来一个单词为模块的名称。如：用户模块，包名为net.vschool.user

3、关于模块的访问操作，采用分层形式,一般分为：

Dao层操作：一般定义在net.vschool.xxx.dao 中，其中xxx为模块名称。

Service层操作：一般定义在net.vschool.xxx.service中。

web层操作：一般定义在 net.vschool.xxx.action/controller中。

如下关于用户模块的例子：

net.vschool.user

net.vschool.user.dao

net.vschool.user.action

net.vschool.user.service

**二 类名的书写规范 (Class)**类名必须使用名词，如果一个类名内含多个单词，那么各个单词第一个字母大写，后续字母小写，起伏呈驼峰状，人称驼峰式命名。给类名命名时，必须保证准确、简洁且容易理解。尽量使用完整单词，避免使用缩写词（除了大家公认的）

2.1 类的命名  
2.1.1 一般要求  
1、选择有意义的名字，能快速地传达该类的用途。

2、参照java驼峰命名法，类名的首字母必须采用大写的形式，如果类名为多词组合而成的话，那么每个词的首字母必须采用大写。如：StudentAnswer.java

3、当要区别接口类和实现类的时候，可以在类的后面加上“Impl”。

如：接口类：IUserService.java   接口实现类：UserServiceImpl

4、推荐实体类没有后缀名。

2.1.2 实际应用  
应用系统中经常应用分层，Dao层（数据库访问）、Service层（业务处理）、Web层（页面控制action类），每一层的类的名称尽量带上该层后缀。

1、Dao层

a、接口类：采用JavaBean+Interface+Dao的形式来定义,即，实体对象+Interface+Dao。

如：用户对象接口类： UserInterfaceDao，其中xxx为模块名称。

b、实现类：采用JavaBean+Interface+Impl+Dao的形式来定义,即，实体对象

+Interface+Impl+Dao。 如：用户对象实现类：UserInterfaceImplDao

2、Service层

a、接口类：采用Xxx+Interface+Service的形式来定义,即，模块+Interface+Service。

如：用户管理接口类：UserMsgInterfaceServiec

b、实现类：采用Xxx+Interface+Impl+Service的形式来定义,即，模块+Interface+

Impl+Service。如：用户管理实现类：UserMsgInterfaceImplServiec

3、Web层（action类）

a、实现类：采用县 Xxx+Operator+Action的形式来定义,即，模块+操作+Action。如

用户模块User+删除操作Delete+Action = UserDeleteAction

2.1 变量的命名  
2.2.1 普通变量  
2.2.2.1 一般要求  
1、选择有意义的名字，能快速地传达该变量的用途。

2、参照java驼峰命名法，首字母以小写开头，每个单词首字母大写（第一个单词除外）。

2.2.2.2 实际应用  
1、变量命名采用基本结构为typeVariableName，使用3字符前缀来表示数据类型。

例如，定义一个整形变量：intDocCount，其中int表明数据类型，后面为表意的英文名，每个单词首字母大写。

|  |  |  |
| --- | --- | --- |
| 数据类型或对象类型 | 变量前缀 | 备注 |
| byte | bye | 1、做数组用时，再加前缀-a,如字符串数组：astr，  2、自定义类型的变量可以采用本身的名称，把首字母改为小写。  3、采用名称要能代表在方法中的意义。如果员工列表：employeeList |
| char | chr |
| float | flt |
| boolean | bln |
| Integer/int | int |
| short | sht |
| Long/long | lng |
| Double/double | dbl |
| string | str |

2、变量使用技巧：

a、在一段函数中不使用同一个变量表示前后意义不同的两个数值。

b、除非是在循环中，否则一般不推荐使用单个字母作为变量名，i、j、k等只作为小型循环的循环索引变量。

c、避免用Flag来命名状态变量。

d、用Is来命名逻辑变量，如：blnFileIsFound。通过这种给布尔变量肯定形式的命名方式，使得其它开发人员能够更为清楚的理解布尔变量所代表的意义。

e、如果需要对变量名进行缩写时，一定要注意整个代码中缩写规则的一致性。例如，如果在代码的某些区域中使用intCnt，而在另一些区域中又使用intCount，就会给代码增加不必要的复杂性。建议变量名中尽量不要出现缩写。

2.2.2 静态变量  
1、选择有意义的名字，能快速地传达该变量的用途。

2、参照java驼峰命名法，采用全部大写的形式来书写，对于采用多词合成的变量采用“\_”来连接各单词。如：USER\_LIST

2.3 方法的命名  
2.3.1 一般要求  
1、选择有意义的名字，能快速地传达该方法的用途。

2、参照java驼峰命名法，首字母以小写开头，每个单词首字母大写（第一个单词除外）。

2.3.2 实际应用  
1、方法表示一种行为，它代表一种动作，最好是一个动词或者动词词组或者第一个单词为一个动词。

2、属性方法：以get/set开头，其后跟字段名称，字段名称首字母大写。如：getUserName()

3、数据层方法：只能以insert（插入）,delete（删除）,update（更新）,select（查找）,count（统计）开头，其他层方法避免以这个5个单词开头，以免造成误解。

4、服务层方法，根据方法的行为命名，只描述方法的意义，而不采用方法的目的命名。比如系统的添加新用户，用户可以前台注册，也可以管理员后台添加，方法会被重用，所以最好不要用使用register，采用add会更好写。避免使用与web层相关的方法。

5、Web层方法最好是贴近web的语言，如register，login，logout等方法。

**三 注释的书写规范 （Javadoc）**Java除了可以采用我们常见的注释方式（//、/\* \*/）之外，Java语言规范还定义了一种特殊的注释，也就是我们所说的Javadoc注释，以/\*\*开头，而以\*/结束， Javadoc 注释可以被自动转为在线文档，省去了单独编写程序文档的麻烦。 推荐使用。

Javadoc注释主要涉及范围：类、属性、方法。

例如：

复制代码代码如下:

package org.ietf.jgss;

import java.net.InetAddress;

import java.util.Arrays;

/\*\*

 \* 该类的整体性描述。

 \*

 \* @author 作者

 \* @version 1.0, 05/22/07

 \* @since 1.0

 \*/

public class ChannelBinding {

/\*\*

 \* 对该变量的备注信息

 \*/

private InetAddress initiator;

/\*\*

 \* 对该变量的备注信息

 \*/

private InetAddress acceptor;

/\*\*

 \* 对该变量的备注信息

 \*/

    private  byte[] appData;

    /\*\*

     \* 对该类的构造函数的备注信息。

     \*

     \* @param initAddr 对参数的备注。

     \* @param acceptAddr对参数的备注。

     \* @param appData对参数的备注。

     \*/

    public ChannelBinding(InetAddress initAddr, InetAddress acceptAddr,

              byte[] appData) {

         initiator = initAddr;

         acceptor = acceptAddr;

         if (appData != null) {

              this.appData = new byte[appData.length];

              java.lang.System.arraycopy(appData, 0, this.appData, 0,

                   appData.length);

         }

    }

    /\*\*

     \* 对该类的具体一函数的备注信息

     \*

     \* @param obj 参数的备注信息

     \* @return 返回值的备注信息

     \*/

    public boolean equals(Object obj) {

         if (this == obj)

              return true;

         if (! (obj instanceof ChannelBinding))

              return false;

         ChannelBinding cb = (ChannelBinding) obj;

         return Arrays.equals(appData, cb.appData);

    }

}

**四 其他书写规范**

4.1 Jsp页面名称的书写规范  
1．全部采用小写的英文字符和”\_ ”组成。

2．整体采用模块名+操作的形式。如：user\_view.jsp

3．Jsp页面尽可能与action的意思对应，如UserListAction 对应者user\_list.jsp

**接口：**

使用驼峰式命名。除了用名词外，还可以用形容词命名（体现其功能特性）

**方法：**

规定用动词命名，适合用驼峰式命名，但与类名的最大区别在于，首字母必须小写

**变量：**

规定为名词，其他同“方法”命名方式一样。变量名非常关键，应含有具体意义且易于理解，一般不允许使用单个字母做变量名。除非一些临时性变量，像在循环中使用到的计数器等。在使用单个字母做变量名时，一般I、J、K用来命名整形变量。

**常量：**

规定全用大写字母表示，如果名字必须用多个单词来表示，那么各单词间用“-“分隔。常量要求必须意义明确，能表达出常量的含义。