JavaWeb_JavaEE_命名规则 - 每天进步1%

命名规范
命名规范命名规范
命名规范

本规范主要针对Java开发制定的规范项目命名
项目命名项目命名
项目命名

项目创建，名称所有字母均小写，组合方式为：
com.company.projectName.component.hiberarchy。
1. projectName：项目名称
2. component：模块名称
3. hiberarchy：开发层次名称
例如：
com.company.tims.exchange.dao 类文件夹命名

类命名规范

基本命名规范：
类命名
命名规范：以大写字母开头，如果有多个单词，每个单词头字母大写
例如：UserInfo

接口命名
命名规范：以大写字母"I"开头，如果有多个单词，每个单词头字母大写
例如：IStudentInfo

接口实现类命名：
命名规范：将实现的接口名称的首字母"I"去掉，以"Impl作为结尾"，如果有多个单词，每个单

词头字母大写。
例如：StudentInfoImpl

J2EE+SSH框架命名规范

servlet类命名：
命名规范：以Servlet单词结尾
例如：LoginServlet

POJO命名：
使用hibernate自动生成的类即可

DAO类命名：
使用hibernate自动生成的类即可

Action类命名：
命名规范：Action的命名以POJO名称来制定，POJO名称Action
例如：
一个POJO名称为Diary，其对应的action为DiaryAction

ActionForm类命名：
命名规范：ActionForm的命名以POJO名称来制定，POJO名称Form
例如：
一个POJO名称为Diary，其对应的actioForm为DiaryForm

业务逻辑接口命名:
命名规范：业务逻辑接口的命名以POJO名称来制定，IPOJO名称Service
例如：
一个POJO名称为Diary，其对应的业务逻辑接口为IDiaryService

业务逻辑实现类命名:
命名规范：业务逻辑接口实现类的命名以POJO名称来制定
例如：
一个POJO名称为Diary，对应的业务逻辑接口实现类名为DiaryServiceImpl

类变量命名：
命名规范：变量名首字母必须小写，如果该变量名有多个单词组成，后面的单 词首字母大写，

单词与单词之间不要使用"_"做连接，变量名访问控制必须为私有， 可以对其增加setter与

getter方法。
例如：

```
private int studentAge;
public int getStudentAge(){
return studentAge;
}
public void setStudentAge(int studentAge) {
```

this.studentAge=studentAge;
}

常量命名：
命名规范：所有字母大写，如果有多个单词组成，单词与单词之间以” _“隔开。而且该变量

必须是公共、静态、final类型
例如：public static final String USER_NAME=”userName“;

方法命名：
命名规范：首字母必须小写，如果该变量名有多个单词组成，后面的单词首字母 大写，单词与

单词之间不要使用"_"做连接。单词不要使用名词。
例如：public int checkLogin（String name,String pwd）{}

注释规范：注释规范是整个开发规范中最为重要的组成部分，必须严格执行。
类的注释：
作用：注释整个类，简单概述该类作用。
书写规范：类的注释必须写在该类的声明语法之前。在注释中要描述该类的基 本作用，作者，

日期，版本，公司名称，版权声明。
格式：

```
/* *
* 类功能描述：（大致描述类的功能）
* @author：编写者名称
*
* @version： 类文件的版本号 从1.0开始（自己确定版本号的增改
* 情况）， 修改情况（修改时间、作者、改动情况）
*
* @see 包名.参考类名 （列出父类，引入类，每个类占一行），如果有
* 可省略
* 相关数据如：（便于理解本类的一些常量数据及某些数据的格式
* 或认为比较重要的数据，如果没有可省略）
*/
```

类的声明语法

例如：

```
/**
* <p>Title:管理员模块数据处理类 </p>
* <p>Description: 两个数相加</p>
* <p>Copyright: Copyright (c) 2007</p>
* <p>Company:华腾软件公司</p>
*
```

```
* @author 童金虎
*
* @version $Revision: 1.7 $ $Date: 2007/07/08$
*/
public class AdminDAO
```

变量、常量注释：
作用：简单描述该变量的意义。
书写规范：变量注释必须写在变量定义之前，简单描述其代表的意义。
格式：

```
/**
* 变量功能描述：（大致描述变量的功能）
*/
```

例如：

```
/**
*定义年龄变量
*/
public int age;
```

方法注释：
作用：对该方法功能简单描述，其参数、返回值意义的注解。
书写规范：方法注释必须写在方法定义之前。该注释包括：方法其功能的简单 描述，方法的参

数、返回值类型、返回值意义简单的描述。
格式：

```
/**
* 方法功能说明
* @param args (参数类型可以写在参数后，也可以省
* 略。每个参数占一行)
* @return 输出参数（多种情况写在同一行）
* @see 类#参考方法 （与此方法有调用关系的方法供参
* 考，不必每个方法都完整列出，要选择有意义的，每个
* 方法占一行）
* @exception 异常处理类（方法中能够引发的异常，每
* 个异常占一行）
*/
```

例如：

```
/**
* 修改管理员密码
* @param adminId 管理员编号
* @param oldPassword 旧密码
```

```
* @param password 新密码
* @return boolean 是否编辑成功
* @throws UserException
* @throws ServiceException
*/
public booleaneditAdminPassword(int adminId,String oldPassword,
String password) throws UserException,ServiceException;
```

Jsp页面命名：
命名规范：jsp页面名称要以小写字母开头，如果有多个单词组成，后面的单词以 大写字母开头。名称要体现出该页面的意义，最好能够与模块名称联系在一起。
例如：
login.jsp --登录页面
register.jsp --注册页面
message.jsp --客户留言页面

J2EE项目工程文件夹组织规范：
目的：规范学员web应用程序的资源组织形式，形成良好的文件组织习惯。文件的组织形式应当体现模块的划分。
根据eclipse工具的特征，项目的目录结构为：
src
----存放java文件

Java代码

```
1. WebRoot
2. |--images  --存放web程序所需的公共图片
3. |--css  --存放web程序所需的公共样式表
4. |--js  --存放web程序所需的公共js文件
5. |--commons  --存放web程序所需的公共文件
6. |--功能模块文件夹(存放与某个功能模块相关的资源)
7.       |--images  --存放与该功能模块相关的图片
8.       |--css  --存放与该模块相关的样式表文件
9.       |--js  --存放与该模块相关的js文件
10. |--jsp、html页面
11. |--WEB-INF
12.       |--classes
13.       |--lib
```

14. |--tld文件

J2EE项目提交规范

项目完成时要将项目作为一个产品交付用户，良好的项目组织规范可以使用户可以方便的找寻项目中需要的资源，同时也是一个公司专业性的体现。项目提交时，要按照下列文件格式进行提交。

项目主文件夹：

作用：存放项目其他资源文件。

命名规范：时间_班级编号_第X小组。

例如：070706_GS2T18_第四小组。

项目主文件夹下面包括以下文件夹和文件：

|--src：保存.java文件。

|--database：保存数据库的脚本文件或者数据库备份文件。

|--source：保存eclipse工程中WebRoot目录下的所有文件。

|--depend：保存编译该程序必须依赖的其他jar文件。

|--javadoc：保存所有类生成的javadoc api文档。

|--war：保存程序的归档文件

|--xx.war：已经打包好的工程文件，可以直接运行。

|--project：保存开发项目原工程代码及文件。

|--产品说明书.doc：图文方式展现该产品使用方法。

|--build.xml：ant脚本，用于生成运行的war文件。

|--项目解说.ppt：进行项目讲解的ppt（ppt仅供在校模拟项目使用，不用于其他商业用途）

注：一个完整的项目中，数据库必须有一定量的有效的测试数据来支持该程序的运行

包的命名

Java包的名字都是由小写单词组成。但是由于Java面向对象编程的特性，每一名Java程序员都可以编写属于自己的Java包，为了保障每个 Java包命名的唯一性，在最新的Java编程规范中，要求程序员在自己定义的包的名称之前加上唯一的前缀。由于互联网上的域名称是不会重复的，所以程序员一般采用自己在互联网上的域名称作为自己程序包的唯一前缀。

例如： net.frontfree.javagroup

类的命名

类的名字必须由大写字母开头而单词中的其他字母均为小写；如果类名称由多个单词组成，则每个单词的首字母均应为大写例如TestPage；如果类名称中包含单词缩写，则这个所写词的每个字母均应大写，如：XMLExample,还有一点命名技巧就是由于类是设计用来代表对象的，所以

在命名类时应尽量选择名词。
例如： Circle

方法的命名
方法的名字的第一个单词应以小写字母作为开头，后面的单词则用大写字母开头。
例如： sendMessge

常量的命名
常量的名字应该都使用大写字母，并且指出该常量完整含义。如果一个常量名称由多个单词组成，则应该用下划线来分割这些单词。
例如： MAX_VALUE

参数的命名
参数的命名规范和方法的命名规范相同，而且为了避免阅读程序时造成迷惑，请在尽量保证参数名称为一个单词的情况下使参数的命名尽可能明确。

Javadoc注释
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，它是用来记录我们代码中的API的。Javadoc注释是一种多行注释，以/**开头，而以*/结束，注释可以包含一些HTML标记符和专门的关键词。使用Javadoc 注释的好处是编写的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。
例如：

```
/**
* This is an example of
* Javadoc
*
* @author darchon
* @version 0.1, 10/11/2002
*/
```

在每个程序的最开始部分，一般都用Javadoc注释对程序的总体描述以及版权信息，之后在主程序中可以为每个类、接口、方法、字段添加 Javadoc注释，每个注释的开头部分先用一句话概括该类、接口、方法、字段所完成的功能，这句话应单独占据一行以突出其概括作用，在这句话后面可以跟随更加详细的描述段落。在描述性段落之后还可以跟随一些以Javadoc注释标签开头的特殊段落，例如上面例子中的@auther和@version，这些段落将在生成文档中以特定方式显示。

变量和常量命名

变量命名的方法采用匈牙利命名法，基本结构为scope_typeVariableName，它使用3字符前缀来表示数据类型，3个字符的前缀必须小写，前缀后面是由表意性强的一个单词或多个单词组成的名字，而且每个单词的首写字母大写，其它字母小写，这样保证了对变量名能够进行正确的断句。例如，定义一个整形变量，用来记录文档数量：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。这样，在一个变量名就可以反映出变量类型和变量所存储的值的意义两方面内容，这使得代码语句可读性强、更加容易理解。byte、int、char、long、float、 double、boolean和short。

变量类型和首字母对照关系如下表：

数据类型/对象类型 / 变量前缀 / 备注

byte bye

char chr

float flt

boolean bln 做布尔变量时，使用bln

Integer/int int

String str

Single sng

short sht

Long/long lng

Double/double dbl

Currency cur

Variant bln astr obj vnt 做布尔变量用时，用bln，做字符串数组用时，用astr，做为对象使用时，用obj，不确定时，用vnt。

对于数组，在数据类型的前缀前再增加一个a，例如字符串数组为astr。对于在多个函数内都要使用的全局变量，在前面再增加“g_”。例如一个全局的字符串变量：g_strUserInfo。

在变量命名时要注意以下几点：

- 选择有意义的名字，注意每个单词首字母要大写。
- 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。
- i、j、k等只作为小型循环的循环索引变量。
- 避免用Flag来命名状态变量。
- 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。
- 如果需要的话，在变量最后附加计算限定词，如：curSalesSum。
- 命名不相包含，curSales和curSalesSum。
- Static Final 变量的名字应该都大写，并且指出完整含义。

• 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。
• 通过在结尾处放置一个量词，就可创建更加统一的变量，它们更容易理解，也更容易搜索。例如，请使用 strCustomerFirst和strCustomerLast，而不要使用strFirstCustomer和strLastCustomer。常用的量词后缀有：First（一组变量中的第一个）、Last（一组变量中的最后一个）、Next（一组变量中的下一个变量）、Prev（一组变量中的上一个）、Cur（一组变量中的当前变量）。
• 为每个变量选择最佳的数据类型，这样即能减少对内存的需求量，加快代码的执行速度，又会降低出错的可能性。用于变量的数据类型可能会影响该变量进行计算所产生的结果。在这种情况下，编译器不会产生运行期错误，它只是迫使该值符合数据类型的要求。这类问题极难查找。
• 尽量缩小变量的作用域。如果变量的作用域大于它应有的范围，变量可继续存在，并且在不再需要该变量后的很长时间内仍然占用资源。它们的主要问题是，任何类中的任何方法都能对它们进行修改，并且很难跟踪究竟是何处进行修改的。占用资源是作用域涉及的一个重要问题。对变量来说，尽量缩小作用域将会对应用程序的可靠性产生巨大的影响。

关于常量的命名方法，在JAVA代码中，无论什么时候，均提倡应用常量取代数字、固定字符串。也就是说，程序中除0，1以外，尽量不应该出现其他数字。常量可以集中在程序开始部分定义或者更宽的作用域内，名字应该都使用大写字母，并且指出该常量完整含义。如果一个常量名称由多个单词组成，则应该用下划线“_”来分割这些单词如：NUM_DAYS_IN_WEEK、MAX_VALUE。

source代码模块 config配置文件模块 test 测试模块包命名
包命名包命名
包命名

规则o 全部小写。o 标识符用点号分隔开来。为了使包的名字更易读。如
com.leadal.shanty
常用几个包名o dao:数据层o service:逻辑层o model:持久类定义包□ 实现Serializable接口，创建serialVersionUID□ 主键统一用id标识，Long类型o web:表示层及控制层o enums:枚

举类型o config:配置文件类包o resource:资源文件包类文件命名
类文件命名类文件命名
类文件命名

尽量以英文进行类定义 所有类都以大写字母开头 组合词每个词以大写字母开头，不用下划线

或其他符号 避免使用单词的缩写，除非它的缩写已经广为人知，如HTTP变量命名
变量命名变量命名
变量命名

第一个字母小写，中间单词的第一个字母大写 不用_或&作为第一个字母。 尽量使用短而且具

有意义的单词 单字符的变量名一般只用于生命期非常短暂的变量。i,j,k,m,n一般用于
integers； c,d,e一般用于characters
如果变量是集合，则变量名应用复数。
String myName;

int[] students;

int i;

int n;

char c;

btNew;

(bt是Button的缩写)

常量命名
常量命名常量命名
常量命名

所有常量名均全部大写，单词间以‘_’隔开
int MAX_NUM; 方法命名规则
方法命名规则方法命名规则
方法命名规则

规则o 第一个单词一般是动词。o 第一个字母是小些，但是中间单词的第一个字母是大写。o

如果方法返回一个成员变量的值，方法名一般为get+成员变量名，
如若返回的值是bool变量，一般以is作为前缀。o
　如果方法修改一个成员变量的值，方法名一般为：set + 成员变量
名。
　常用动词o 获取单个对象□ loado 获取列表□ find□ listo 获取分页□ pageo 搜索□

searcho 保存□ saveo 添加□ add□ inserto 删除□ delete□ removeo 修改□ updateDAO
DAODAO
DAO命名规则
命名规则命名规则
命名规则

接口类命名以DAO结尾 实现类类命名以DAOImpl结尾 结构参考如下： dao.dom4j

UserDAOImpl.hibernate.UserDAOImpl.UserDAO Service
ServiceService
Service命名规范
命名规范命名规范
命名规范

接口类命名以Service结尾 实现类类命名以ServiceImpl结尾 结构参考如下： service.impl

.UserServiceImpl.UserService Struts Action
Struts Action Struts Action
Struts Action 命名规范
命名规范命名规范
命名规范

类命名与Action结尾 结构参考如下： web.struts.UserAction.GroupAction
JavaWeb JavaEE 命名规则

分享到：