



# EE基础与应用

翁秀木

# 任务清单

- 任务1，现有5个子任务，需上交
- 任务2，现有4个子任务，需上交
- 任务3，现有1个子任务，需上交
- 任务4，不需上交
- 任务5，不需上交
- 任务6，现有3个子任务，需上交

# 任务1 –玩转常用字符/日期工具、初识单例模式与JUnit

## [目标]

- 1) 掌握java字符与日期类中的基本方法的使用，如String、Calendar、Date、SimpleDateFormat、Character、StringBuilder、StringBuffer。
- 2) 掌握如何寻找适合的java API中的方法，以完成任务；
- 3) 掌握Apache Commons项目lang包的下载、配置和查看源代码；
- 4) 掌握如何寻找适合的Commons项目lang包中的方法，以完成任务；
- 5) 掌握单例模式（singleton）的代码编写，并用Eclipse中的Debug功能来调试singleton代码。
- 6) 掌握JUnit包的下载、配置和查看源代码，并懂得创建和运行JUnit test case类。

## 任务1 – 玩转常用字符/日期工具、初识单例模式与JUnit

### [基本要求]

- 1) 在Eclipse中开发；
- 2) 用Ctrl+Alt+F自动格式化代码；
- 3) 遵循我们的Java语言Code Style（参考附录1）；
- 4) 尽量提高代码质量。

**任务1.1** 用`java.lang.String`中的方法，比较两字符串“MILAN”和“milan”是否相等（忽略大小写）。再选用`StringBuffer`或`StringBuilder`，将两个字符串合并。再把合并后的字符串中全部变成小写。再把“lan”全替换成“ckey”。

[扩展要求]

1) 能在Eclipse中看Java API源码； 2) 不能用Apache Commons Lang包。

**任务1.2** 用Apache commons项目中的StringUtils工具类中的适当方法，找出字符串“FORZAMILAN”中有几个“A”。再思考若用java.lang.String中的方法，如何实现同样的功能。

[扩展要求]

1) 下载Apache Commons的lang包（3.2.1），并配置到Eclipse中； 2) 能在Eclipse中看StringUtils源码。

**任务1.3** 用java.util.Calendar类创建日期，使其等于2000年2月29日上午8点25分，并将其打印出来，要求格式为“2000-02-29 08:25 AM”。再用Apache Commons项目Lang包中的DateUtils类，将这个日期修改为2013年5月3日下午8点25分，并将其打印出来，要求格式为“2000/05/03 20:25”。

[扩展要求]

1) 下载Apache Commons的lang包（3.2.1），并配置到Eclipse中；2) 能在Eclipse中看DateUtils源码；3) 懂得查看SimpleDateFormat类的javadoc。

## 任务1.4 寻找生活中1对1或1对多的实例，例如蜂后与工蜂，用java实现单例模式（singleton）类，并写出调试代码。

### [要求]

- 1) 理解单例模式；
- 2) 用Eclipse的Debug功能调试该单例模式类。



**任务1.5** 用JUnit做单元测试，以测试String类的indexOf()、StringUtils类的isEmpty(CharSequence)和equals(CharSequence, CharSequence)，创建测试用例类（Test Case），并运行之。

[要求]

1) 下载JUnit3.8.1包，配置到Eclipse中，要能看JUnit源码； 2) 测试应尽量覆盖各种情况； 3) 上交测试类源码、测试结果（能看到JUnit视图）的拷屏图。

### [上交]

1. **<1>**在“任务1”目录（已创建）下，创建自己的子目录，其命名为“序号姓名”（序号为两位数字，序号和姓名间无空格），如“92翁秀木”；**<2>**，在“序号姓名”子目录下，再创建“任务1”子目录；**<3>**并在该目录下创建“任务1.1”、“任务1.2”、“任务1.3”、“任务1.4”、“任务1.5”子目录；**<4>**将解答文件上传到相应的“任务1.\*”子目录下。

2. 上交java源代码文件、debug和JUnit单元测试的拷屏图。

# 任务2 – 玩转Collections框架

## [目标]

- 1) 掌握Collections框架的使用，包括List（ArrayList与LinkedList）、Set（HashSet与TreeSet）、Map（HashMap与TreeMap）、工具类（Collections与Arrays）、Iterator与ListIterator。
- 2) 掌握何时用List或Set或Map；何时用ArrayList或LinkedList；何时用HashSet或TreeSet；何时用HashMap或TreeMap。
- 3) 掌握Apache Commons项目collections包的下载、配置和查看源代码；
- 4) 掌握如何寻找适合的Commons项目collections包内的类，以完成任务。

## 任务2 – 玩转Collections框架

### [基本要求]

- 1) 在Eclipse中开发；
- 2) 用Ctrl+Alt+F自动格式化代码；
- 3) 遵循我们的Java语言Code Style（参考附录1）；
- 4) 尽量提高代码质量。

## 任务2 – 玩转Collections框架

**任务2.1：** 有一个集合，集合中的元素是一组String类对象（其值都是英文字母）。根据业务逻辑，只需读该集合中的元素，不需增删。为使访问集合的效率更高，请选用ArrayList或LinkedList类，来实现这个集合。并使用工具类Collections中的方法，将集合中的String类对象元素按字母表倒序排列。

[扩展要求]

null

## 任务2 – 玩转Collections框架

### 任务2.2： 玩转HashMap和HashSet

1> 创建一个HashMap，往里头存放n组键值对（key-value pair），该HashMap**键的类必须是自己创建的用户类**，再用Iterator遍历HashMap中所有键值对，并将其打印输出到控制台。

2> 创建一个HashSet，往里头存放n个元素，该HashSet**元素的类必须是自己创建的用户类**，再用Iterator遍历HashSet中所有元素，并将其打印输出到控制台。

#### [扩展要求]

1> 该HashMap**不能存放重复键**； 2> 该HashSet**不能存放重复元素**。

## 任务2 – 玩转Collections框架

### 任务2.3： 玩转TreeMap和TreeSet

1> 创建一个TreeMap，往里头存放n组键值对（key-value pair），该TreeMap键的类必须是自己创建的用户类，再用Iterator遍历TreeMap中所有键值对，并将其打印输出到控制台。

2> 创建一个TreeSet，往里头存放n个元素，该TreeSet元素的类必须是自己创建的用户类，再用Iterator遍历TreeSet中所有元素，并将其打印输出到控制台。

#### [扩展要求]

可用“实现Comparable接口”和“创建Comparator类”两种方法。

## 任务2 – 玩转Collections框架

### 任务2.4： 玩转Apache Commons项目collections包

需求：创建一个ArrayList，往里头存放三个Integer元素（3， 6， 9），然后选用Commons项目collections包内的某个合适的类，将该List中的元素循环打印输出到控制台。比如，若要求依次打印4个元素，则可以打印出3， 6， 9， 3；若要求依次打印5个元素，则可以打印出3， 6， 9， 3， 6。

#### [扩展要求]

- 1) 下载Apache Commons的collections包（4.0），并配置到Eclipse中；
- 2) 能在Eclipse中看collections包源码。



## 任务2 – 玩转Collections框架

### [上交]

1. **<1>**在“任务2”目录（已创建）下，创建自己的子目录，其命名为“序号姓名”（序号为两位数字，序号和姓名间无空格），如“92翁秀木”；**<2>**在该目录下创建“任务2.1”、“任务2.2”、“任务2.3”、“任务2.4”子目录；**<4>**将解答文件上传到相应的“任务2.\*”子目录下。

2. 上交java源代码文件。

# 任务3 – 初涉Java 5

## [目标]

- 1) 了解Java 5新特性中的泛型、自动装箱拆箱、导入静态成员、增强版的for循环、标注、可变参数。
- 2) 能看得懂使用了上述特性的代码。

# 任务3 – 初涉Java 5

## [基本要求]

- 1) 在Eclipse中开发；
- 2) 用Ctrl+Alt+F自动格式化代码；
- 3) 遵循我们的Java语言Code Style（参考附录1）；
- 4) 尽量提高代码质量。

# 任务3 – 初涉Java5

1> 创建一个新的类，要求用到泛型（泛型类、泛型方法、泛型通配符）、用到可变参数。

2> 创建一个上面所创建类的JUnit测试类，要求用到标注、导入静态成员、自动装箱和拆箱、增强版的for循环。

[扩展要求]

null

# 任务3 – 初涉Java5

## [上交]

1. **<1>**在“任务3”目录（已创建）下，创建自己的子目录，其命名为“序号姓名”（序号为两位数字，序号和姓名间无空格），如“92翁秀木”；**<2>**将解答文件上传到“序号姓名”目录下。

2. 上交java源代码文件。

# 任务4 –Ant的使用

## [目标]

- 1) 了解Apache Ant的用途和基本用法。
- 2) 掌握如何编写Ant的构建配置文件，以自动编译、自动运行Junit测试类。
- 3) 掌握在eclipse中运行Ant
- 4) 掌握不依赖eclipse，如何配置并运行Ant。

# 任务4 – Ant的使用

[基本要求]

null

## 任务4 – Ant的使用

### 4.1 在eclipse中使用ant

在eclipse中，编写Ant构建文件，并运行，使得可以自动编译和运行JUnit测试类。

[扩展要求]

null



# 任务4 – Ant的使用

## [上交]

- 不需上交

# 任务5 –玩转应用服务器

## [目标]

- 1) 掌握如何在eclipse中创建动态web项目，并将其以war包格式导入导出当前工作目录。
- 2) 掌握如何在eclipse中配置tomcat、Jboss和GlassFish应用服务器，并部署web应用。
- 3) 掌握如何不依赖eclipse，启动和关闭tomcat、Jboss和GlassFish应用服务器，查看服务器日志，查看部署的应用，查看JSP转化成的Servlet java源码。
- 4) 掌握如何通过tomcat、Jboss和GlassFish应用服务器的管理平台来，部署web应用。

# 任务5 –玩转应用服务器

[基本要求]

null

## 任务5 –玩转应用服务器

### 5.1

在eclipse中创建动态web项目，并将其以war包格式导入导出当前工作目录，然后在eclipse中配置tomcat、Jboss和GlassFish应用服务器，并部署web应用。

[扩展要求]

null

## 任务5 –玩转应用服务器

### 5.2

不依赖eclipse，启动和关闭tomcat、Jboss和GlassFish应用服务器，查看服务器日志，查看部署的应用，查看JSP转化成的Servlet java源码。

[扩展要求]

null

## 任务5 –玩转应用服务器

### 5.3

通过tomcat、Jboss和GlassFish应用服务器的管理平台来，部署web应用。

[扩展要求]

null

# 任务5 –玩转应用服务器

## [上交]

- 不需上交

# 任务6 –Servlet+JSP+JSTL+EL

## [目标]

- 1) 掌握运用Servlet基本知识开发web应用。
- 2) 掌握运用JSP基本知识开发web应用。
- 3) 掌握运用JSTL基本知识开发web应用。
- 4) 掌握运用EL基本知识开发web应用。



# 任务6 –Servlet+JSP+JSTL+EL

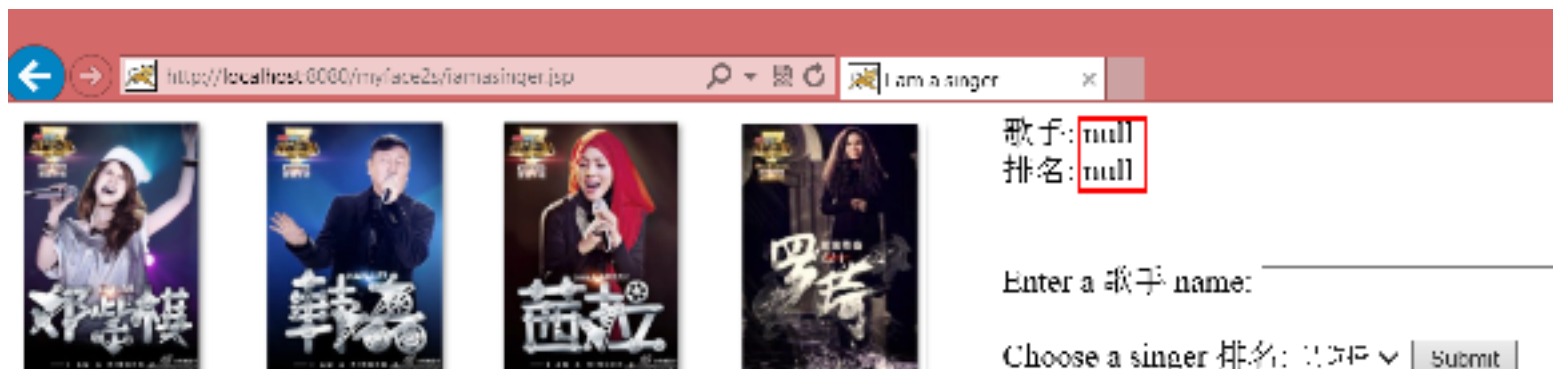
## [基本要求]

- 1) 在Eclipse中开发；
- 2) 用Ctrl+Alt+F自动格式化代码；
- 3) 遵循我们的Java语言Code Style（参考附录1）；
- 4) 尽量提高代码质量。

# 任务6 –Servlet+JSP+JSTL+EL

## 6.1 理解myface2s应用， 修改我是歌手null bug

理解myface2s的实现。并修改从index.jsp进入iamasinger.jsp时的null bug，即不出现null，而显示空白。



[扩展要求]

修改bug后，在eclipse中将myface2s部署到Tomcat、JBoss、Glassfish上，并运行myface2s。

# 任务6 -Servlet+JSP+JSTL+EL

6.2 基于myface3， 创建SimplicityServlet， 使得在myface3的index.jsp中点击Simplest is the key之后， 会先触发SimplicityServlet， 然后再跳转到simplicity.jsp。



[扩展要求]

修改bug后， 在eclipse中将myface3部署到Tomcat、JBoss、Glassfish上， 并运行myface3 。

# 任务6 –Servlet+JSP+JSTL+EL

6.3 基于myface4, 1)修改fish\_img.jsp, 用<c:choose>来代替<c:if>;  
 2)用JSTL改写simplicity.jsp, 即用JSTL替换其中的JSP脚本元素。使得功能不变, 即不指定loop参数值时, 默认loop为1, 可打印出如下图所示; 若在url后指加上?loop=n, 则可以打印出n组Do the simplest...和YAGNI



[扩展要求]  
 修改bug后, 在eclipse中将myface4部署到Tomcat、JBoss、Glassfish上, 并运行myface4 。

# 任务6 –Servlet+JSP+JSTL+EL

## [上交]

- <1>在“任务6”目录（已创建）下，创建自己的子目录，其命名为“序号姓名”（序号为两位数字，序号和姓名间无空格），如“92翁秀木”；<2>将各子任务的解答文件上传到“序号姓名”目录下。
- 6.1需将修改后的代码和效果截图贴在文档“**Java EE基础与应用\_任务\_v6.1\_上交.docx**”
- 6.2需上交SimplicityServlet.java源码文件（包括该Servlet的配置信息），包括修改后的index.html文件
- 6.3需上交改写后的Simplicity.jsp和fish\_img.jsp文件

# 附录1 Our Java Coding Style 2.2

- 包含代码头信息（代码头注释请参见本张ppt的备注）
- 尽量使用自动格式化工具
- 方法名尽量用动词，首字母小写，不用下划线。
- 类名用名词，首字母大写，不用下划线。
- 变量名以字母做开头字符，且首字母小写，不用下划线。
- 命名要用骆驼命名法。
- 常量字母都大写，单词间用下划线分隔。
- 包名（package）都是小写，且范围由大到小，例如net.gupt.cs.jee
- 名称要有意义，不宜过长。
- 足够且有意义的注释