## 开发环境

### 软件环境

|  |  |  |
| --- | --- | --- |
| 项目 | 名称 | 资源地址 |
| 1．版本管理 | | |
| Server | Visual SVN Server 2.5.1 | http://www.visualsvn.com/server/ |
| 2．云端开发 | | |
| OS | Windows 7 Ultimate 32bit |  |
| IDE | Eclipse IDE for Java Developers v3.5（Galileo）windows 32bit | http://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/galileo/SR2/eclipse-java-galileo-SR2-win32.zip |
| Plug-in |  |  |
|  | Subclipse1.8.x | http://subclipse.tigris.org/update\_1.8.x |
|  | Android Development Tools | https://dl-ssl.google.com/android/eclipse/ |
|  | EclipseUML3.5.2 |  |
| 3．云服务开发 | | |

### 硬件环境

### 1.3 Eclipse开发环境配置

1．注释模板

（1）文件注释

Window —〉 Preferences —〉 Java —〉 Code Style —〉 Comments —〉Files

/\*\*========================================

\* File: ${file\_name}

\* Package: ${package\_name}

\* Create: by ${user}

\* Date: ${date}:${time}

\*\*======================================\*/

（2）类注释

Window —〉 Preferences —〉 Java —〉 Code Style —〉 Comments —〉Types

/\*\*

\* <p>${type\_name}</p>

\*

\* <p>功能描述</p>

\*

\* **@author** ${user}

\*

\* ${tags}

\*/

（3）构造函数注释

Window —〉 Preferences —〉 Java —〉 Code Style —〉 Comments —〉Constructors

/\*\*

\* <p>${enclosing\_type}</p>

\*

\* <p>构造函数</p>

\* ${tags}

\*/

（4）方法注释

Window —〉 Preferences —〉 Java —〉 Code Style —〉 Comments —〉Mothods

/\*\*

\* <p>${enclosing\_method}</p>

\*

\* <p>功能描述</p>

\*

\* ${tags}

\*/

2．信息提示

Window —〉 Preferences —〉 Java —〉 Editor —〉 Content Assist

Auto activation triggers for java：.abcdefghijklmnopqrstuvwxyz

3．文本编码

Window —〉 Preferences —〉Workspace —〉Text file encoding—〉other:UTF-8

### 1.4 编码规范

#### 1.4.1命名规范

##### 1.4.1.1 基本原则

* 使用完整的英文描述符为变量/字段/类/接口/包等命名，不得使用汉字或汉语拼音；
* 使用业务领域的专业（英文）或近义术语，如楼面，无法翻译成英文，则可用waiterorder（服务员点单）近义替代；
* 尽量不用缩写，只有当名字过长时使用，一般取单词的前三个字母或采用1.4.1.2节中常用的缩写形式，不超过20个字符；
* 采用驼峰命名原则，方法、变量首字母小写（如getTableName()），类和接口首字母大写(TableManager)
* 尽量避免使用数字，但可采用一些日常约定用法，如2代表to（int2Byte）；

##### 1.4.1.2 文件、包命名

* 文件与类同名，采用驼峰命名原则，首字母大写；
* 基本包：com.hithing.hsc
* 包命名全部使用小写字母，命名原则：[基本包].[项目名].[模块名]，包的层次控制在5层以内（包含基本包），如com.hithing.hsc.bll.manage

##### 1.4.1.3 类、接口命名

* 所有单词首字母大写，并能确切表示该类或接口的含义。接口使用I前缀和able后缀，如IAsynSendable表示具有可异步发送（消息）行为的接口

##### 1.4.1.4 字段

* 常量采用英文大写单词，并使用下划线进行分词，如INVALID\_TABLE\_ID；
* 变量命名使用有意义的英文描述符，采用驼峰原则，首字母小写，如tableId
* 对于控件类型的变量，采用以下原则：英文描述符+缩写。如tableNameEdt；
* 用常量替代 “神秘的数”，如if(j < 5)….用if(j < MAX\_CONNECT\_TIMES)…，提高程序可读性
* 使用复数表示多值，如orderItems

##### 1.4.1.5 方法

* 采用驼峰原则，首字母小写，第一个单词选用具有实际意义的动词
* 取值方法用get前缀，设值方法用set前缀，判断方法用is（has）前缀，如getId()，setName()，isUsing()

##### 1.4.1.6 缩写

|  |  |  |
| --- | --- | --- |
| 序号 | 全称 | 缩写 |
| 1 | Button | btn |
| 2 | EditText | edt |
| 3 | Dialog及其子类 | dlg |
| 4 | TextView | txv |
| 5 | ListView | lsv |
| 6 | GridView | gdv |
|  |  |  |

#### 1.4.2声明

按照字段、方法、内部接口、内部类的顺序声明。

##### 1.4.2.1字段

声明顺序

* 常量
* 类变量
* 实例变量
  + 公有字段
  + 受保护字段
  + 私有字段

##### 1.4.2.2方法

原则：尽量限制成员函数的可见性

声明顺序

* 构造方法
* 静态公有方法
* 静态私有方法
* 公有方法
* 友元方法
* 受保护方法
* 私有方法
* 重载父类的方法
* Main方法

#### 1.4.3 类与接口设计

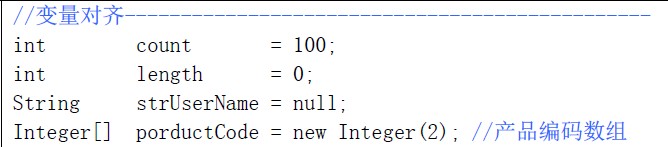
##### 1.4.3.1 基本原则

* 类的划分遵循单一职责原则，即一个类只能做一件事情，划分粒度要适中
* 多用接口或使用抽象类，提高代码重用度
* 多用设计模式，根据需求和业务分析程度，随时进行代码重构
* 将不需要继承的类声明为final

#### 1.4.4 缩进和对齐

* 子功能块在其父功能块后缩进，缩进量为一个TAB(4个字符)
* 关系密切的行应对齐，包括类型、修饰、名称、参数等
* 连续赋值时对齐操作符

示例



#### 1.4.5 android部分

##### 1.4.5.1 基本原则

* 代码中不出现中文，程序中用到的中文字符串统一在strings.xml中定义
* 使用style.xml实现风格重用
* 使用include和viewstub实现布局文件嵌套及重用

##### 1.4.5.2 命名规范

* 活动的layout文件命名：act\_活动名\_英文描述符，如act\_waiterorder\_main，表示楼面活动主布局，英文描述符单词之间用下滑线分词
* 通用的layout文件命名：cmn\_英文描述符，如cmn\_radio\_item，表示radio类型的条目
* layout中的id命名：活动名\_view缩写\_英文描述符，如waiterorder\_edt\_search，表示楼面活动界面中的搜索编辑框
* activity中的View变量命名：英文描述符+View缩写，如searchEdt
* Strings.xml中的id命名：活动缩写（不超过6个字符，超过6个字符则缩写）\_View缩写\_英文描述符，如worder\_dlg\_print\_title.表示楼面活动中使用的打印对话框标题
* 活动、服务命名使用Activity+英文描述符，Service+英文描述符，如ActivityWaiterOrder和ServicePrintInvoice.

## 运行环境

（略）

## 程序库及包结构

本软件命名为Greeder（贪吃者），共包括三个程序库和一个应用程序

### 3.1 DataAccessLib库

数据访问库，完成数据持久层所需要的各种逻辑。目前包括database包和entity包

#### 3.1.1 com.hithing.hsc.dal.database包

用于存放不同数据库的映射类，目前已经实现了android的sqlite数据库的映射。

1. GreederSqliteConfigureUtil.java
2. GreederSqliteHelper.java

#### 3.1.2 com.hithing.hsc.dal.entity包

用于存放定义的数据层实体，已使用以下实体

1. BaseDalEnt.java：基数据实体，所有数据实体均继承自该实体
2. HallDalEnt.java：分厅数据实体
3. HallSortEnt.java：分厅大类数据实体

### 3.2 BizLogicLib库

#### 3.2.1 com.hithing.hsc.bll.android包

用于存放业务层为android系统特别提供的控制类，已完成下列功能

（1）GreederSettingsHelper.java 用于辅助偏好设置功能开发的类

#### 3.2.2 com.hithing.hsc.bll.control包

业务控制类包，用于存放与餐饮业务有关的控制类

#### 3.2.3 com.hithing.hsc.bll.entity包

业务实体包，（待定）目前尚未使用业务实体，最终是否加入此包有待商榷。

#### 3.2.4 com.hithing.hsc.bll.manage包

与功能（但与餐饮业务关系不大）有关的控制类，多用于对某类对象的管理操作，目前包括以下类

（1）DinnerTablesManager.java：餐台管理类

### 3.3 ComponentLib库

用于开发界面时，自定义的UI组件库

#### 3.3.1 com.hithing.hsc.widget包

已完成以下小部件

（1）MultiColorStyleAdapter.java：为AdapterView提供条目级不同色彩显示风格的适配器

（2）MultiStyleAdapter.java：为AdapterView提供条目级不同显示风格的适配器

### 3.4 GreederApp应用程序

#### 3.4.1 com.hithing.hsc.app包

目前包括

（1）GreederMain.java：主活动

（2）GreederSetting.java：设置活动

（3）GreederWaiter.java：楼面活动