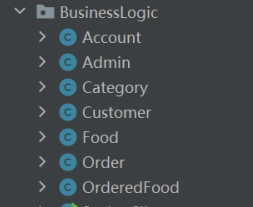
**SCUT点餐系统——PreOrder程序设计说明书**

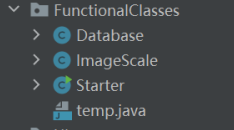
概述

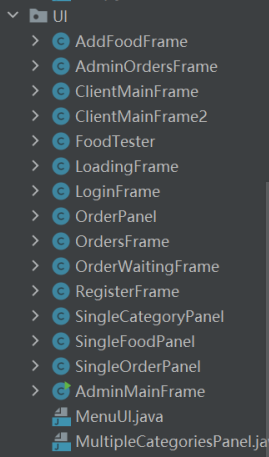
选择以食堂为背景的点餐系统为题的主要原因是切合生活实际、符合生活需要，并且这类生活场景实用性的程序开发设计更能引起我们对程序设计的改进思考和问题解决。根据题干要求系统至少包括以下信息：菜式信息和客户点单信息。菜式信息简单包括：菜的编号、菜的名字、菜的类型、菜价、菜的介绍等，而客户点单信息简单包括：本单编号，餐桌编号，用餐人数，日期，菜名列表（菜名、数量、价格等）、账单总额等，为了方便维护，我选择了使用数据库MySql来管理。系统基本功能包括（1）提供菜式类别，每个类别下可以展示菜品（2）选择餐桌编号、用餐人数、菜式等，由用户确认后可以下单，这时计算总金额（根据人数追加茶位费），或者用户可以取消下单。（3）用户加单，在点菜单中追加菜式，更新总金额。（4）选择结账，完成结账后，清空本餐桌点菜单。（5）简单实现了后台数据维护管理系统界面，简单提供了增加或减少餐厅食物以及增加或修改系统图片等基础功能。简单分析可知我们可以将整个系统分为三个block实现（具体划分情况将在以下各板块给出）开发环境JDK-19.0.1，Java-IDE IDEAIntelliJ， MySQL5.0。

程序设计概要

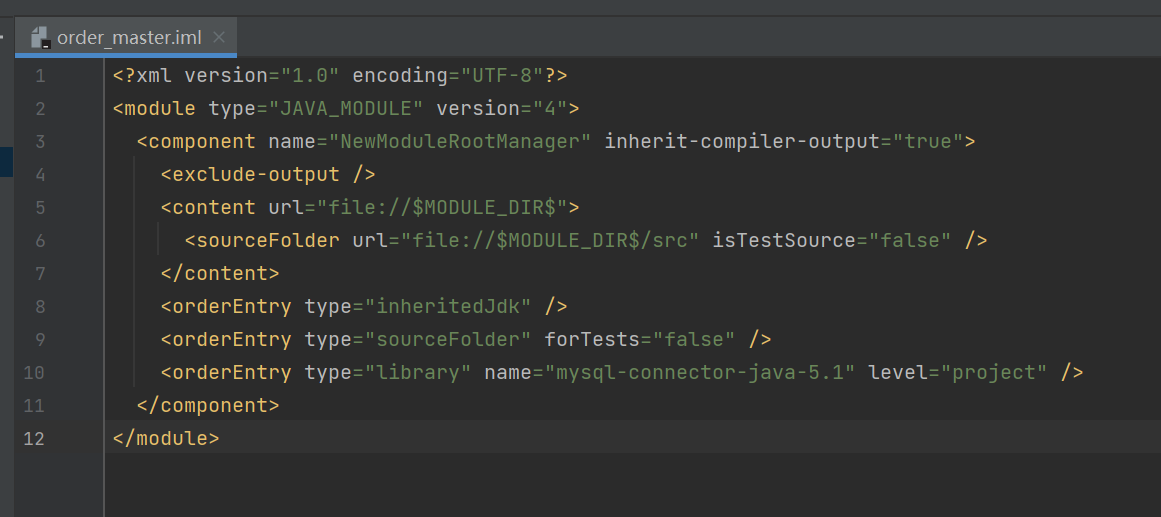
 该程序可以大致分为3个block——底层逻辑block，功能函数block和UI功能block。

底层逻辑中（BussinessLogic）将根据点餐系统所需要的基本对象设计基本类以体现java面向对象编程的基本思想。Account类别中定义了我们用户登录系统的基本类信息，Admin类则是管理员相关类，Category类中给出了食堂基本食物分类的信息，通过java的继承（extension）性质我设计了Customer类，他与Account类的区别在于此类给出了更多实际性的信息以及对应的功能函数——用户的名字、电话号码等具体信息和发送订单的行文函数。类似地，food类也通过继承category类实现了某一餐点在点餐系统中所具有的特定信息，如食物编号、价格等；订单类（Order）中将记录当前发起订单的用户信息和对应的点单信息；已点食物（ordered food）类则封装好了每一单被选择食物的具体信息，如顾客的评价、是否追加等。

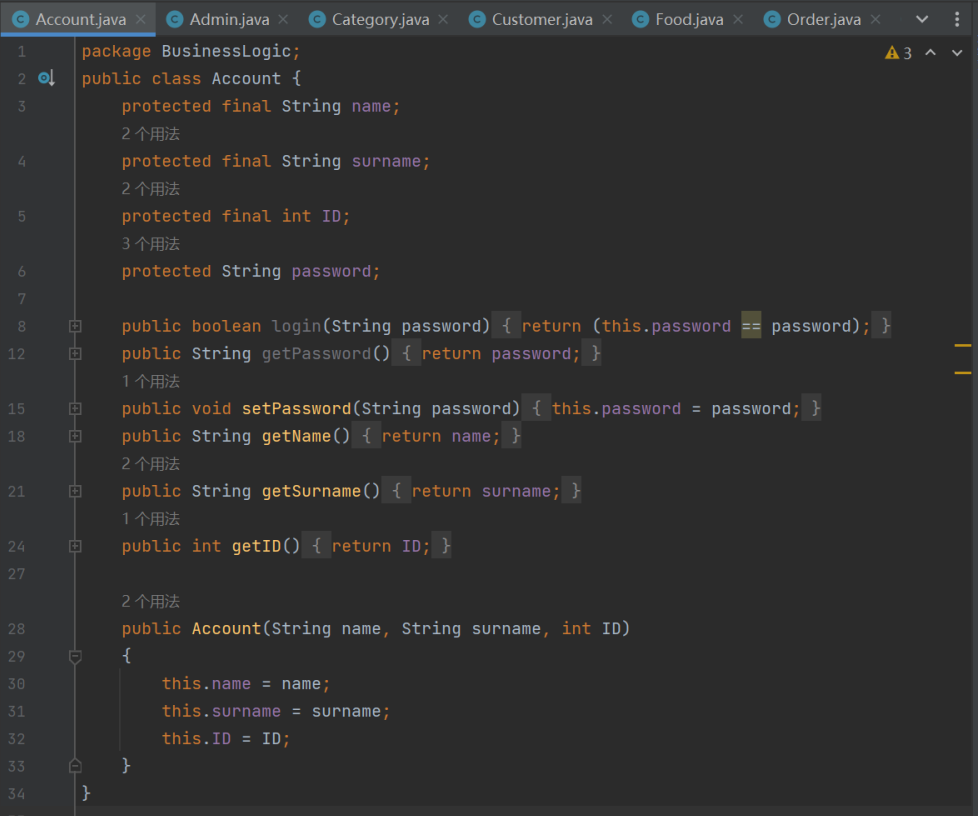
功能函数（Functional Class）中我们具体编写了3个java文件，如右图所示，Database.java 文件主要封装了java程序连接数据库MySql的一些函数；ImageScale.java 文件中包含了对数据库中读取的二进制文件获取UI边界矩阵（用于UI画面布局使用）；Starter.java顾名思义便是用户的启动文件，通过运行这个文件，调用所有核心功能，进入我们的食堂点餐系统。

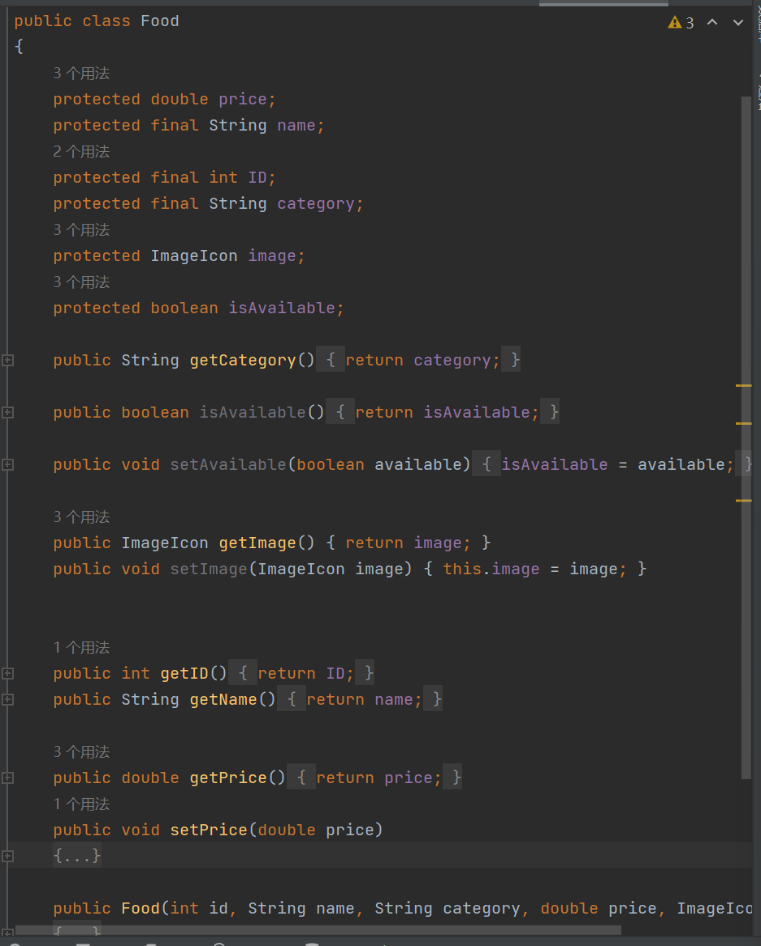


最后一个block则是所有的UI编写文件。由于我们所掌握的知识有限，因此全部采用JOptionPane来实现，由此也产生了许多繁冗的代码，封装成一个个功能文件可以使程序的可维护性得到大大提升。

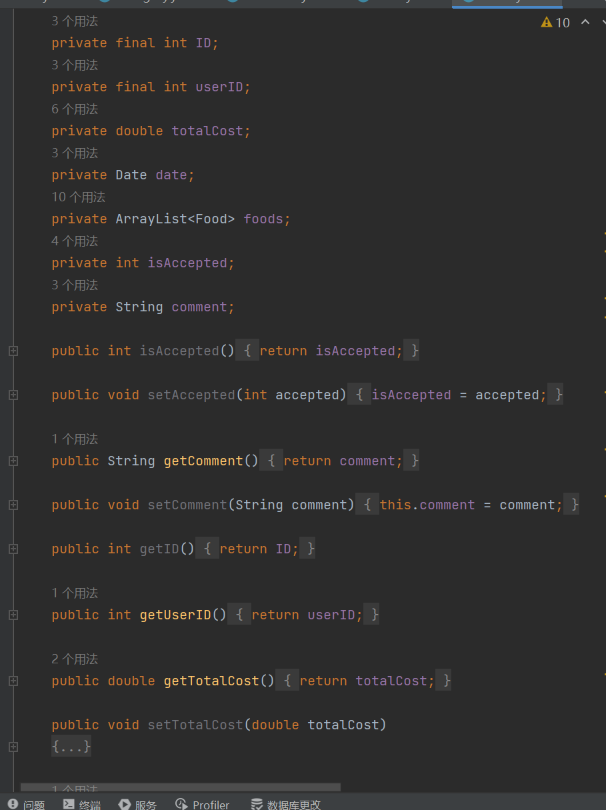
程序设计细节

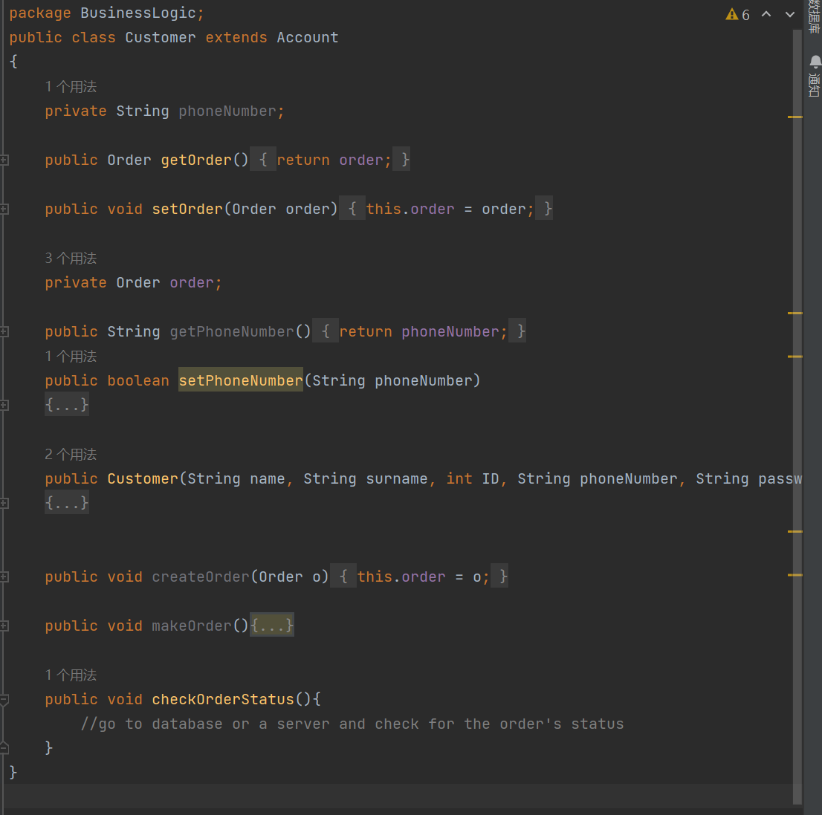
要能使用mysql-connector-java-5.1的包，通过查阅资料，修改project预编译文件iml信息增加lib：mysql-connector-java 即可。

 以上给出了class Account的具体定义。Account中包含了账户对象的基本属性——账户的姓名，用户在系统中的代号ID，用户对应用于登录的密钥password。这四项基本属性为了维护java编程的封装性使用protected关键字保护，并且给出了各个数据成员对应的get和set函数。其中构造函数使用了简单的参数列表初始化对应类成员变量。

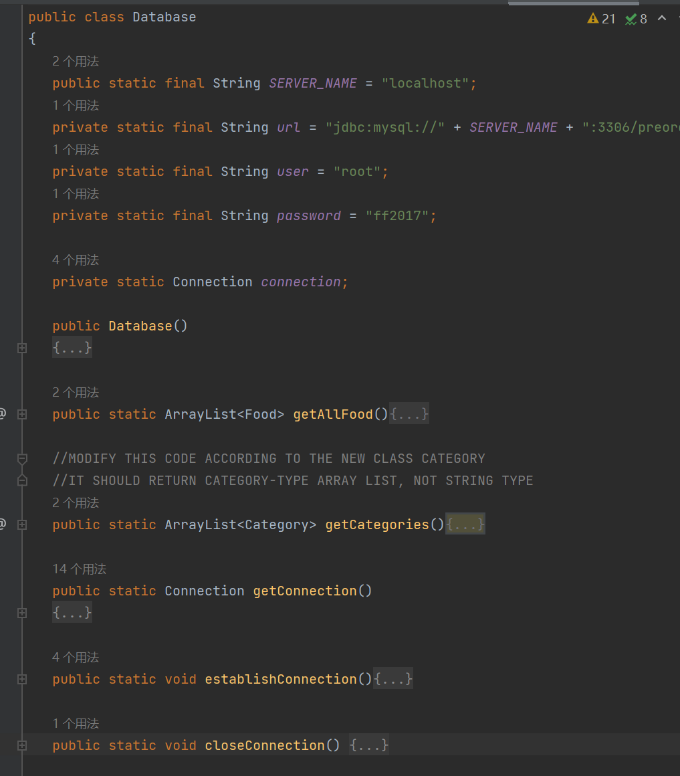
以上给出了食物种类Category的基本定义，由于category需要涵盖的信息有某一食物类名称name；在数据库中便于查找和维护的序列号id和对应用于UI显示用的image，因此定义这三个数据成员为该类私有数据成员，并给出对应的set和get函数。需要额外注意的是，一般Java在文件读取与处理中，一般都应增加空指针判断，避免程序尝试读取没有意义的指针导致抛出异常。

上图给出了基类Food的基本定义。食物包含基本数据成员：该食物的价格price：double类型；食物名称name: String类型；该食物在数据库中的便于维护的编号id：int类型；食物对应的种类category： String类型；食物对应的图标：image：ImageIcon类型以及当前是否上架isAvailable： boolean类型。这些基本数据成员都应该以protected方式保存以便于后续多态性的利用。类中还给出了所有私有数据成员对应的set和get函数。对于价格price的set函数要注意检验数据合理性。

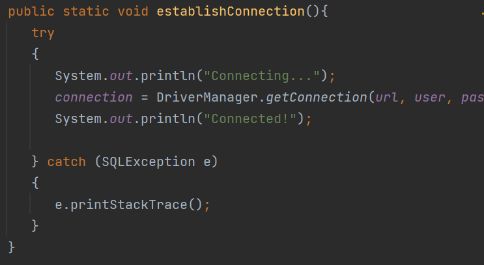
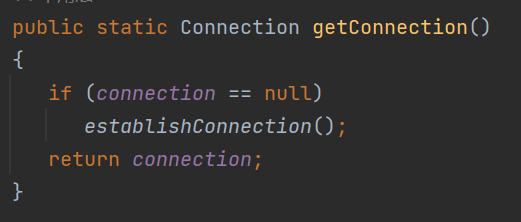
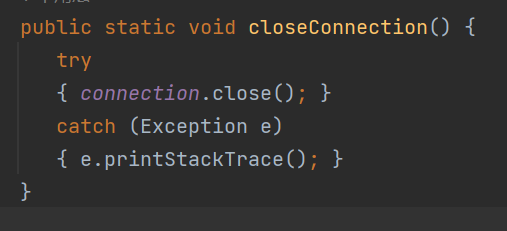
 左图给出了订单类的部分定义（class Order）。设计订单类是为了将订单所涉及的用户、食物及其信息、用餐金额等数据封装起来，因此设计一下私有数据成员：订单号ID：int； 订单发起用户编号userid：int；订单当前累计总额totalCost：double；订单发起日期date：Date；订单所包含的食物foods：ArrayList<Food>；顾客对当前订单的评价comment:String；以及当前订单是否被接受的状态表示变量：isAccepted: int；并给出了所有私有数据成员的对应set和get 函数。



上图给出了顾客类的定义。顾客类作为账户类所派生出的具体类，我们使用extend关键词继承Account类。为了确保用户以较为安全的方式登陆系统——使用手机号码的方式，因此我们在account类的基础上增加了其派生类用户类所私用的数据成员phonenumber，顾客所发起的订单order：Order(自定义类)。

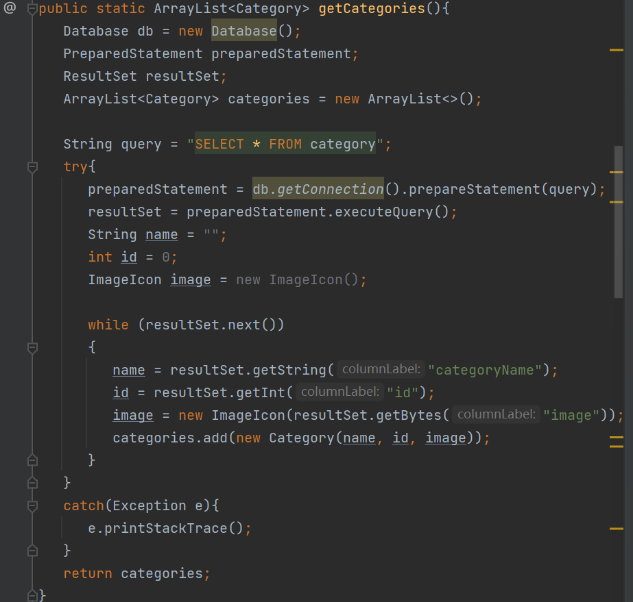
 底层代码逻辑建立基本完成后，为了保持和体现Java编程封装性的优势，我将编写功能函数文件。为了方便维护和管理用户信息，我使用Oracle的MySql5.0保存和管理数据，由于MySql具有单独一套语言体系，为了便于使用java链接sql数据库和修改数据，我们必须编写相应函数定义方言函数以实现Mysql的增删查改基本功能：

我们定义数据库类型Database用于保存从MySql中查询到的数据表。首先数据库的类中必须包含登录MySql的服务器地址，服务器名，用户名及登录密码，这些变量都是无需再更改的，因此如上图所示我们直接定义对应变量为private static final类型。

 链接数据库涉及到3个基本步骤——建立链接，获取链接以及关闭链接：

在数据库的链接建立完毕后，我们应该即使从数据库中读取我们所需要的相关信息，例如所有类别，所有餐点的价格和图片。

 当我们需要从数据库中读取所有食品相关信息时就采用上图所示的方法：通过定义好的MySql方言“SELECT \* FROM foods”语句可以对数据库中的某一表（此处查找范围是表头为foods的table）进行遍历搜索。利用Mysql查找的数据一般使用resultset类型的变量存储结果。整个搜索逻辑可以概述为：只要当前数据库中foods类型的表没有被遍历完全，该函数就会将对应项信息依次存入临时变量，并通过创建food对象使用具有具体值的临时变量构造新对象存入运行类foods的ArrayList变量中即可。

而获取所有食物种类与上述method原理类似，只有涉及的变量不太相同，给出图后就不再赘述。