

Lab2 - 星爸爸咖啡在线销售系统 v1.0

2019年 软件工程课程系列实验

1. 实验概述

1.1. 实验目标

通过本次Lab，完成一个在线咖啡销售系统的几项基本功能，体验软件开发过程，体验代码质量和软件设计的重要性；基于Git和DevCloud云平台进行团队协作开发。

1.2. 实验要求

- 以小组为单位；
- 根据实验内容完成代码；
- 基于Git进行团队协作；
- 撰写实验心得报告。

2. 实验内容

2.1. 项目背景

为紧跟时代潮流，方便店内咖啡选购和咖啡销售，星爸爸咖啡连锁公司希望开发一个**在线咖啡销售系统**。星爸爸咖啡连锁公司给出了对这个系统的一些**基本需求**，架构师已编写了基本类的**UML设计**以及**开发要求**，希望旗下优秀的开发团队能完成这一任务。

该项目需要支持如下功能：

- 注册：售货员可以使用唯一的账号在系统中进行**注册**；
- 登录：售货员可以使用已注册的账号密码在系统中进行**登录**；
- 获取登录状态：系统支持当前**状态的检查**；
- 计算商品价格：售货员可以使用系统进行不同杯型咖啡的**价格计算**。

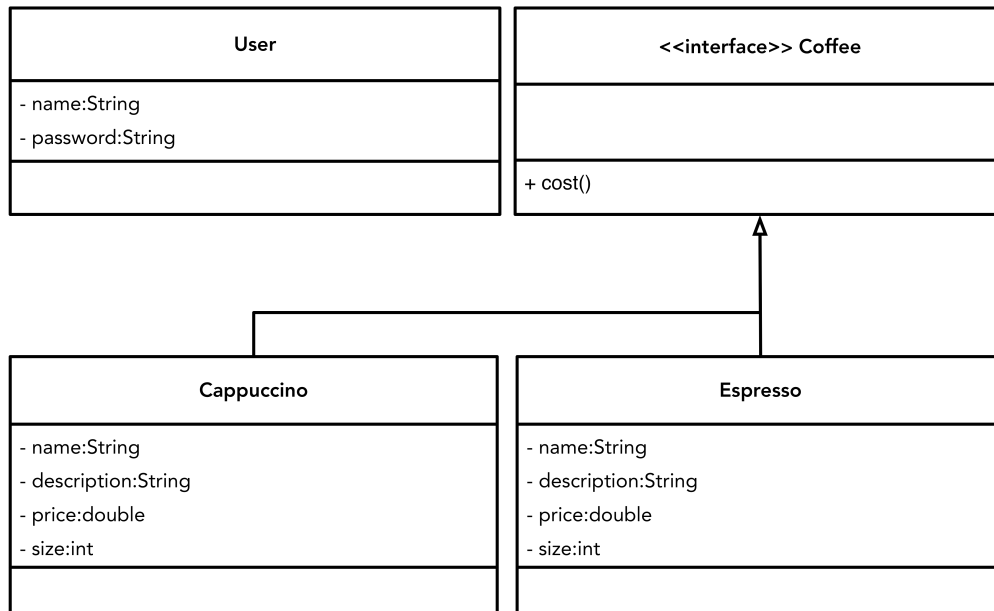
2.2. 项目涉及人员

- 售货员：具有系统功能的基本操作权限；

2.3. 项目要求

2.3.1. 来自架构师的UML设计

架构师给出关键类的设计如下：



这里对各个类进行说明：

- **User** 表示用户
 - `name` 表示用户的账户，即唯一识别用户的标识码；
 - `password` 表示用户设定的密码；
- **Coffee** 表示咖啡
 - `name` 表示唯一识别咖啡的标识码；
 - `description` 表示咖啡描述；
 - `price` 表示咖啡价格；
 - `size` 表示咖啡杯型；
 - `cost()` 为计算咖啡价格的方法。
- **Espresso**和**Cappuccino**表示某种具体的咖啡饮料。

2.3.2. 数据说明

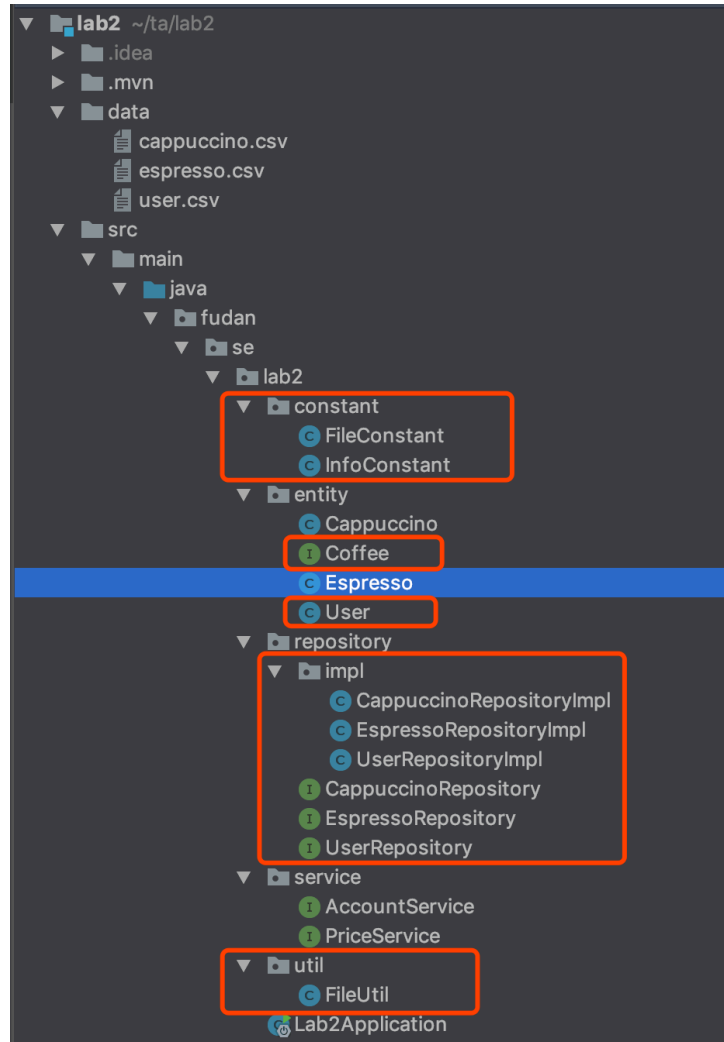
项目的data文件夹中有两种数据：

- `coffee.csv`：表示咖啡的基本信息，存储方式为(`name` , `description` , `price` , `size`)
 - `espresso.csv` 表示Espresso的基本信息
 - `cappuccino.csv` 表示Cappuccino的基本信息
- `user.csv`：表示账户的基本信息，存储方式为(`name` , `password`)

2.3.3. 项目说明

架构师为开发团队搭建了基础框架（请到Classroom中根据模板生成项目获取本次Lab的基础代码，并clone至本地进行开发，开发中建议使用IDEA或是Eclipse等集成开发环境导入项目，注意选用导入Maven项目）：

这里对代码进行一些说明：



- 工具类（上图红框中为框架提供的工具类和固定代码，请勿删改已有代码和接口）
 - 数据存取
 - `UserRepositoryImpl` 实现了 `UserRepository` 接口，提供数据存储操作：
 - `createUser` : 创建单个User对象
 - `getUser` : 根据 `name` 获取单个User对象
 - `CappuccinoRepositoryImpl` 实现了 `CappuccinoRepository` 接口，提供数据存储操作：
 - `createCappuccino` : 创建单个Cappuccino对象
 - `getCappuccino` : 根据 `name` 获取单个Cappuccino对象

- `EspressoRepositoryImpl` 实现了 `EspressoRepository` 接口，提供数据存储操作：
 - `createEspresso` : 创建单个Espresso对象
 - `getEspresso`: 根据 `name` 获取单个Espresso对象
- 数据删除修改
 - 目前请直接修改对应的 `csv` 文件进行数据删除
- `FileUtil` , 提供文件读写操作
- (以上接口在执行失败后将抛出异常, 请注意处理)

- 常量

- `FileConstant` , 提供文件相关常量数据
- `InfoConstant` , 提供消息相关常量数据

- 开发说明

- `PriceService`

- 对 `Coffee` (`Espresso` `Cappuccino`) 中的 `cost` 方法进行相应实现, 使 `PriceService` 支持计算商品价格的功能;
- 实现类统一命名为 `PriceServiceImpl` , 并放在`service/impl`路径下;

- `AccountService`

- 实现 `signup` `login` 方法, 使 `AccountService` 支持注册、登录功能;
- 实现 `checkStatus` 方法, 使 `AccountService` 支持获取用户登录状态的功能;
- 实现类统一命名为 `AccountServiceImpl` , 并放在`service/impl`路径下;

- `Main`

- 这个类作为客户端, 实现系统的业务逻辑:

```

@SpringBootApplication
public class Lab2Application {

    public static void main(String[] args) {
        SpringApplication.run(Lab2Application.class, args);

        // here to write your main business logic.
        // step1: signup

        // step2: login

        // step3: check status

        // step4: price service
    }
}

```

- 开发需求

- 注册

- 在未获得用户账号之前，售货员需要进入系统，使用唯一的账户名进行注册，注册成功后，系统将持久化记录这个账号（可以在 `user.csv` 中查看）。
- 若账户名重复，系统将抛出运行时异常（`RuntimeException`），例如若账户名已存在，提示以下信息：`User already exist, name : xxx`（这部分请参考工具类 `FileUtil` 中的抛异常实现）。

- 登录

- 售货员进入系统进行其他操作前，需要进行登录操作。售货员使用已注册的账户名和密码进行登录，如果密码与系统记录的账号密码匹配，则登录成功。登录成功后系统将记录登录状态；
- 若售货员登录失败，系统将抛出运行时异常（`RuntimeException`）提示登录失败，例如若密码错误，提示以下信息：`Username or password error.`；
- 若售货员登录失败则无法进行接下来的其他操作（获取商品描述、计算商品价格）。

- 查看状态

- 系统可以获取当前售货员的登陆状态，成功为true，失败为false。

- 计算商品价格

- 售货员可以选定咖啡种类、咖啡杯型，系统将为售货员计算该杯型咖啡的最终价格。咖啡的最终价格计算如下：`最终价格=咖啡价格+杯型价格`（单位：\$）
- 售货员可以在选定某种咖啡种类杯型的前提下，选定咖啡杯数，系统将为售货员计算多杯咖啡的最终价格，价格计算如下：`多杯最终价格=单杯最终价格*杯数`（单位：\$）
- 咖啡杯型目前分三种，大杯（3）、中杯（2）、小杯（1）。不同杯型的价格不同：

大杯	中杯	小杯
咖啡价格 + 6\$	咖啡价格 + 4\$	咖啡价格 + 2\$

2.3.4. 开发要求

- 各对象（`User`，`Coffee`，`Espresso`，`Cappuccino` 等）在创建时需检查字段的合法性（非空）。
- 各模块日志输出规范：
 - 格式

使用slf4j输出，例如：

```
logger.info(
    MessageFormat.format("User login successfully, name: {0}", "Tom")
);
```

结果：

```
User login successfully, name: Tom
```

- 各模块需在控制台输出以下 信息
 - **AccountService.login:** `User login successfully, name: Tom`
 - **AccountService.signup:** `User signup successfully, name: Tom`
 - **AccountService.checkStatus:**
 - 如果登录成功: `User has logged in`
 - 如果登录失败: `please login`
 - **PriceService.cost:** `name: Espresso, size: 1, number: 2, price: 20$`

以上输出中的 `Tom`，`Espresso`，`1`，`2`，`20` 表示具体用户名、咖啡、杯型、杯数、最终价格，实际测试时按实际配置和输入而定。

- 架构师在给出的框架中给出了明确的接口定义，请严格根据接口定义实现接口功能；
- 除了工具类中已有代码和给定的接口定义外，你们可以增添修改其他任何代码，例如增加属性，增加辅助类，增加接口内部实现，whatever。

3. 注意事项

- 请注意代码质量和规范；

- 请考虑各项功能的可扩展性和易维护性；
- 请勿修改定义好的接口；
- 请勿改动工具类；
- 请勿抄袭！！
- 请按时提交项目及报告。

4. 作业提交

4.1. 提交代码

- 按框架导入项目Lab2；
- 按要求实现代码；
- 进入Classroom提交作业-Lab2。

4.2. 提交实验报告

- 阐述各个功能模块的实现思想（包括遵循的编码规范以及如何体现可扩展性和易维护性）；
 - 阐述实现中遇到的问题与解决方案（如果有的话）；
 - 附上小组分工；
 - 进入Classroom提交作业-Lab2实验报告。
-
- 发布时间：2019年3月13日
 - 完成时间：一周
 - 截止时间: 2019年3月20日晚 24：00