**敏捷开发小学期实践报告**

组员：

张志成：14080205

阚飒：14080208

姚汝达：14080216

刘子琪：14080221

李爽：14080224

2016年6月30日

#### 测试

1.1输入测试

1. 存入正常物品
2. 存入没有促销信息物品
3. 存入促销和折扣同时存在的物品
4. 存入促销和VIP折扣同时存在的物品
5. 存入促销，VIP折扣和折扣同时存在的商品

1.2 Pos机测试

1. 正常的用户购买测试
2. VIP用户购买测试
3. 总价测试
4. 节省价测试

1.3促销测试

1. 正常用户购买打折测试
2. VIP购买打折测试
3. 正常用户购买促销测试
4. VIP购买促销测试
5. VIP购买打折、促销和VIP折扣测试

1.4 VIP测试

1. VIP积分测试

#### 二、关系说明

Item 的集合是 itemgroup

Itemgroup和user的集合是 shoppingchart

User的集合是userlist

Report和shoppingchart 的组合是 pos

#### 三、实现

**3.1Item类（物品类）**

属性：String编号，String名字，String单位，double价格，double打折，boolean促销，double会员折扣

行为：相关get（）和set（），构造函数

**3.2ItemGroup类（同种物品的集合类）**

属性：User用户，Item物品，int 数量

行为：相关get（）和set（），构造函数

add () 添加一个物品 {数量+1}  
subTotal() 得到集合的总价{

if（有促销，且数量大于2）

返回数量×价格

Else if（用户为VIP）

返回数量×价格×会员折扣×折扣

Else

返回数量×价格×折扣

}

saving() 得到节省的价钱{

if（有促销，且数量大于2）

返回（数量÷2）×价格

Else if（用户为VIP）

返回数量×价格×（1-会员折扣）×（1-折扣）

Else

返回数量×价格×（1-折扣）

}

**3.3Pos类（POS机类，输出信息）**

属性：无

行为：getShopList（）{

输出购物清单

If（user为VIP）

输出会员信息

For 遍历购物单信息

输出购物单的信息

输出总价

If（有折扣，促销商品）

输出节省价格

}

**3.4Report类（结果类，计算购物单信息）**

属性：list 物品集合，int 会员等级，int会员积分

行为：相关get（）和set（），构造函数

subTotal() 计算集合的总价{

for 遍历商品list中的商品组

返回购物商品的总价

}

saving() 计算节省的价钱{

for 遍历商品list中的商品组

返回购物商品的节省价

}

**3.5ShoppingChart类（购物单类）**

属性：List 商品组， User 用户

行为：相关get（）和set（）

**3.6User类（用户类）**

属性：String编号，String姓名，boolean是否是会员，int积分

行为：相关get（）和set（），构造函数

addScore（）{

if（用户是VIP）

积分叠加

}

**3.7UserList类（用户表单类）**

属性：User 用户， Item 物品

行为：相关get（）和set（），构造函数

#### 四、步骤

1.第一次迭代

添加了 商品模块

商品类item

商品组类 itemgroup

添加了 pos机模块

Pos机输出类 pos

Pos机数据处理类 report

Pos机购物单类 shoppingchart

添加了 物品折扣信息 discount

更改了 pos机折扣信息输出

2.第二次迭代

添加了 文件处理模块

文件读取类 inputpaser

数据处理返回类 datagenerate

添加了 物品促销信息 promotion

更改了 pos机促销信息输出

更改了 文件读取促销信息

3.第三次迭代

添加了 用户模块

用户类 user

用户组类 userlist

添加了 物品VIP折扣信息

更改了 pos机VIP折扣信息输出

更改了 文件读取VIP折扣信息

添加了 用户积分信息 score

添加了 用户积分等级信息 level

更改了 pos机用户信息输出

#### 五、小结

1. 关于版本控制github：

我这次第一次接触了版本控制github，首先使用的时候非常混乱，后来渐渐掌握了版本控制的技巧。

我认为使用版本控制的前提是将需求做到比较清晰，对于每个人的任务都有明确的定义，这样大家完成的代码才能够更好的聚合。

其次是对于分支的感想，对于每个人的分支，在这里改变文件并不会影响总的版本，这样还可以进一步确认个人完成的代码是否符合要求。

最后是我们认为在版本控制的时候只需要提交重要的代码部分，而不是将整个project提交，这样不仅可以做到更加清晰明确，还能够减少整合时的难度。

1. 关于任务计划Trello：

本次的开发并没有很充分的用到trello的功能，这是因为我们在开始没有十分明确的分工，这样在计划任务的时候没有很清晰，所以在定制计划时就会导致有些任务的时间会超时，有些需求也是做完之后才添加的。

1. 关于敏捷开发TDD过程：

这次的TDD开发过程，着重点在于测试代码写在先，然后按照测试代码写功能代码开发。这就要求测试代码尽可能的包含到全面的测试，但在黑盒测试的前提下又对于代码没有足够的了解，致使我在写测试代码的时候有点不知道怎么测试程序运行过程中的具体内容是什么，最后只能测试输出的准确性。

之后再对代码有了一定了解的时候，对于运行过程中的测试也有了很好的测试。

#### 六、建议

建议讲一两节课的内容再放了去自己做

自学也是没问题的，所以这里建议的讲，并不是说把TDD与这次的开发讲的一清二楚；而是（1）把TDD的优势、思想、操作方法大体讲一下；（2）举例子讲一下github与trello最好怎样使用（不是操作方法）。

我们认为这两点如果不清楚，很容易走弯路，而从老师第一节课布置任务的时候，给大家的感觉是默认了同学们都已熟悉TDD与那两个软件，事实上是即便课上听了，也不是那么容易记得住的，毕竟没有实践过。于是大家下去后都查阅资料，有IBM写的有百科上的有知道上的，有的矛盾，有的看不出所以然，测试用例写到个什么程度，关于代码的设计跟测试有多大关系，都是疑问。在实践的时候经过自己体会大多都有了自己的想法，不过还是在前两次检查的时候每次都跟老师的需要有很大的不同，每次检查时候问出来的东西，颠覆了原来的理解，成了难得的一对一教学。所以我最后想，可能在这样软件开发过程（第一次实践）的理论知识上，如果老师多说一些，会对前几次迭代更友好，不过，不讲也很好，充满疑惑与不解的状态是求知欲与积极性很强的。

第二点也是一样的，软件的使用大家悟性都可以，但是怎样用在帮助开发上就不是靠悟性的了我觉得。就trello来说看板里面每个栏目里的小看板，放类似“增加积分功能”的小功能是所有人的一致想法，因为这么想对小的项目，刚一开始很难想象小看板竟然是“第一次迭代”这样级别的，是在第一次检查之后，助教给我们看的时候才知道的，里面每个人负责自己的，还要用任务列表分出很多小任务，可是助教的例子是大项目，有很多可以分工的，在我们这里把一个功能的类和其他引用的类区分开来很不现实，在第三次迭代之前我们都想不明白怎样分工，完全是大家挤在一两个人电脑旁边做出来的。然后第二次检查后才知道每个人自己的branch不是每个开发都要用，每个测试与开发两个人就可以。所以对于我们来说，遇到了很多无关于学习这些思想与技术的弯路，浪费了不少精力与时间，所以还是建议说一些必要的使用方法，比如branch建立的依据，小项目的分工方法，trello小看板的规模（或者给出例子就可以）。这样先进行全班的教学，否则检查的时候在旁边听每个组都犯相同的错，问相似的关乎最基本思想的问题，感觉不是学习得很高效。