# 银行管理系统1.0

架构的搭建，实现分层开发，完成存款、取款、查询等基本功能。

包名：com.cx.bank.test (测试层)

TestBank 测试类（该类通过：1代表查询、2代表取款、3代表存款功能来调用业务层的各个方法）

包名：com.cx.bank .manager （业务层）

ManagerImpl业务类：

inquiry()---查询方法

ManagerImpl类的方法

withdrawals()---取款方法（实现“余额不足”逻辑判断）

deposit()---存款方法（实现“存款不能为负”逻辑判断）

包名:com.cx.bank.model (模型层)

MoneyBean 实体类（该类有一个money属性）（单例实现）

# 银行管理系统1.1

1. 在业务层加退出功能。
2. 面向接口编程：

A、定义业务层接口。

B、测试层通过定义业务接口类型变量实现测试层和业务具体实现解耦。

C、控制层通过接口访问业务层。

# 银行管理系统1.2

1. 增加工具包”com.cx.bank.util”
2. 在工具包中添加
   1. InvalidDepositException

作用：存款为负数时抛出异常

* 1. AccountOverDrawnException

作用：取款超出余额时抛出异常

1. 把业务层的异常抛给测试层处理

# 银行管理系统1.3

* 模型层：增加UserBean类，用于封装用户数据
* 业务层：
  + 1. 增加转账功能：

方法名：transfer()

功能需求：用户A转账给用户B，A的余额不能低于转账金额，转账完成后A余额要对应减少，B余额对应增加。

* + 1. 增加注册功能

方法名：register()

功能需求：

* 用户在登录、注册界面选择注册功能
* 用户输入用户名、密码
* 判断name.properties文件是否存在
* 当存在说明用户已经注册，否则创建properties对象。
* 从本地读取properties模板，将数据load到properties对象中。
* 将用户名、密码设置到properties对象中
* 创建文件输出流，创建name.proerties文件。
* 注册成功
  + 1. 增加登录功能

方法名：login()

功能需求：

* 用户在登录注册界面选择登录功能
* 用户输入用户名、密码
* 判断用户是否存在、密码是否正确
* 用户登录成功后进入管理界面
  + 1. 实现业务层单例
  + 测试层：增加登录注册界面
  + 持久层：
    1. 增加持久层(单例实现)

包名：com.cx.bank.dao

* + 1. 在持久层增加持久层接口

接口名：BankDaoInterface

功能说明：

存储余额说明—saveMoney()

用户注册说明—register()

用户登录说明—login()

获取余额说明—getBalance()

转账说明—transfer()

* + 1. 在持久层增加持久层接口实现(文件存储)

类名：BankDaoImpl

功能说明：

存储余额说明—saveMoney()

用户注册说明—register()

用户登录说明—login()

获取余额说明—getBalance()

转账说明—transfer()

* 工具包：增加MD5类，用于对用户密码进行加密(已经上传到压缩包中)

备注：

1.存储文件模板

Bank.properties

文件内容

password=123

money=10.0

userName=Bank

2.其他资料在压缩包中

注意：

该版本项目完成后以“学号”+“姓名”+“项目版本”命名文件夹，并打成压缩文件发给自己班级对应负责人。负责人收齐后再发给我。

1. 该版本项目要求在7月11之前完成。在去深圳之前需要收齐