

管家婆家庭记账软件

1. 项目介绍

1.1. 项目安排

1.1.1. 项目目标

本项目为 JAVAEE 基础班综合项目，包含了若干个知识点，达到将基础班所学知识综合使用，提高了我们对项目的理解与知识点的运用。

熟练 View 层、Service 层、Dao 层之间的方法相互调用操作、
熟练 dbutils 操作数据库表完成增删改查

通过本项目，让我们了解公司项目开发的流程，充分的掌握项目需求分析、设计与功能的代码实现。提高同学们独立分析需求与功能实现的能力。

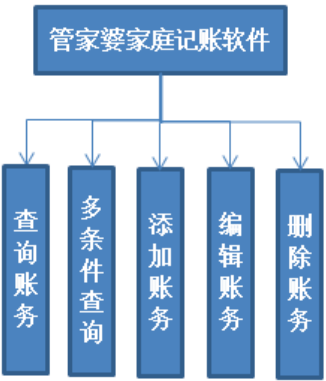
1.1.2. 项目时间

本项目通过 1 天课程时间完成。

1.2. 项目功能分析

1.2.1. 项目演示

1.2.2. 项目功能介绍



- 查询账务
- 多条件组合查询账务
- 添加账务
- 编辑账务
- 删除账务

2. 项目环境搭建

2.1. 技术选型和 jar 包介绍

每个项目都要使用一些已经成熟的技术，它们通常是由一些专业组织或团队所提供的开源免费技术。在今后的学习过程中，我们会逐渐对这些专业组织有所了解。本项目中使用的技术如下：

- apache 的 commons 组件：
 - commons-dbutils-1.4.jar：封装并简化了 JDBC；
 - commons-dbcp-1.4.jar：apache commons 提供的数据库连接池组件，命名为 DBCP；
 - commons.pool-1.3.jar：DBCP 连接池依赖该 jar 包；
- mysql-connector-java-5.1.28-bin.jar：MySQL 的 JDBC 驱动包，用 JDBC 连接 MySQL 数据库必须使用该 JAR 包。

2.2. 工具类介绍

每个项目都会用到一些工具类，本项目也不例外。我们不会教大家如何实现这些类，而是让大家知道在项目我们如何使用这些工具类，下面是本项目中所使用的工具类以及介绍：

- JDBCUtils：用来创建数据库连接池对象；

在项目的实现过程中，我们会说明工具类的使用，这里就不再多介绍。

JDBCUtils.java

```
public class JDBCUtils {  
    public static final String DRIVER_CLASS_NAME = "com.mysql.jdbc.Driver";  
    public static final String URL = "jdbc:mysql://localhost:3306/gjp";  
    public static final String USERNAME = "root";  
    public static final String PASSWORD = "root";  
  
    private static BasicDataSource dataSource = new BasicDataSource();  
    static {  
        dataSource.setDriverClassName(DRIVER_CLASS_NAME);  
        dataSource.setUrl(URL);  
        dataSource.setUsername(USERNAME);  
        dataSource.setPassword(PASSWORD);  
    }  
}
```

```
public static DataSource getDataSource() {  
    return dataSource;  
}  
}
```

2.3. 数据表创建

对一个项目而言，表设计是非常重要的，因为应用程序中所有的操作都是基于数据库表而进行的，所以我们第一步就是创建数据库表。

管家婆项目的数据库设计很简单，我们只需找到 gjp.sql 文件，然后执行之即可。下面是创建库及表的 SQL 语句：

2.3.1. 创建数据库

创建管家婆数据库 gjp

```
CREATE DATABASE gjp;
```

2.3.2. 创建数据库表

创建账务表 gjp_zhangwu

```
CREATE TABLE gjp_zhangwu (  
    zwid INT PRIMARY KEY AUTO_INCREMENT,  
    flname VARCHAR(200),  
    money DOUBLE,  
    zhangHu VARCHAR(100),  
    createtime DATE,  
    description VARCHAR(1000)  
);
```

2.3.3. 添加表数据

添加账务表数据

```
INSERT INTO gjp_zhangwu(zwid,flname,money,zhangHu,createtime,description) VALUES  
(1,'吃饭支出',247,'交通银行','2016-03-02','家庭聚餐');  
INSERT INTO gjp_zhangwu(zwid,flname,money,zhangHu,createtime,description) VALUES  
(2,'工资收入',12345,'现金','2016-03-15','开工资了');  
INSERT INTO gjp_zhangwu(zwid,flname,money,zhangHu,createtime,description) VALUES  
(3,'服装支出',1998,'现金','2016-04-02','买衣服');  
INSERT INTO gjp_zhangwu(zwid,flname,money,zhangHu,createtime,description) VALUES
```

```

(4, '吃饭支出', 325, '现金', '2016-06-18', '朋友聚餐');
INSERT INTO gjp_zhangwu(zwid, flname, money, zhangHu, createtime, description) VALUES
(5, '股票收入', 8000, '工商银行', '2016-10-28', '股票大涨');
INSERT INTO gjp_zhangwu(zwid, flname, money, zhangHu, createtime, description) VALUES
(6, '股票收入', 5000, '工商银行', '2016-10-28', '股票又大涨');
INSERT INTO gjp_zhangwu(zwid, flname, money, zhangHu, createtime, description) VALUES
(7, '工资收入', 5000, '交通银行', '2016-10-28', '又开工资了');
INSERT INTO gjp_zhangwu(zwid, flname, money, zhangHu, createtime, description) VALUES
(8, '礼金支出', 5000, '现金', '2016-10-28', '朋友结婚');
INSERT INTO gjp_zhangwu(zwid, flname, money, zhangHu, createtime, description) VALUES
(9, '其他支出', 1560, '现金', '2016-10-29', '丢钱了');
INSERT INTO gjp_zhangwu(zwid, flname, money, zhangHu, createtime, description) VALUES
(10, '交通支出', 2300, '交通银行', '2016-10-29', '油价还在涨啊');
INSERT INTO gjp_zhangwu(zwid, flname, money, zhangHu, createtime, description) VALUES
(11, '吃饭支出', 1000, '工商银行', '2016-10-29', '又吃饭');
INSERT INTO gjp_zhangwu(zwid, flname, money, zhangHu, createtime, description) VALUES
(12, '工资收入', 1000, '现金', '2016-10-30', '开资');
INSERT INTO gjp_zhangwu(zwid, flname, money, zhangHu, createtime, description) VALUES
(13, '交通支出', 2000, '现金', '2016-10-30', '机票好贵');
INSERT INTO gjp_zhangwu(zwid, flname, money, zhangHu, createtime, description) VALUES
(14, '工资收入', 5000, '现金', '2016-10-30', '又开资');

```

2.4. 项目分层（分包）的作用

程序为什么要分包分层？

以顾客去饭店吃饭案例分析一下：

小饭店：一个服务员搞定(接待顾客\点菜\炒菜)

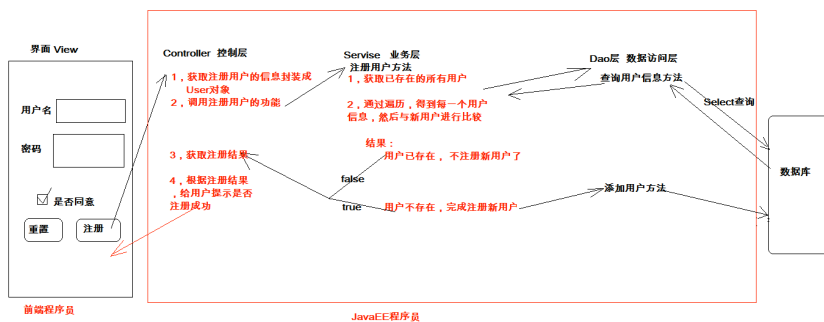
大饭店：

- 迎宾员(是否有预定\ 询问吃中餐还是西餐或者烧烤等\ 几位用餐 \ 领路到指定的包间\ 找点菜员为顾客点菜)
- 点菜员(记录顾客点餐内容\ 记录是否有忌口等问题\ 找厨师为顾客炒菜)
- 厨师(按照顾客菜肴清单,进行炒菜)

通过案例发现，当程序规模小的时候，可以一个人全部完成；但程序规模大的时候，一个人难以完成，这时，要采用多人合作的方式来完成程序开发。

多人合作方式将会碰到工作任务分配问题，这时我们会想，每个人负责完成项目的一块内容就可以了。那么，这一块块内容的划分，就需要我们采用分层（分包）的方式完成了。

通过下图（用户注册功能）来讲解下，项目中常见的分层（分包）。



- view 层作用: 视图层,即项目中的界面
- controller 层作用: 控制层, 获取界面上的数据,为界面设置数据; 将要实现的功能交给业务层处理
- service 层作用: 业务层, 功能的实现, 与 controller 控制层和数据访问层 DAO 交互, 将对数据库的操作交给 DAO 数据访问层来处理
- dao 层作用: 数据访问层, 用来操作数据库表的数据
- db 数据库: 这里指 MySQL
- domain 实体包: 存放 JavaBean
- tools 工具包: 存放项目中使用到的工具类
- test 测试包: 存放项目功能测试的代码

2.5. 工程创建及包管理

1. 使用 Eclipse 创建 Java 工程, 命名为 gjp
2. 创建工程包
 - cn.itcast.gjp.app: 存放 main 方法类;
 - cn.itcast.gjp.domain: 存放 JavaBean;
 - cn.itcast.gjp.view: 存放界面, 及表现层类;
 - cn.itcast.gjp.service: 存放业务层类;
 - cn.itcast.gjp.dao: 存放数据访问层类;
 - cn.itcast.gjp.tools: 存放工具类
3. 创建 lib 文件夹, 用来存储使用的 jar 包

3. 功能模块

3.1. 相关类创建

完成本项目中类的创建, 无需在类中添加代码。

1. 复制已编写好的工具类 JDBCUtils.java 到 tools 包中;
2. 复制 jar 包 `mysql-connector-java-5.1.28-bin.jar`、`commons-dbutils-1.4.jar`、`commons-dbcp-1.4.jar`、`commons-pool-1.3.jar`, 到 lib 文件夹中, 通过 Build Path 操作,

Comment [L1]: MySQL 的 JDBC 驱动包, 用 JDBC 连接 MySQL 数据库必须使用该 JAR 包

Comment [L2]: 封装并简化了 JDBC

Comment [L3]: apache commons 提供的数据库连接池组件, 命名为 DBCP

Comment [L4]: DBCP 连接池依赖该 jar 包

添加到 classPath 路径中，提供给 JDBCUtils 使用；

3. 在 app 包中，创建类 MainApp.java，编写 main 主方法，用来完成本项目的启动
4. 在 domain 包中，创建类 ZhangWu.java，它是用来封装账务信息的 JavaBean。
5. 在 dao 包中，创建类 ZhangWuDao.java，给 ZhangWuDao 类添加一个成员变量 QueryRunner 对象，因为我们使用 dbutils 来操作数据库。
6. 在 service 包中，创建类 ZhangWuService.java，给 ZhangWuService 类添加一个类型为 ZhangWuDao 的成员变量，因为 service 依赖 dao。
7. 在 view 包中，创建类 MainView.java，给 MainView 类添加一个类型为 ZhangWuService 的成员变量，因为本项目中 view 依赖 service。

编写 app 包中 MainApp.java

```
/**
 * 主方法类
 */
public class MainApp {
    public static void main(String[] args) {
    }
}
```

编写 domain 包中 ZhangWu.java

```
/**
 * 账务类
 */
public class ZhangWu {
}
```

编写 Dao 包中 ZhangWuDao.java

```
/**
 * 账务数据层类
 */
public class ZhangWuDao{
    //获取数据库连接池，得到操作表数据的对象 QueryRunner
    QueryRunner qr = new QueryRunner(JDBCUtils.getDataSource());
}
```

编写 Service 包，ZhangWuService.java

```
/**
 * 账务业务层类
 */
public class ZhangWuService {
    //service都依赖dao
    private ZhangWuDao zhangWuDao = new ZhangWuDao();
}
```

编写 view 包, MainView.java

```
public class MainView {  
    //本项目中view依赖service  
    private ZhangWuService zhangWuService = new ZhangWuService();  
}
```

3.2. 账务 JavaBean

JavaBean 是指的是 Java 中的类, 该类中的成员变量与数据库表中的字段相对应(变量名对应数据库表字段名、变量数据类型对应数据库表字段类型), 并提供空参数构造方法、set、get 方法。

我们现在来完成账务类的代码编写, 如下:

```
/**  
 * 账务类  
 */  
public class ZhangWu {  
    private int zwid;// id  
    private String flname;// 分类名称  
    private String zhangHu;// 账户名称  
    private double money;// 金额  
    private String createtime;// 创建时间  
    private String description;// 说明  
  
    public String getFlname() {  
        return flname;  
    }  
    public void setFlname(String flname) {  
        this.flname = flname;  
    }  
    public String getZhangHu() {  
        return zhangHu;  
    }  
    public void setZhangHu(String zhangHu) {  
        this.zhangHu = zhangHu;  
    }  
    public int getZwid() {  
        return zwid;  
    }  
    public void setZwid(int zwid) {  
        this.zwid = zwid;  
    }  
    public double getMoney() {  
        return money;  
    }  
}
```

```
    }  
    public void setMoney(double money) {  
        this.money = money;  
    }  
    public String getCreatetime() {  
        return createtime;  
    }  
    public void setCreatetime(String createtime) {  
        this.createtime = createtime;  
    }  
    public String getDescription() {  
        return description;  
    }  
    public void setDescription(String description) {  
        this.description = description;  
    }  
}
```

3.3. 功能界面菜单

```
-----管家婆家庭记账软件-----  
1.添加账务  2.编辑账务  3.删除账务  4.查询账务  5.退出系统  
请输入要操作的功能序号[1-5]:
```

界面菜单的完成，是项目编写的第一步。

我们通过输出语句，完成界面菜单的输出，之后再根据输入的功能序号，进行对应功能的调用执行。

3.3.1. 功能实现步骤

- 1 编写 MainView 类 run 方法
 - 1.1 完成功能界面菜单显示
 - 1.2 接收键盘输入的功能选项
 - 1.3 根据选项值，调用对应的功能方法
- 2 编写 MainApp 类的 main 主方法
 - 2.1 调用 MainView 类中 run 方法，实现将程序执行起来，显示功能界面菜单。

3.3.2. 功能实现代码

1.在 view 包 MainView 类中编写 run 方法

```
/**  
 * 运行方法
```



```

*/
public void run() {
    /*
     * 1. 打印菜单 2. 获取用户输入 3. 调用对应方法
     */
    boolean flag = true;
    Scanner in = new Scanner(System.in);
    while (flag) {
        System.out.println("-----管家婆家庭记账软件-----");
        System.out.println(1.添加账务 2.编辑账务 3.删除账务 4.查询账务 5.退出系统");
        System.out.println("请输入要操作的功能序号[1-5]:");
        int op = in.nextInt();
        switch (op) {
            case 1:
                addZhangWu();
                break;
            case 2:
                editZhangWu();
                break;
            case 3:
                deleteZhangWu();
                break;
            case 4:
                selectZhangWu();
                break;
            case 5:
                System.out.println("再见! ");
                flag = false;
                break;
            default:
                System.out.println("输入错误! ");
        }
    }
}
}

```

2.app 包 MainApp 类的 main 主方法中，调用 MainView 类中 run 方法

```

public static void main(String[] args) {
    new MainView().run();
}

```

3.4. 查询所有账务



3.4.1. 功能实现步骤

- 1 编写 MainView 类中 selectZhangWu 方法
 - 1.1 通过输出语句，显示出要查询账务的方式
 - 1.2 接收键盘的输入项，调用对应的方法（1.查询所有 2.按条件查询）
- 2 编写 MainView 类中 selectAll 查询所有账务方法
 - 2.1 调用 ZhangWuService 类 selectAll 方法，返回包含所有账务数据的 List<ZhangWu>集合
 - 2.2 调用 MainView 类中 print 方法，实现控制台显示所有账务数据
- 3 编写 MainView 类中 print 方法
 - 3.1 使用输出语句，打印出账务表的表头名称

ID	类别	账户	金额	时间	说明
----	----	----	----	----	----
 - 3.2 遍历账务集合，将每个账务信息输出打印
- 4 编写 ZhangWuService 类中 selectAll 方法
 - 4.1 调用 ZhangWuDao 类中 selectAll 方法，返回包含所有账务数据的 List<ZhangWu>集合
- 5 编写 ZhangWuDao 类中 selectAll()方法
 - 5.1 通过 QueryRunner 对象，调用 query 方法查询数据库表 gjp_zhangwu，返回包含所有账务数据的 List<ZhangWu>集合

3.4.2. 功能实现代码

- 1. 编写 MainView 类中 selectZhangWu 方法

```
/**
 * 查询账务方法
 */
```

```

public void selectZhangWu() {
    System.out.println("1.查询所有 2.按条件查询");
    Scanner in = new Scanner(System.in);
    int op = in.nextInt();
    switch (op) {
        case 1:
            selectAll();
            break;
        case 2:
            select();
            break;
        default:
            System.out.println("输入错误! ");
    }
}
}

```

2. 编写 MainView 类中 selectAll 方法

```

/**
 * 查询所有账务方法
 */
public void selectAll() {
    List<ZhangWu> zhangWuList = zhangWuService.selectAll();
    print(zhangWuList);
}

```

3. 编写 MainView 类中 print 方法

```

/*
 * 输出账务方法
 */
private void print(List<ZhangWu> zhangWuList) {
    System.out.println("ID\t类别\t\t账户\t\t金额\t\t时间\t\t说明");
    for (ZhangWu zw : zhangWuList) {
        System.out.println(zw.getZwid() + "\t" + zw.getFlname() + "\t\t"
            + zw.getZhangHu() + "\t\t" + zw.getMoney() + "\t\t"
            + zw.getCreatetime() + "\t" + zw.getDescription());
    }
}
}

```

4. 编写 ZhangWuService 类中 selectAll 方法

```

/**
 * 查询所有
 */
public List<ZhangWu> selectAll() {
    return zhangWuDao.selectAll();
}

```

```
}
```

5. 编写 ZhangWuDao 类中 selectAll()方法

```
/**
 * 查询所有账务
 */
public List<ZhangWu> selectAll() {
    String sql = "select * from gjp_zhangwu";
    try {
        return qr.query(sql, new BeanListHandler<ZhangWu>(ZhangWu.class));
    } catch (SQLException e) {
        throw new RuntimeException(e);
    }
}
```

3.5. 多条件查询账务

```
-----管家婆家庭记账软件-----
1. 添加账务  2. 编辑账务  3. 删除账务  4. 查询账务  5. 退出系统
请输入要操作的功能序号[1-5]:4
1. 查询所有  2. 按条件查询
2
请输入 查询起始时间: 20160101
请输入 查询结束时间: 20160630
ID      类别      账户      金额      时间      说明
1       吃饭支出    交通银行    247.0     2016-03-02    家庭聚餐
2       工资收入    现金      12345.0   2016-03-15    开工资了
3       服装支出    现金      1998.0    2016-04-02    买衣服
4       吃饭支出    现金      325.0     2016-06-18    朋友聚餐
```

3.5.1. 功能分析

- 编写 MainView 类中 select 方法
 - 通过键盘输入查询日期的范围
 - 调用 ZhangWuService 类中 select 方法，返回查询日期范围内的账务信息集合 List<ZhangWu>
 - 调用 MainView 类中的 print 方法，将账务信息集合中的内容显示在控制台中
- 编写 ZhangWuService 类中 select 方法
 - 调用 ZhangWuDao 类中 select 方法，返回查询日期范围内的账务信息集合 List<ZhangWu>
- 编写 ZhangWuDao 类中 select 方法
 - 通过 QueryRunner 对象，调用 query 方法查询数据库表 gjp_zhangwu，返回包含查询日期范围内的账务数据 List<ZhangWu>集合

3.5.2. 功能实现步骤

1. 编写 MainView 类中 select 方法

```
/**
 * 按条件查询账务方法
 */
public void select() {
    /**
     * 1. 获取用户输入查询日期范围。
     * 2. 调用service的select()方法完成条件查询功能
     */
    Scanner in = new Scanner(System.in);
    System.out.print("请输入查询起始时间: ");
    String start = in.next();
    System.out.print("请输入查询结束时间: ");
    String end = in.next();
    List<ZhangWu> zhangWuList = zhangWuService.select(start, end);
    print(zhangWuList);
}
```

2. 编写 ZhangWuService 类中 select 方法

```
/**
 * 按条件查询账务方法
 * @param start
 * @param end
 * @return
 */
public List<ZhangWu> select(String start, String end) {
    return zhangWuDao.select(start, end);
}
```

3. 编写 ZhangWuDao 类中 select 方法

```
/**
 * 按条件查询账务方法
 * @param start
 * @param end
 * @return
 */
public List<ZhangWu> select(String start, String end) {
    String sql = "select * from gjp_zhangwu where createtime between ? and ?";
    Object[] params = {start, end};
    try {
        return qr.query(sql, new BeanListHandler<ZhangWu>(ZhangWu.class),
            params);
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
```

```
    } catch (SQLException e) {  
        throw new RuntimeException(e);  
    }  
}
```

3.6. 添加账务

```
-----管家婆家庭记账软件-----  
1.添加账务 2.编辑账务 3.删除账务 4.查询账务 5.退出系统  
请输入要操作的功能序号[1-5]:1  
请输入类别:彩票支出  
请输入账户:现金  
请输入金额:200  
请输入时间:20160701  
请输入说明:欧洲杯  
添加账务成功!
```

3.6.1. 功能分析

- 1 编写 MainView 类中 addZhangWu 方法
 - 1.1 键盘输入新添加的账务信息
 - 1.2 调用 ZhangWuService 类中 addZhangWu 方法，用来指定账务的添加
 - 1.3 添加完毕后，使用输出语句，提示“添加账务成功!”
- 2 编写 ZhangWuService 类中 addZhangWu 方法
 - 2.1 调用 ZhangWuDao 类中 addZhangWu 方法，用来指定账务的添加
- 3 编写 ZhangWuDao 类中 addZhangWu 方法
 - 3.1 通过 QueryRunner 对象，调用 update 方法更新数据库表 gjp_zhangwu，完成指定账务添加到数据库表中

3.6.2. 功能实现步骤

1. 编写 MainView 类中 addZhangWu 方法

```
/**  
 * 添加账务方法  
 */  
public void addZhangWu() {  
    /**  
     * 1. 获取用户输入，封装到bean中。  
     * 2. 调用service的addZhangWu()方法完成添加功能  
     */  
    Scanner in = new Scanner(System.in);  
    ZhangWu zw = new ZhangWu();
```

```

        System.out.print("请输入类别: ");
        zw.setFlname(in.next());
        System.out.print("请输入账户: ");
        zw.setZhangHu(in.next());
        System.out.print("请输入金额: ");
        zw.setMoney(in.nextDouble());
        System.out.print("请输入时间: ");
        zw.setCreatetime(in.next());
        System.out.print("请输入说明: ");
        zw.setDescription(in.next());

        zhangWuService.addZhangWu(zw);
        System.out.println("添加账务成功! ");
    }
}

```

2. 编写 ZhangWuService 类中 addZhangWu 方法

```

/**
 * 添加账务
 * @param zw
 */
public void addZhangWu(ZhangWu zw) {
    zhangWuDao.addZhangWu(zw);
}

```

3. 编写 ZhangWuDao 类中 addZhangWu 方法

```

/**
 * 添加账务
 * @param zw
 */
public void addZhangWu(ZhangWu zw) {
    String sql = "insert into gjp_zhangwu(flname,money,zhanghu,createtime,description) values(?,?,?,?,?)";
    try {
        Object[] params =
{zw.getFlname(),zw.getMoney(),zw.getZhangHu(),zw.getCreatetime(),zw.getDescription()};
        qr.update(sql, params);
    } catch (SQLException e) {
        throw new RuntimeException(e);
    }
}
}

```

3.7. 编辑账务

```
-----管家婆家庭记账软件-----
1.添加账务 2.编辑账务 3.删除账务 4.查询账务 5.退出系统
请输入要操作的功能序号[1-5]:2
请输入ID: 1
请输入新类别: 吃饭支出
请输入新账户: 交通银行
请输入新金额: 2500
请输入新时间: 20160701
请输入新说明: 家庭大聚餐
编辑账务成功!
```

3.7.1. 功能分析

- 1 编写 MainView 类中 editZhangWu 方法
 - 1.1 键盘输入要编辑的账务信息 ID 号
 - 1.2 键盘输入要修改的账务信息内容
 - 1.3 调用 ZhangWuService 类中 editZhangWu 方法，用来将指定的账务信息进行更新
 - 1.4 更新完毕后，使用输出语句，提示 “编辑账务成功!”
- 2 编写 ZhangWuService 类中 editZhangWu 方法
 - 2.1 调用 ZhangWuDao 类中 editZhangWu 方法，用来将指定的账务信息进行更新
- 3 编写 ZhangWuDao 类中 editZhangWu 方法
 - 3.1 通过 QueryRunner 对象，调用 update 方法更新数据库表 gjp_zhangwu，完成数据库中指定账务更新操作

3.7.2. 功能实现步骤

1. 编写 MainView 类中 editZhangWu 方法

```
/**
 * 编辑账务方法
 */
public void editZhangWu() {
    /**
     * 1. 获取用户输入，封装到bean中。
     * 2. 调用service的editZhangWu()方法完成添加功能
     */
    Scanner in = new Scanner(System.in);
    ZhangWu zw = new ZhangWu();
    System.out.print("请输入ID: ");
    zw.setZwid(in.nextInt());
    System.out.print("请输入新类别: ");
    zw.setFlname(in.next());
```



```

        System.out.print("请输入新账户: ");
        zw.setZhangHu(in.next());
        System.out.print("请输入新金额: ");
        zw.setMoney(in.nextDouble());
        System.out.print("请输入新时间: ");
        zw.setCreatetime(in.next());
        System.out.print("请输入新说明: ");
        zw.setDescription(in.next());

        zhangWuService.editZhangWu(zw);
        System.out.println("编辑账务成功! ");
    }
}

```

2. 编写 ZhangWuService 类中 editZhangWu 方法

```

/**
 * 编辑账务
 * @param zw
 */
public void editZhangWu(ZhangWu zw) {
    zhangWuDao.editZhangWu(zw);
}

```

3. 编写 ZhangWuDao 类中 editZhangWu 方法

```

/**
 * 编辑账务
 * @param zw
 */
public void editZhangWu(ZhangWu zw) {
    String sql = "update gjp_zhangwu set fname=?,
money=?,zhanghu=?,createtime=?,description=? where zwid=?";
    try {
        Object[] params =
{zw.getFname(),zw.getMoney(),zw.getZhangHu(),zw.getCreatetime(),zw.getDescription(),zw.getZwid()};
        qr.update(sql, params);
    } catch (SQLException e) {
        throw new RuntimeException(e);
    }
}
}

```

3.8. 删除账务

```
-----管家婆家庭记账软件-----
1.添加账务 2.编辑账务 3.删除账务 4.查询账务 5.退出系统
请输入要操作的功能序号[1-5]:3
请输入ID: 2
删除账务成功!
```

3.8.1. 功能分析

- 1 编写 MainView 类中 deleteZhangWu 方法
 - 1.1 键盘输入要删除的账务信息 ID 号
 - 1.2 调用 ZhangWuService 类中 deleteZhangWu 方法，用来将指定的账务信息删除
 - 1.3 删除完毕后，使用输出语句，提示 “删除账务成功!”
- 2 编写 ZhangWuService 类中 deleteZhangWu 方法
 - 2.1 调用 ZhangWuDao 类中 deleteZhangWu 方法，用来将指定的账务信息删除
- 3 编写 ZhangWuDao 类中 deleteZhangWu 方法
 - 3.1 通过 QueryRunner 对象，调用 update 方法更新数据库表 gjp_zhangwu，完成数据库中指定账务删除操作

3.8.2. 功能实现步骤

1. 编写 MainView 类中 deleteZhangWu 方法

```
/**
 * 删除账务方法
 */
public void deleteZhangWu() {
    /**
     * 1. 获取用户输入，封装到bean中。
     * 2. 调用service的deleteZhangWu()方法完成添加功能
     */
    Scanner in = new Scanner(System.in);
    System.out.print("请输入ID: ");

    zhangWuService.deleteZhangWu(in.nextInt());
    System.out.println("删除账务成功! ");
}
```

2. 编写 ZhangWuService 类中 deleteZhangWu 方法

```
/**
 * 删除账务
 * @param zwid
 */
```

```
public void deleteZhangWu(int zwid) {  
    zhangWuDao.deleteZhangWu(zwid);  
}
```

3. 编写 ZhangWuDao 类中 deleteZhangWu 方法

```
/**  
 * 删除账务  
 * @param zwid  
 */  
public void deleteZhangWu(int zwid) {  
    String sql = "delete from gjp_zhangwu where zwid=?";  
    try {  
        qr.update(sql, zwid);  
    } catch (SQLException e) {  
        throw new RuntimeException(e);  
    }  
}
```

3.9. 功能模块总结

恭喜大家，我们已经完成了分类模块所有功能。通过分类模块功能的实现，大家可能会发现如下几个问题：

- 难度大的地方：自行功能分析的能力欠缺，需要讲师来分析；
- 代码实现：
 - View（Controller）：代码量最大；
 - Service 层：代码量最小；
 - DAO 层：相似度高，难度不大；

到现在为止我们是第一次写这么复杂的项目，所以大家之间也没有练习过功能分析的能力，所以第一次需要讲师来分析是正常的，当大家不断练习后，有了正确的思维模式，自然就能分析了。

view 层的作用是“界面”，用来完成数据显示给用户。当前项目 view 层中，包含了 Controller 层代码。

Controller 层的作用是“调度”，调度的是表现层 view 和业务层 Service，主要功能分为：一是把表现层的数据交给业务层处理；二是把业务层返回的数据交给表现层显示。至于为什么这一层 View（Controller）的代码量大，这里我要说明一下，其实不是 View（Controller）层的代码量大，而是其他层代码量少，所以你会感觉 View（Controller）层代码量大。

Service 层的作用是“业务”，我们也可以把“业务”当成是“功能”。那为什么 Service 层代码量少呢？原因是现在我们写的项目很小，没有什么复杂的业务，而今后大家要写的大型项目代码量最大的就是 Service 层。

DAO 层是操作数据库，现在我们使用的是 commons-dbutils 工具来简化 JDBC，所以我们发现代码不多，比较简单。最后我们还会学习其他 DAO 层的工具，例如：hibernate 和 mybatis，他们都是 JDBC 的封装，用来简化 JDBC。

