

**2023-2024学年第一学期**

**《Java程序设计》**

**综合测验**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **姓名** | **学号** | **院系专业** | **班级** | **指导老师** | **成绩** |
| 朱皓天 | 202302210314 | 大数据学院 | 计算机技术与科学 | 吕永强 |  |

基于JDBC的饮料信息维护系统的设计与实现

# 需求分析

基于JDBC的饮料信息维护系统的功能主要包括商品的新增、修改、删除和查询。

# 数据库设计

表2-1 商品信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段类型 | 长度 | 允许空值 | 备注 |
| Band | varchar | 50 | 否 | 饮料的品牌 |
| Cap | varchar | 50 | 是 | 饮料的净含量 |
| Type | varchar | 50 | 是 | 饮料的类型 |
| origin | varchar | 50 | 是 | 饮料的产地 |
| price | double |  | 是 | 饮料的价格 |

# 代码实现

## 查询商品信息核心代码

|  |
| --- |
| */\*\*  \* 查询  \** ***@return*** *\** ***@throws*** *Exception  \*/* @Override *public List*<Drink> queryDrinks() *throws* Exception {  *Connection* conn = getConnection();  String sql = "select \* from t\_drinks";  *PreparedStatement* ps = conn.prepareStatement(sql);  *ResultSet* rs = ps.executeQuery();  *List*<Drink> list = *new* ArrayList<>();  *while* (rs.next()){  Drink drink = *new* Drink();  drink.setBand(rs.getString("band"));  drink.setOrigin(rs.getString("origin"));  drink.setCap(rs.getString("cap"));  drink.setType(rs.getString("type"));  drink.setPrice(rs.getDouble("price"));  list.add(drink);  }  ps.close();  rs.close();  *return* list; } |

## 新增商品信息核心代码

|  |
| --- |
| */\*\*  \* 增加  \** ***@param drink*** *\** ***@throws*** *Exception  \*/* @Override *public void* addDrinks(Drink drink) *throws* Exception {  *Connection* conn = getConnection();  String sql = "insert into t\_drinks VALUES (?,?,?,?,?);";  *PreparedStatement* ps = conn.prepareStatement(sql);  ps.setString(1, drink.getBand());  ps.setString(2, drink.getOrigin());  ps.setString(3, drink.getCap());  ps.setString(4,drink.getType());  ps.setDouble(5,drink.getPrice());  ps.executeUpdate();  ps.close();  conn.close(); } |

## 修改商品信息核心代码

|  |
| --- |
| */\*\*  \* 修改  \** ***@param drink*** *\** ***@throws*** *Exception  \*/* @Override *public void* updateDrinks(Drink drink) *throws* Exception {  *Connection* conn = getConnection();  String sql = "update t\_drinks set cap = ? ,price = ? where band = ?";  *PreparedStatement* ps = conn.prepareStatement(sql);  ps.setString(1,drink.getCap());  ps.setDouble(2,drink.getPrice());  ps.setString(3,drink.getBand());  ps.executeUpdate();  ps.close();  conn.close(); } |

## 删除商品信息核心代码

|  |
| --- |
| */\*\*  \* 删除  \** ***@param band*** *\** ***@throws*** *Exception  \*/* @Override *public void* delDrinks(String band) *throws* Exception {  *Connection* conn = getConnection();  String sql = "delete from t\_drinks where band = ?";  *PreparedStatement* ps = conn.prepareStatement(sql);  ps.setString(1,band);  ps.executeUpdate();  ps.close();  conn.close(); } |

# 系统测试

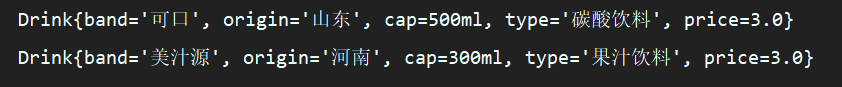
## 查询商品信息测试

（1）查询饮料信息测试用例

表4-1 查询商品功能测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例名称 | 用例说明 | 数据 | 预测结果 | 测试结果 |
| TestQuery | 运行测试查询按钮 | 无 | 查询2条数据 |  |

（2）查询商品信息测试结果



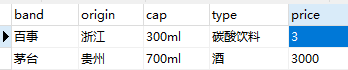
## 新增商品信息测试

（1）新增饮料信息测试用例

表4-2 新增商品功能测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例名称 | 用例说明 | 数据 | 预测结果 | 测试结果 |
| TestAdd | 测试功能输入商品信息，运行测试按钮 | 饮料品牌：茅台  饮料类型：酒  饮料价格：3000  饮料净含量：700ml  饮料产地：贵州 | 数据库新增一条信息 | 数据库新增一条信息 |

（2）新增饮料信息测试结果

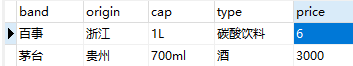


## 修改商品信息测试

（1）更新饮料信息测试用例

表4-3 更新商品功能测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例名称 | 用例说明 | 数据 | 预测结果 | 测试结果 |
| TestUpdate | 测试功能更新商品信息，运行测试按钮 | 饮料品牌：百事  饮料价格：6  饮料净含量：1L | 数据库品牌为百事的一条信息被修改 | 数据库品牌为百事的一条信息被修改 |

（2）更新饮料信息测试结果

## 删除商品信息测试

（1）删除饮料信息测试用例

表4-4 删除商品功能测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例名称 | 用例说明 | 数据 | 预测结果 | 测试结果 |
| TestDel | 测试功能删除商品信息，运行测试按钮 | 饮料品牌：茅台 | 数据库品牌为茅台的一条信息被删除 | 数据库品牌为茅台的一条信息被删除 |

（2）删除饮料信息测试结果



# 总结