



面向对象与多线程综合实验

结合华为云平台的基于JAVASE数
据挖掘系统设计与实现



实验内容

1. 做什么

*GUI界面优化（选做）

日志 物流

用户名:

登录地点:

登录IP:

登录状态: ☐ 登录 ☒ 登出

2. 怎么做?

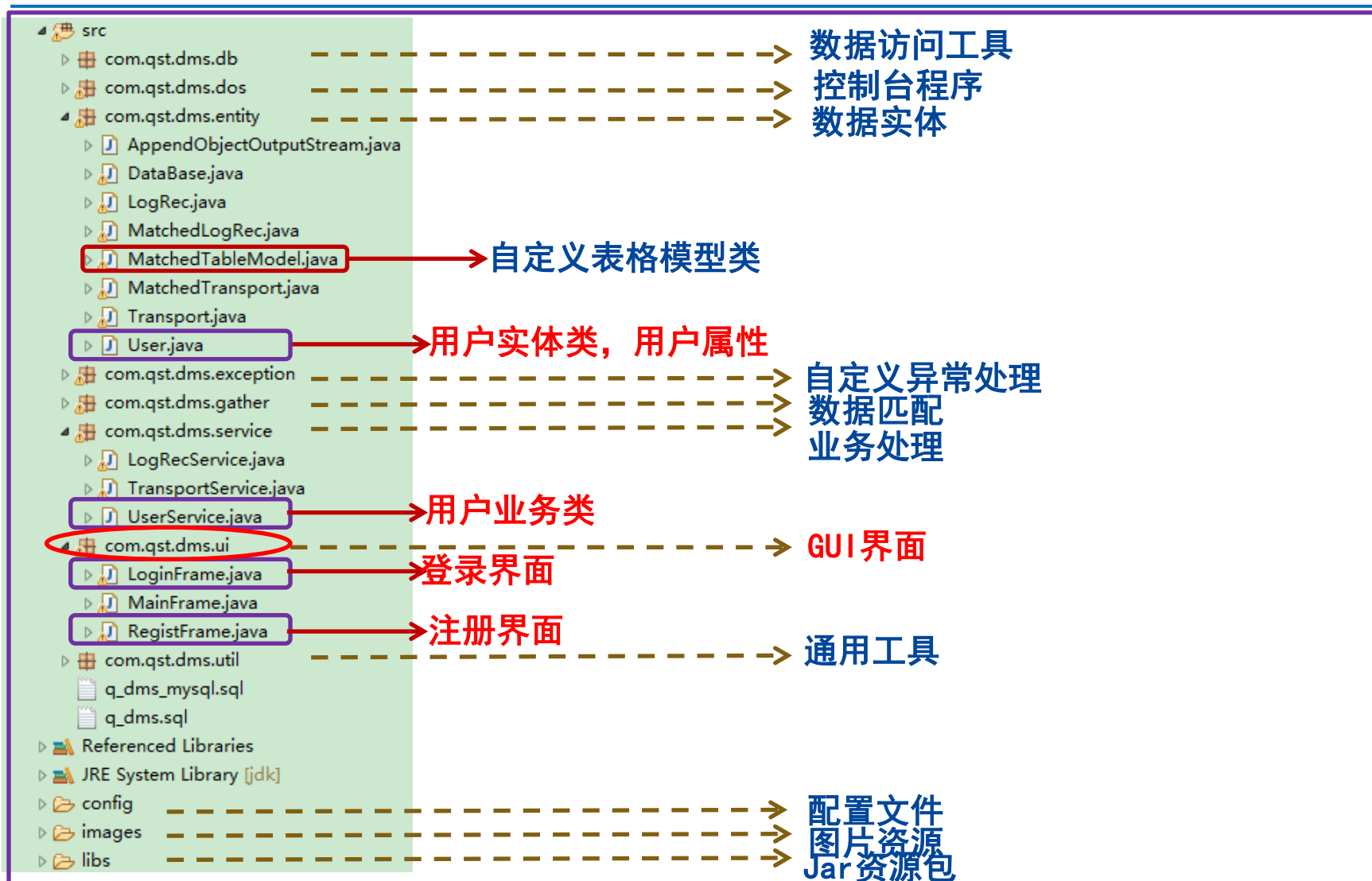
- ◆ 日志和物流数据采集界面，不需要手动输入ID单号，在插入新记录时自动生成数据库主键ID值，并将新生成的ID获取出来（不改变已有的entity数据类）
- ◆ 数据显示表格自定义字段显示

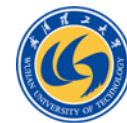
3. 对应项

- ◆ MainForm
- ◆ LogRec/TransportService SaveMatchLogToDB() 方法中由 db. **executeUpdate** 改为 **executeSQLAndReturnPrimaryKey** 方法用来执行插入一条记录后，同时返回记录的主键
- ◆ Entity类中 matchedTableModel 类中更改表头信息
- ◆ Service类中 readLogResult() 方法按需从数据库中获取结果集



实验内容系统框架





1.采集界面删除与ID输入有关组件

日志和物流数据采集界面，不需要手动输入ID单号，单号由数据库自动生成（不改变已有的entity数据类）

欢迎进入数据挖掘系统客户端！

操作 帮助

采集数据 匹配日志数据 匹配物流数据 保存数据 发送数据 显示数据

日志 物流

用户名：

登录地点：

登录IP：

登录状态： ☒ 登录 ☐ 登出

确认 重置

目的地：

经手人：

收货人：

物流状态： 发货中 ▼

确认 重置



新增记录保存到数据库后返回自增ID

executeSQLAndReturnPrimaryKey

与数据库的主键值绑定

```
// 匹配日志信息保存到数据库, 参数是集合
public void saveMatchLogToDB(ArrayList<MatchedLogRec> matchLogs) {
    DBUtil db = new DBUtil();
    try {
        // 获取数据库链接
        db.getConnection();
        for (MatchedLogRec matchedLogRec : matchLogs) {
            // 获取匹配的登录日志
            LogRec login = matchedLogRec.getLogin();
            // 获取匹配的登出日志
            LogRec logout = matchedLogRec.getLogout();
            // 保存匹配记录中的登录日志
            /*String sql = "INSERT INTO gather_logrec(id,time,address,type,username,ip,logtype) VALUES(?,?,?,?,?,?,?)";
            Object[] param = new Object[] { login.getId(),
                new Timestamp(login.getTime().getTime()),
                login.getAddress(), login.getType(), login.getUser(),
                login.getIp(), login.getLogType() };*/
            String sql = "INSERT INTO gather_logrec(time,address,type,username,ip,logtype) VALUES(?,?,?,?,?,?)";
            Object[] param = new Object[] {
                new Timestamp(login.getTime().getTime()),
                login.getAddress(), login.getType(), login.getUser(),
                login.getIp(), login.getLogType() };
            int loginKey=db.executeSQLAndReturnPrimaryKey(sql, param);
            login.setId(loginKey);
            // 保存匹配记录中的登出日志
            param = new Object[] {
                new Timestamp(logout.getTime().getTime()),
                logout.getAddress(), logout.getType(),
                logout.getUser(), logout.getIp(), logout.getLogType() };
            int logoutKey=db.executeSQLAndReturnPrimaryKey(sql, param);
            logout.setId(logoutKey);
            // 保存匹配日志的ID
            sql = "INSERT INTO matched_logrec(loginid,logoutid) VALUES(?,?)";
            param = new Object[] { login.getId(), logout.getId() };
            db.executeUpdate(sql, param);
        }
        // 关闭数据库连接, 释放资源
        db.closeAll();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

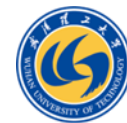


新增记录保存到数据库后返回自增ID

Transport和LogService中，保存匹配的记录到数据库同时返回数据库ID信息

getGeneratedKeys()

```
/*获得主键对应的结果集*/
public int executeSQLAndReturnPrimaryKey(String preparedSql, Object[] param) {
    // 处理SQL,执行SQL
    try {
        // 得到PreparedStatement对象
        pstmt = conn.prepareStatement(preparedSql, Statement.RETURN_GENERATED_KEYS);
        if (param != null) {
            for (int i = 0; i < param.length; i++) {
                // 为预编译sql设置参数
                pstmt.setObject(i + 1, param[i]);
            }
        }
        // 执行SQL语句
        pstmt.executeUpdate();
        rs = pstmt.getGeneratedKeys();
        if(rs.next())
        {
            return rs.getInt(1);
        }
    } catch (SQLException e) {
        // 处理SQLException异常
        e.printStackTrace();
    }
    return -1;
}
```



自定义显示日志和物流表格数据

自行设计并显示日志和物流表格（字段）数据

欢迎进入数据挖掘系统客户端！

操作 帮助

采集数据 匹配日志数据 匹配物流数据 保存数据 发送数据 显示数据

日志 物流

用户名	登录时间	登出时间	登录地点
yyc	2020-11-26 16:41:01.0	2020-11-26 16:41:03.0	为
again发	2020-11-26 17:25:17.0	2020-11-26 17:25:19.0	打
yyc	2020-11-26 17:55:10.0	2020-11-26 17:55:12.0	而
投入	2020-11-26 18:12:32.0	2020-11-26 18:12:34.0	而他

欢迎进入数据挖掘系统客户端！

操作 帮助

采集数据 匹配日志数据 匹配物流数据 保存数据 发送数据 显示数据

日志 物流

物流ID	采集时间	目的地	状态	经手人	收货人	物流类型
4	2020-11-26 ...	wuhan	2	temp3	temp	1
5	2020-11-26 ...	wuhan	2	temp1	temp	2
6	2020-11-26 ...	wuhan	2	temp2	temp	3
7	2020-11-26 ...	水电费	2	第三方	1	1
8	2020-11-26 ...	水电费	2	第三方	1	2
9	2020-11-26 ...	水电费	2	第三方	1	3