目录

[1. 软件接口类 2](#_Toc518338275)

[1.1软件接口说明 2](#_Toc518338276)

[1.2接口方法列表 2](#_Toc518338277)

[2.例子源码 5](#_Toc518338278)

[2.1 select接口 5](#_Toc518338279)

[2.2 insertt接口 6](#_Toc518338280)

[2.3 delete接口 6](#_Toc518338281)

[2.4 update接口 6](#_Toc518338282)

[2.5seleteBlur接口 7](#_Toc518338283)

[2.6seleteOrder接口 7](#_Toc518338284)

|  |  |  |
| --- | --- | --- |
| **版本信息** | **发布者** | **版本改进简要** |
| **1.0/2018/6/26** | **张北贤** |  |
| **2.0/2018/7/2** | **张北贤** | **新增了seleteBlur接口**  **和seleteOrder接口** |

**注：本方法中：selete应写为select，项目中已经改对了，本文档仍然使用selete**

**数据库操作统一接口**

## 软件接口类

### 1.1软件接口说明

该软件接口依赖database.util包的MysqlBaseDao类。该类封装了对数据库的操作，避免了频繁、重复地写操作数据库的代码。重用性高，从而达到减少项目代码量的目的。

调用该类时，会自动进行数据库连接，并根据传入的数据生成数据库查询语句，最终达到期望效果。

### 1.2接口方法列表

|  |  |  |
| --- | --- | --- |
| 接口名称 | 参数及说明 | 返回值及说明 |
| selete (  String tableName,  Set<String> LookupParams,  Map<String,Object>conditionalParams  ) | tableName:  将要查询的目标表的名称 | ResultSet类型，  数据库查询结果集合 |
| LookupParams:  查询的目标字段的集合；为null时表示查找所有字段 |
| conditionalParams:  查询条件的集合，key为表字段，value为实际参数，value暂只支持String和int型；conditionalParams为null时表示无条件查询表 |
| insert(  String tableName,  Map<String,Object> insertParams  ) | tableName:表名 | Int类型  异常返回0  执行成功返回1 |
| insertParams:  需要增加的数据的集合，key为表字段，value为实际参数，value暂只支持String和int型 |
| delete(  String tableName,  Map<String,Object> onditionalParams  ) | tableName:表名 | Int类型  异常返回0  执行成功返回1 |
| conditionalParams:  删除条件的集合，key为表字段，value为实际参数，value暂只支持String和int型；conditionalParams为null时表示删除整张表的数据，保留表 |
| update(  String tableName,  Map<String,Object> updateParams,  Map<String,Object>conditionalParams) | tableName:表名 | Int类型  异常返回0  执行成功返回1 |
| updateParams:  需要更新的参数集合，key为表字段，value为实际参数，value暂只支持String和int型； |
| conditionalParams:更新条件的集合，key为表字段，value为实际参数，value暂只支持String和int型；conditionalParams为null时表示无条件更新表的所有记录相应的字段 |

|  |  |  |
| --- | --- | --- |
| seleteBlur(  String tableName,  Set<String> LookupParams,  Map<String,Object>conditionalParams)  **（注：与selete()方法不同的在于，该方法支持模糊查询，blur及模糊的意思）** | tableName:表名 | ResultSet类型，  数据库查询结果集合 |
| LookupParams:  查询的目标字段的集合；为null时表示查找所有字段 |
| conditionalParams:  查询条件的集合，key为表字段，value为实际参数，value暂只支持String和int型；conditionalParams为null时表示无条件查询表 |
|  |

|  |  |  |
| --- | --- | --- |
| seleteOrder(  String tableName,  Set<String> LookupParams,  Map<String,Object>conditionalParams,  String sortOrder,  Set<String> sortParams  )  **（注：与selete()方法不同的在于，该方法支持模糊查询，blur及模糊的意思）** | tableName:表名 | ResultSet类型，  数据库查询结果集合 |
| LookupParams:  查询的目标字段的集合；为null时表示查找所有字段 |
| conditionalParams:  查询条件的集合，key为表字段，value为实际参数，value暂只支持String和int型；conditionalParams为null时表示无条件查询表 |
| sortOrder:指明升序或降序，"ASC"为升序，"DESC"为降序，为null时不启用排序 |
| sortParams:指明排序所依据的字段的名称的集合，为null时不启用排序 |

## 2.例子源码

### 2.1 select接口

public static void main(String[] args) throws SQLException{

MysqlUtil util=new MysqlUtil();

String tableName="student";

Set<String> LookupParams = new HashSet<>();

LookupParams.add("s\_number");

LookupParams.add("s\_name");

Map<String,Object> conditionalParams=new HashMap<String,Object>();

conditionalParams.put("s\_number", "115583010114");

conditionalParams.put("s\_name", "张北贤");

ResultSet rs=util.selete(tableName,null,conditionalParams);

if(rs.next()){

//System.out.println(rs.getString("s\_number"));

//System.out.println(rs.getString("s\_name"));

}

}

### 2.2 insertt接口

public static void main(String[] args) throws SQLException{

MysqlUtil util=new MysqlUtil();

String tableName="student";

Map<String,Object> insertParams=new HashMap<String,Object>();

insertParams.put("s\_number", "115583010114");

insertParams.put("s\_name", "张北贤");

insertParams.put("password", "123456");

util.insert(tableName, insertParams);

}

### 2.3 delete接口

public static void main(String[] args) throws SQLException{

MysqlUtil util=new MysqlUtil();

String tableName="student";

Map<String,Object> conditionalParams=new HashMap<String,Object>();

String str="zbx";

conditionalParams.put("s\_name", "zbx");

conditionalParams.put("s\_number", "115583010114");

util.delete(tableName, conditionalParams);

}

### 2.4 update接口

public static void main(String[] args) throws SQLException{

MysqlUtil util=new MysqlUtil();

String tableName6="student";

Map<String,Object> conditionalParams=new HashMap<String,Object>();

conditionalParams.put("s\_number", "115583010114");

Map<String,Object> updateParams=new HashMap<String,Object>();

updateParams.put("password", "123456");

updateParams.put("s\_name", "张怡宝");

util.update(tableName6, updateParams, conditionalParams);

}

### 2.5seleteBlur接口

String tableName="abstract\_task";

Set<String> LookupParams = new HashSet<>();

LookupParams.add("task\_number");

LookupParams.add("task\_title");

Map<String,Object> conditionalParams5=new HashMap<String,Object>();

conditionalParams5.put("task\_title", "作业");

ResultSet rs=util5.seleteBlur(tableName, LookupParams, conditionalParams5);

while(rs.next()){

System.out.println(rs.getString("task\_number"));

System.out.println(rs.getString("task\_title"));

}

### 2.6seleteOrder接口

String tableName="online\_task";

Set<String> LookupParams = new HashSet<>();

LookupParams.add("task\_number");

LookupParams.add("question\_type");

Map<String,Object> conditionalParams=new HashMap<String,Object>();

conditionalParams.put("task\_number", "002");

Set<String> sortParams = new HashSet<>();

sortParams.add("question\_type");

sortParams.add("task\_number");

ResultSet rs=

util5.seleteOrder(tableName, LookupParams, conditionalParams, ConstantString.DESC, sortParams);

while(rs.next()){

System.out.println(rs.getString("task\_number")+" "+rs.getString("question\_type"));

}