数据库公共操作类封装

**1 封装类**

|  |  |  |
| --- | --- | --- |
| 序号 | 类名 | 说明 |
| 1 | SQLString | SQL语句构造 |
| 2 | DataTable | 存储从数据库中获取到的数据 |
| 3 | DBHelper | 数据库访问操作 |

**2 SQLString类封装**

此类封装了一个构造SQL语句的类，首先传入一个SQL语句（可以一次传入完整的SQL语句，也可以传入部分SQL语句，传入部分SQL方法需要在后续步骤中一次或多次继续传入剩余SQL语句，append方法），其中SQL语句中等号后面的值可用“？”代替。再采用set方法按顺序传入等号后面的值以替换“？”，最后生成一条完整的SQL语句。

|  |  |  |
| --- | --- | --- |
| 序号 | 方法名 | 说明 |
| 1 | SQLString | 构造函数，传入一条完整的SQL语句或部分SQL语句 |
| 2 | append | 添加剩余SQL |
| 3 | setInt | 设置Int类型数据 |
| 4 | setFloat | 设置Float类型数据 |
| 5 | setDouble | 设置Double类型数据 |
| 6 | setString | 设置String类型数据 |
| 7 | setDate | 设置时间类型数据 |
| 8 | toString | 生成完整的SQL语句 |

/\*\*

\* 构造函数

\* **@param** sql 一条完整的SQL语句或部分SQL语句

\*/

**public** SQLString(String sql);

/\*\*

\* 添加剩余SQL

\* **@param** s 剩余的SQL语句

\*/

**public** **void** append(String s);

/\*\*

\* 设置Int类型数据

\* **@param** i 第几个“？” (从1开始)

\* **@param** value “？”位置的值

\*/

**public** **void** setInt(**int** i, **int** value);

/\*\*

\* 设置Float类型数据

\* **@param** i 第几个“？” (从1开始)

\* **@param** value “？”位置的值

\*/

**public** **void** setFloat(**int** i, **float** value);

/\*\*

\* 设置Double类型数据

\* **@param** i 第几个“？” (从1开始)

\* **@param** value “？”位置的值

\*/

**public** **void** setDouble(**int** i, **double** value);

/\*\*

\* 设置String类型数据

\* **@param** i 第几个“？” (从1开始)

\* **@param** value “？”位置的值

\*/

**public** **void** setString(**int** i, String value);

/\*\*

\* 设置时间类型数据

\* **@param** i 第几个“？” (从1开始)

\* **@param** value “？”位置的值

\*/

**public** **void** setDate(**int** i, Date value);

/\*\*

\* 生成完整的SQL语句

\* **@return** 完整的SQL语句

\*/

@Override

**public** String toString();

**3 DataTable类封装**

此类封装了一个按照表的结构存储从数据库中读取的数据，存储数据的行、列顺序和SQL读取出来的数据顺序一致。

|  |  |  |
| --- | --- | --- |
| 序号 | 方法名 | 说明 |
| 1 | DataTable | 构造函数，构造数据结果集 |
| 2 | getString | 获取String数据 |
| 3 | getInteger | 获取Integer类型数据 |
| 4 | getLong | 获取Long类型数据 |
| 5 | getFloat | 获取Float类型数据 |
| 6 | getDouble | 获取Double类型数据 |
| 7 | getDate | 获取Date类型数据 |
| 8 | rows | 获取行数 |
| 9 | cols | 获取列数 |

/\*\*

\* 构造函数

\* **@param** rs 数据结果集

\* **@throws** SQLException

\*/

**public** DataTable(ResultSet rs) **throws** SQLException;

/\*\*

\* 获取String数据

\* **@param** row 行下标

\* **@param** key 字段名

\* **@return** String类型数据

\*/

**public** String getString(**int** row,String key);

/\*\*

\* 获取String数据

\* **@param** row 行下标

\* **@param** index 列下标

\* **@return** String类型数据

\*/

**public** String getString(**int** row,**int** index);

/\*\*

\* 获取Integer类型数据

\* **@param** row 行下标

\* **@param** key 字段名

\* **@return** Integer类型数据

\*/

**public** Integer getInteger(**int** row,String key);

/\*\*

\* 获取Integer类型数据

\* **@param** row 行下标

\* **@param** index 列下标

\* **@return** Integer类型数据

\*/

**public** Integer getInteger(**int** row,**int** index);

/\*\*

\* 获取Long类型数据

\* **@param** row 行下标

\* **@param** key 字段名

\* **@return** Long类型数据

\*/

**public** Long getLong(**int** row, String key );

/\*\*

\* 获取Long类型数据

\* **@param** row 行下标

\* **@param** index 列下标

\* **@return** Long类型数据

\*/

**public** Long getLong(**int** row, **int** index);

/\*\*

\* 获取Float类型数据

\* **@param** row 行下标

\* **@param** key 字段名

\* **@return** Float数据类型

\*/

**public** Float getFloat(**int** row, String key);

/\*\*

\* 获取Float类型数据

\* **@param** row 行下标

\* **@param** index 列下标

\* **@return** Float类型数据

\*/

**public** Float getFloat(**int** row, **int** index);

/\*\*

\* 获取Double类型数据

\* **@param** row 行下标

\* **@param** index 列下标

\* **@return** Double类型数据

\*/

**public** Double getDouble(**int** row, **int** index);

/\*\*

\* 获取Double类型数据

\* **@param** row 行下标

\* **@param** key 字段名

\* **@return** Double类型数据

\*/

**public** Double getDouble(**int** row, String key);

/\*\*

\* 获取Date类型数据

\* **@param** row 行下标

\* **@param** key 字段名

\* **@return** Date类型数据

\*/

**public** Date getDate(**int** row, String key);

/\*\*

\* 获取Date类型数据

\* **@param** row 行下标

\* **@param** index 列下标

\* **@return** Date类型数据

\*/

**public** Date getDate(**int** row, **int** index );

/\*\*

\* 获取行数

\* **@return** 行数

\*/

**public** Integer rows();

/\*\*

\* 获取列数

\* **@return** 列数

\*/

**public** Integer cols();

**4 DBHelper类封装（数据库操作类）**

本类封装了数据库的增删查改操作，需要传入完整的SQL语句。本类默认参数：用户“root”，无密码，MYSQL驱动，以及MYSQL本地地址。若不用默认参数，可以继承此类，重载getDUPU方法。

|  |  |  |
| --- | --- | --- |
| 序号 | 方法名 | 说明 |
| 1 | query | 执行查询语句 |
| 2 | insertOrUpdate | 执行插入语句、更新语句、删除语句 |
| 3 | getDUPU | 获取驱动，账号，密码，地址；与默认参数不一致时重载此方法 |

/\*\*

\* 执行查询语句

\* **@param** sql 查询SQL语句

\* **@return** 查询数据集

\* **@throws** SQLException

\*/

**public** DataTable query(String sql) **throws** SQLException;

/\*\*

\* 执行插入语句、更新语句、删除语句

\* **@param** sql 修改的SQL语句

\* **@return** SQL语句执行时影响的条数

\* **@throws** SQLException

\*/

**public** **int** insertOrUpdate(String sql) **throws** SQLException;

/\*\*

\* 获取驱动，账号，密码，地址

\* 与默认参数不一致时重载此方法

\* **@return** [0]:驱动字符串，[1]:账号，[2]:密码，[3]:链接地址URL

\*/

**protected** String[] getDUPU();