**访问数据库**

* 数据库连接被用于向数据库服务器发送命令和SQL语句，在连接建立后，需要对数据库进行访问，执行sql语句
* 在java.sql包中有3个接口分别定义了对数据库的调用的不同方式：
  + Statement
  + PreparedStatement
  + CallableStatement

**Statement**

* 通过调用Connection对象的createStatement方法创建该对象。
* 该对象用于执行静态的SQL语句，并且返回执行结果。
* Statement接口中定义了下列方法用于执行SQL语句：
  + ResultSet excuteQuery(String sql)
  + Int excuteUpdate(String sql)

*/\*\*  
 \* 功能描述: <br>通用的更新方法：包括insert、update、delete  
 \* 版本1.  
 \* 〈〉  
 \*  
 \** ***@Param:*** *[sql]  
 \** ***@Return:*** *void  
 \** ***@Author:*** *nino  
 \** ***@Date:*** *2019/2/28 11:24  
 \*/***public void** update(String sql) {  
 Test01 test01 = **new** Test01();  
 Connection conn = **null**;  
 Statement statement = **null**;  
 **try** {  
 conn = JDBCUtil.*getConnection*();  
 statement = conn.createStatement();  
 statement.executeUpdate(sql);  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 } **finally** {  
 JDBCUtil.*releaseSource*(statement, conn);  
 }  
}

**public class** JDBCUtil {  
  
 */\*\*  
 \* 功能描述:关闭Statement和Connection  
 \*  
 \** ***@Param:*** *[statement, connection]  
 \** ***@Return:*** *void  
 \** ***@Author:*** *nino  
 \** ***@Date:*** *2019/2/28 11:42  
 \*/* **public static void** releaseSource(Statement statement, Connection conn) {  
 **if** (statement != **null**) {  
 **try** {  
 statement.close();  
 } **catch** (SQLException e) {  
 e.printStackTrace();  
 }  
 }  
 **if** (conn != **null**) {  
 **try** {  
 conn.close();  
 } **catch** (SQLException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
  
 */\*\*  
 \* 功能描述:  
 \* 1.获取连接的方法  
 \* 通过读取配置文件从数据库服务器获取一个连接  
 \*  
 \** ***@Param:*** *[]  
 \** ***@Return:*** *java.sql.Connection  
 \** ***@Author:*** *nino  
 \** ***@Date:*** *2019/2/28 11:38  
 \*/* **public static** Connection getConnection() **throws** Exception {  
 *// 1.准备连接数据库的4个字符串  
 // 1）创建properties对象* Properties properties = **new** Properties();  
  
 *// 2）获取jdbc.properties 对应的输入流* InputStream in = JDBCUtil.**class**.getClassLoader().getResourceAsStream(**"jdbc.properties"**);  
  
 *// 3）加载2）对应的输入流* properties.load(in);  
  
 *// 4）具体决定4个字符串* String driver = properties.getProperty(**"driver"**);  
 String jdbcUrl = properties.getProperty(**"jdbcUrl"**);  
 String user = properties.getProperty(**"user"**);  
 String password = properties.getProperty(**"password"**);  
  
 *// 2.加载数据库数据库驱动* Class.*forName*(driver);  
 *// 3.通过DriverManager的getConnection()方法获取数据库连接* **return** DriverManager.*getConnection*(jdbcUrl, user, password);  
 }  
}