JDBC

是Java语音操作关系型数据库的一套API，也就是一个接口。各个数据库产商去实现JDBC编写操作对应数据库的类也就是驱动。

官方定义了一套操作所以关系型数据库的规则，也就是接口。各个数据库产商去实现这套接口，提供数据库的驱动jar包，我们可以使用这套JDBC接口编程，实际上执行代码的是驱动jar包中的实现类。

使用JDBC连接数据库的步骤：

1. 引入对应的数据库的jar包
2. 注册驱动（可以省略）
3. 获取连接对象Connection (DriverManager.getConnection(url,username,password))
4. 获取执行SQL语句的对象Statement (connection.createStatement())
5. 编写SQL语句
6. 执行对应的SQL语句
7. 处理返回结果
8. 释放资源

url固定格式:jdbc:mysql://127.0.0.1:端口号/数据库名称？键值对

（如果是本机且端口号是3306则可以省略jdbc:mysql:///数据库名称？键值对

）

DriverManager:

DriverManager作用：

1. 注册驱动
2. 获取数据库连接对象

Connection:

(1)

1. 获取普通执行SQL语句的对象

（Statement createStatement()）

1. 预编译SQL的SQL执行对象：防止SQL注入

(PreparedStatement preparedStatement(sql))

1. 事务管理：
2. 开启事务：setAutoCommit(boolean autoCommit) 默认是true自动提交事务，false开启手动提交。
3. 提交事务：commit()
4. 回滚事务：rollback()

Statement:

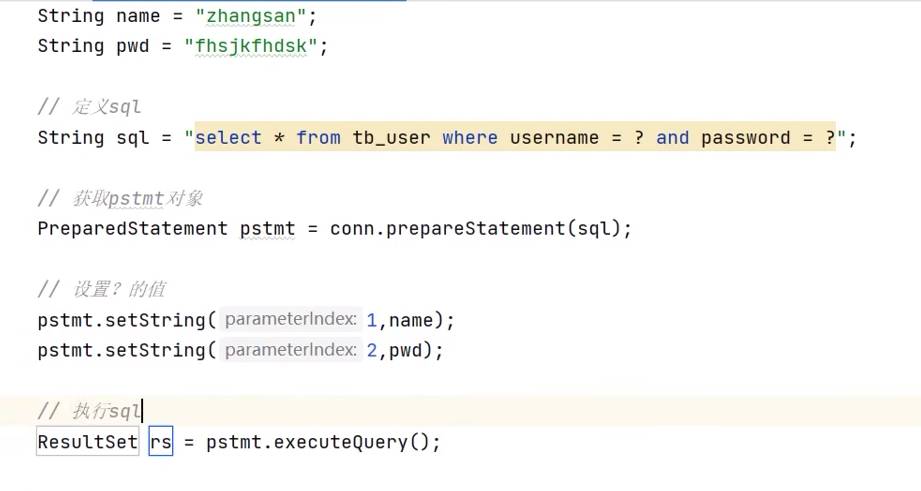
作用：执行SQL语句

1.Int executeUpdate(sql);执行DML,DDL语句

返回值：（1）DML语句影响数据库的行数（2）DDL语句执行后 ，执行成功有可能返回0

1. ResultSet executeQuery(sql);执行DQL语句

返回值封装了对应数据



PreparedStatement:

作用：1.预编译SQL，并执行SQL语句。2.防止SQL注入，会对字符进行转义

（预编译：将对应字段的值用?占位，程序开始后会先将SQL语句传给数据库进行预编译由于字段值已被?占位所以SQL语句是一样，传给数据库只会预编译一次，提高了效率）

数据库

数据库连接池：

·数据库连接池是一个容器，负责分配，和管理数据库连接

·存储着多个Connection对象，允许应用程序重复使用同一个connection对象而不是重新建立一个

·释放空闲时间超过最大空闲时间的数据库连接来避免因为没有释放数据库连接而引起的数据库连接遗漏。

好处：

·资源重利用

·提升系统响应速度

·避免数据库连接遗漏

数据库连接池Druid使用步骤：

1. 导入对应的jar包
2. 定义配置文件连接本地MySQL
3. 在类中加载配置文件

（Properties prop = new Properties( );

prop.load(new FileInputStream(“配置文件的相对路径”))）

1. 获取数据库连接池对象

（DataSource dataSource = DruidDataSourceFactory.createDataSource(prop)）

1. 获取连接

（Connection conn = dataSource.getConnection( );）