简介：

1. Driver接口实现类：
   1. 简介：

1、由驱动程序管理器类(java.sql.DriverManager)去调用 这些Driver实现。

* 1. 加载驱动：

1、加载 JDBC 驱动需调用 Class 类的静态方法 forName()，向其传递要加载的 JDBC 驱动的类名，

2、Class.forName(“com.mysql.jdbc.Driver”);。

* 1. 注册驱动：

1、DriverManager 类是驱动程序管理器类，负责管理驱动程序。

使用DriverManager.registerDriver(com.mysql.jdbc.Driver)来注册驱动。

2、通常不用显式调用 DriverManager 类的 registerDriver() 方法来注册驱动程序类的实例。

1. URL
   1. 简介：

1、JDBC URL 用于标识一个被注册的驱动程序，驱动程序管理器

通过这个 URL 选择正确的驱动程序，从而建立到数据库的连接。

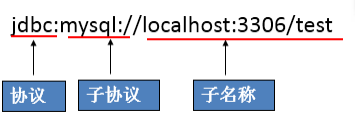
b）组成：

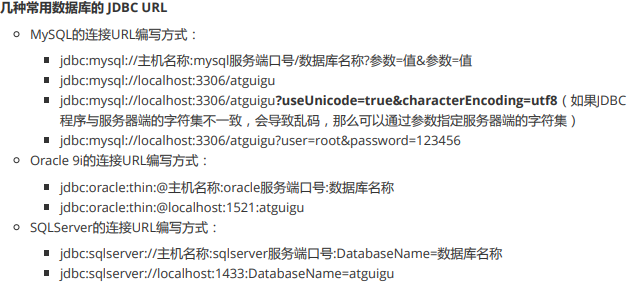
1、jdbc:子协议:子名称

2、协议：JDBC URL中的协议总是jdbc

3、子协议：子协议用于标识一个数据库驱动程序

4、子名称：一种标识数据库的方法。子名称可以依不同的子协议而变化，用子名称的目的是为了定位数据库 提供足够的信息。包含**主机名**(对应服务端的ip地址)，**端口号**，**数据库名**





1. 用户名&密码：
   1. user,password可以用“属性名=属性值”方式告诉数据库
   2. 可以调用 DriverManager 类的 getConnection() 方法建立到数据库的连接
2. 连接方式一：

显示的出现第三方数据库API，并且直接显示注册驱动



1. 连接方式二：

增加使用反射，实例化Driver，使得移植性更好



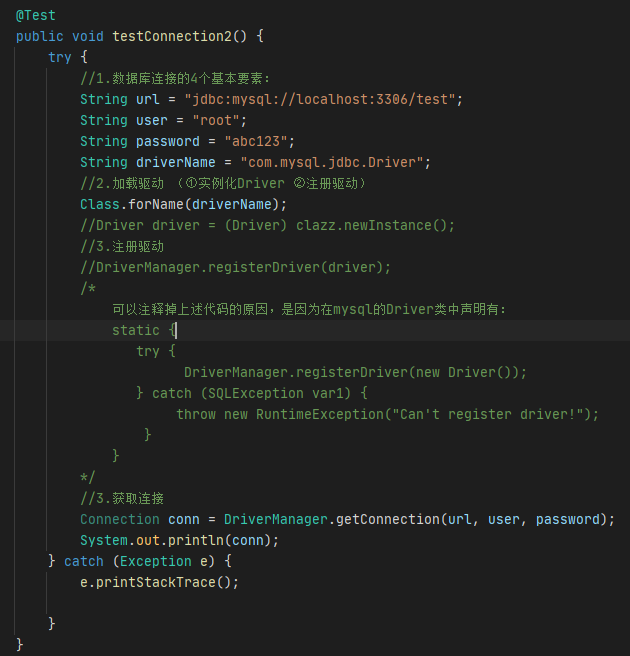
1. 连接方式三：

使用DriverManager替换Driver，在获取链接前，需要先调用registerDriver（）方法，注册驱动



1. 连接方式四：

省略registerDriver（）注册驱动步骤，因为在Driver实现类中的静态代码块中已经进行了注册驱动



1. 连接方式五：

使用配置文件加载配置信息，实现解耦

如果需要修改配置信息，不需要重新打包

