C3P0连接池：

连接池用来共享连接Connection

用池来管理Connection

优化获得连接和释放资源这种非常

Java为数据库连接提供了公共的接口（规范）：javax.sql.DateSource.让自己连接池实现这个接口

自己写连接池：

1. 创建连接池

private static LinkedList<Connection> pool = new LinkedList<Connection>();

常见的连接池DBCP，C3P0

1.继承

2.装饰者计模式增加方法

固定格式：接口A，已知实现类C，需要装饰者创建代理类B

1. 创建B类，实现接口A
2. 提供类B的构造方法，参数类型为A，用于接收A接口的其他实现类C
3. 给类B添加类型为A的成员变量，用于存放A接口的其他实现类
4. 增强需要的方法
5. 实现不需要增强的方法

3.动态代理

* 第一步：引入C3P0连接池的jar包.
* 第二步：编写代码：

1. 通过对象ComboPooledDataSource cpds = new ComboPooledDataSource();调用方法
2. ComboPooledDataSource cpds = new ComboPooledDataSource();会进行判断，判断是否有配置文件如果没有就执行后面的代码，如果有则使用配置文件

* \* 手动设置参数:
* \* 配置文件设置参数：

1. 配置文件名称：c3p0-config.xml（名字是固定的）
2. 配置文件路径：src下面（类路径）

DBUtils;

* DBUtils是java编程中的数据库操作实用工具,小巧简单实用。DBUtils封装了对JDBC的操作,简化了JDBC操作,可以少写代码
* Dbutils三个核心功能介绍
  + QueryRunner中提供对sql语句操作的API，提供连接池底层会自动连接维护Connection
  + ResultSetHandler接口,用于定义select操作后,怎样封装结果集.
  + DbUtils类,它就是一个工具类,定义了关闭资源与事务处理的方法
* 我们知道在执行select语句之后得到的是ResultSet，然后我们还需要对ResultSet进行转换，得到最终我们想要的数据。你可以希望把ResultSet的数据放到一个List中，也可能想把数据放到一个Map中，或是一个Bean中。
* DBUtils提供了一个接口ResultSetHandler，它就是用来ResultSet转换成目标类型的工具。你可以自己去实现这个接口，把ResultSet转换成你想要的类型。
* DBUtils提供了很多个ResultSetHandler接口的实现，这些实现已经基本够用了，我们通常不用自己去实现ResultSet接口了。

QueryRunner核心类

* QueryRunner核心类
  + QueryRunner(DataSource ds),
    - 提供数据源(连接池), DBUtils底层自动维护连接connection
  + update(String sql, Object... params) ,
    - 执行更新数据
  + query(String sql, ResultSetHandler<T> rsh, Object... params) ,
    - 执行查询

ResultSetHandler结果集处理类.

BeanHandler：

单行处理器！把结果集转换成Bean，该处理器需要Class参数，即Bean的类型；

BeanListHandler：

多行处理器！把结果集转换成List<Bean>；

MapListHandler:

字段作为key值作为values存储在map集合里面

ColumnListHandler:

可以传参数，默认访问第一列，可以传参数的传的是列名

ScalarHandler：

单行单列处理器！把结果集转换成Object。一般用于聚集查询，

例如select count(\*) from tab\_student。