conf.properties

```
driverClass=com.mysql.jdbc.Driver
url=jdbc:mysql://localhost:3306/mydb1
username=root
password=root
```

JDBCUtils

```
package cn.tedu.utils;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Properties;

//JDBC创建连接和关闭资源的工具类
public class JDBCUtils {
    //配置文件信息只需要加载一次，所以设置为静态对象
    public static Properties prop = new Properties();
    static{
        try {
            prop.load(
            new FileInputStream(
            new File(
            //通过当前类调用类加载器，类加载器启动的时候可以读取src目录下的配置文件。
            //getClassLoader()获取类加载器
            //getResource()获取src目录
            //getPath()目的是得到一个String类型的文件路径给new File()使用
            JDBCUtils.class.getClassLoader()
            .getResource("conf.properties").getPath())));
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
    //工具类不能共创建对象，所以把构造函数私有化
    private JDBCUtils(){

    }
    //创建连接
    public static Connection getConnection() throws Exception{
        Class.forName(prop.getProperty("driverClass"));
        return DriverManager.getConnection(
                prop.getProperty("url"),
                prop.getProperty("username"),
                prop.getProperty("password"));
```

```
    }
    //关闭资源
    public static void close(Connection conn,Statement stat,ResultSet rs){
        if(rs != null){
            try {
                rs.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }finally{
                rs = null;
            }
        }
        if(stat != null){
            try {
                stat.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }finally{
                stat = null;
            }
        }
        if(conn != null){
            try {
                conn.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }finally{
                conn = null;
            }
        }
    }

}
```

Login

```
package cn.tedu.jdbc;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.Scanner;

import cn.tedu.utils.JDBCUtils;

//模拟登录功能
public class Login {
    public static void main(String[] args) {
        System.out.println("请输入用户名：");
        Scanner sc = new Scanner(System.in);
        String username = sc.nextLine();
        System.out.println("请输入密码：");
        String password = sc.nextLine();
//      testLogin(username,password);
        testPreLogin(username,password);
```

```
    }

    //preparedstatement访问数据库  防止sql注入攻击
    private static void testPreLogin(String username, String password) {
        Connection conn = null;
        PreparedStatement ps = null;
        ResultSet rs = null;
        try {
            conn = JDBCUtils.getConnection();
            ps = conn.prepareStatement("select * from user where username=? and
password = ?");
            ps.setString(1, username);
            ps.setString(2, password);
            rs = ps.executeQuery();
            if(rs.next()){
                System.out.println("登录成功");
            }else{
                System.out.println("登录失败");
            }
        } catch (Exception e) {
            e.printStackTrace();
        }finally{
            JDBCUtils.close(conn, ps, rs);
        }

    }

    private static void testLogin(String username, String password) {
        Connection conn = null;
        Statement stat = null;
        ResultSet rs = null;
        try {
            conn = JDBCUtils.getConnection();
            stat = conn.createStatement();
            rs = stat.executeQuery(
        "select * from user where username = '"+username+"' and password =
'"+password+"'");
            if(rs.next()){
                System.out.println("登录成功");
            }else{
                System.out.println("登录失败");
            }
        } catch (Exception e) {
            e.printStackTrace();
        }finally{
            JDBCUtils.close(conn, stat, rs);
        }

    }

}
```

执行多条sql的操作

PreparedStatement

```
```

```java
package cn.tedu.batch;

import java.sql.Connection;
import java.sql.PreparedStatement;

import cn.tedu.utils.JDBCUtils;

/*PreparedStatement批处理机制:
 优点:
    1.有预编译功能
    2.sql语句预留在数据库服务器中，无需每次都发送sql主干
    3.由于只发送sql参数，所以sql语句的执行效率较高。
 缺点:
    1.不可以书写不同语义的sql
 *
 * */

public class PreparedBatchDemo1 {
    public static void main(String[] args) {
        Connection conn = null;
        PreparedStatement ps = null;
        try {
            conn = JDBCUtils.getConnection();
            ps = conn.prepareStatement("insert into t1 values(?,?)");
            for(int i=0;i<100000;i++){
                //设置每一个批处理的参数
                ps.setInt(1, i);
                ps.setString(2, "name"+i);
                //将数据添加到批处理中
                ps.addBatch();
                if(i%1000 == 0){
                    //每一千条数据执行一次批处理
                    ps.executeBatch();
                    //每一千条清空一次批处理。
                    ps.clearBatch();
                    System.out.println("第"+i/1000+"次处理完毕");
                }
            }
            //数据没有达到最后1000条的数据通过此句执行。
            ps.executeBatch();
        } catch (Exception e) {
            e.printStackTrace();
        }finally{
            JDBCUtils.close(conn, ps, null);
        }
    }

}
```

Statement

```java
package cn.tedu.batch;

import java.sql.Connection;
import java.sql.Statement;
```

```java
import cn.tedu.utils.JDBCUtils;

//Statement批处理机制：
/* 优点：
 *  1.可以书写不同语义的sql。
 * 缺点：
 *  1.没有预编译功能
 *  2.每次sql语句的全部内容都需要传输。
 *  3.无法将主干预留在服务器中，只传输参数，所以效率较低。
 *
 * */
/*
 create table t1(id int,name varchar(20))
 insert into t1 values(1,'aaa')
 insert into t1 values(2,'bbb')
 insert into t1 values(3,'ccc')
 insert into t1 values(4,'ddd')
 * */
public class StateBatchDemo1 {
    public static void main(String[] args) {
        Connection conn = null;
        Statement stat = null;
        try {
            conn = JDBCUtils.getConnection();
            stat = conn.createStatement();
            //通过传输器添加批处理语句
            stat.addBatch("create table t1(id int,name varchar(20))");
            stat.addBatch(" insert into t1 values(1,'aaa')");
            stat.addBatch(" insert into t1 values(2,'bbb')");
            stat.addBatch(" insert into t1 values(3,'ccc')");
            stat.addBatch(" insert into t1 values(4,'ddd')");
            //通知数据库服务器执行批处理
            stat.executeBatch();
            System.out.println("批处理执行完毕");
        } catch (Exception e) {
            e.printStackTrace();
        }finally{
            JDBCUtils.close(conn, stat, null);
        }

    }
}
```

连接池c3p0

c3p0.properties

```
c3p0.driverClass=com.mysql.jdbc.Driver
c3p0.jdbcUrl=jdbc:mysql://localhost:3306/mydb1
c3p0.user=root
c3p0.password=root
```

xc3p0-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<c3p0-config>
  <default-config>
    <property name="driverClass">com.mysql.jdbc.Driver</property>
    <property name="jdbcUrl">jdbc:mysql://localhost:3306/mydb1</property>
    <property name="user">root</property>
    <property name="password">root</property>

  </default-config>


</c3p0-config>
```

C3P0Demo1

```
package cn.tedu.pool;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

import org.apache.commons.dbcp.BasicDataSource;

import com.mchange.v2.c3p0.ComboPooledDataSource;

public class C3P0Demo1 {
    public static void main(String[] args) {
        Connection conn = null;
        PreparedStatement ps = null;
        ResultSet rs = null;
        ComboPooledDataSource source = new ComboPooledDataSource();

        try {
            conn = source.getConnection();
            ps = conn.prepareStatement("select * from exam");
            rs = ps.executeQuery();
            while(rs.next()){
                String name = rs.getString("name");
                System.out.println("name:"+name);
            }

        } catch (SQLException e) {
            e.printStackTrace();
        }finally{
            if(rs != null){
                try {
                    rs.close();
                } catch (SQLException e) {
                    e.printStackTrace();
                }finally{
                    rs = null;
```

```
                }
            }
            if(ps != null){
                try {
                    ps.close();
                } catch (SQLException e) {
                    e.printStackTrace();
                }finally{
                    ps = null;
                }
            }
            if(conn != null){
                try {
                    //归还连接
                    conn.close();
                } catch (SQLException e) {
                    e.printStackTrace();
                }finally{
                    conn = null;
                }
            }
        }

    }

}
```

连接池：dbcp

dbcp.properties

```
driverClassName=com.mysql.jdbc.Driver
url=jdbc:mysql://localhost:3306/mydb1
username=root
password=root
```

DBCPDemo1

```
package cn.tedu.pool;

import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Properties;

import javax.sql.DataSource;

import org.apache.commons.dbcp.BasicDataSource;
```

```java
//DBCP连接池
public class DBCPDemo1 {
    @SuppressWarnings("static-access")
    public static void main(String[] args) throws  Exception {
        Connection conn = null;
        PreparedStatement ps = null;
        ResultSet rs = null;
        BasicDataSource source = new BasicDataSource();
        source.setDriverClassName("com.mysql.jdbc.Driver");
        source.setUrl("jdbc:mysql://localhost:3306/mydb1");
        source.setUsername("root");
        source.setPassword("root");
        //dbcp获取连接
        /*Properties prop = new Properties();
        prop.load(new FileInputStream(new File(
        DBCPDemo1.class.getClassLoader()
        .getResource("dbcp.properties")
        .getPath()
                )));
        BasicDataSourceFactory factory = new BasicDataSourceFactory();
        DataSource source = factory.createDataSource(prop);*/

        try {
            conn = source.getConnection();
            ps = conn.prepareStatement("select * from exam");
            rs = ps.executeQuery();
            while(rs.next()){
                String name = rs.getString("name");
                System.out.println("name:"+name);
            }

        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }finally{
            if(rs != null){
                try {
                    rs.close();
                } catch (SQLException e) {
                    e.printStackTrace();
                }finally{
                    rs = null;
                }
            }
            if(ps != null){
                try {
                    ps.close();
                } catch (SQLException e) {
                    e.printStackTrace();
                }finally{
                    ps = null;
                }
            }
            if(conn != null){
                try {
                    //归还连接
                    conn.close();
                } catch (SQLException e) {
```

```
                    e.printStackTrace();
                }finally{
                    conn = null;
                }
            }
        }
    }

}
```