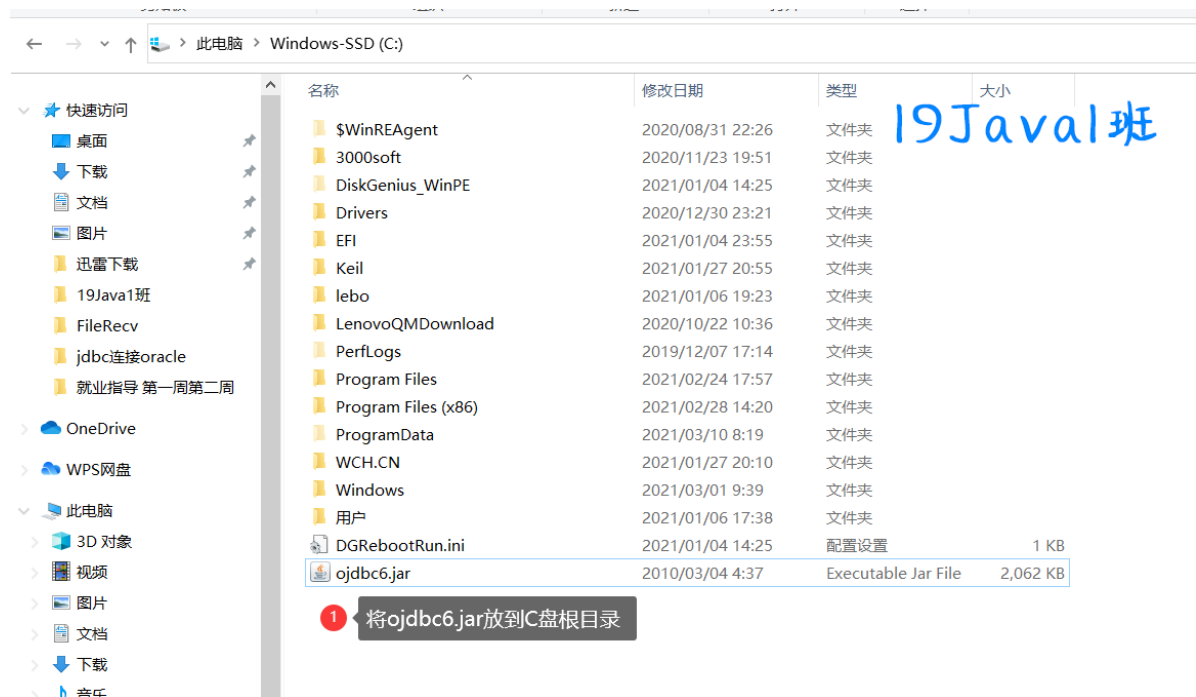


JavaWeb连接Oracle数据库

1. 将ojdbc6.jar复制到C盘根目录



2. 将ojdbc6本地安装到Maven

2.1 复制下面的命令

```
mvn install:install-file -DgroupId=com.oracle -DartifactId=ojdbc6 -Dversion=11.2.0.1.0 -Dpackaging=jar -Dfile=C:\ojdbc6.jar
```

2.2 打开命令行窗口，粘贴并执行



2.3 安装成功

```
C:\Windows\system32\cmd.exe
Microsoft Windows [版本 10.0.19041.804]
(c) 2020 Microsoft Corporation. 保留所有权利。

C:\Users\MaPenghui>mvn install:install-file -DgroupId=com.oracle -DartifactId=ojdbc6 -Dversion=11.2.0.1.0 -Dpackaging=jar -Dfile=C:\ojdbc6.jar
[INFO] Scanning for projects...
[INFO] -----< org.apache.maven:standalone-pom >-----
[INFO] Building Maven Stub Project (No POM) 1
[INFO] -----[ pom ]-----
[INFO] --- maven-install-plugin:2.4:install-file (default-cli) @ standalone-pom ---
[INFO] Installing C:\ojdbc6.jar to E:\apache-maven-3.6.3\repository\com\oracle\ojdbc6\11.2.0.1.0\ojdbc6-11.2.0.1.0.jar
[INFO] Installing C:\Users\MAPENG\1\AppData\Local\Temp\mvninstall16654105823385270310.pom to E:\apache-maven-3.6.3\repository\com\oracle\ojdbc6\11.2.0.1.0\ojdbc6-11.2.0.1.0.pom
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 1.670 s
[INFO] Finished at: 2021-03-11T14:44:26+08:00
[INFO] -----
C:\Users\MaPenghui>
```

如果提示BUILD SUCCESS, 则代表配置成功

3. 配置项目中的pom文件

添加依赖

```
<dependency>
  <groupId>com.oracle</groupId>
  <artifactId>ojdbc6</artifactId>
  <version>11.2.0.1.0</version>
</dependency>
```

```
IndexController.java x README.md x Test.java x pom.xml (yyy) x
16 <properties>
17   <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
18   <maven.compiler.source>1.7</maven.compiler.source>
19   <maven.compiler.target>1.7</maven.compiler.target>
20 </properties>
21
22 <dependencies>
23   <dependency>
24     <groupId>com.oracle</groupId>
25     <artifactId>ojdbc6</artifactId>
26     <version>11.2.0.1.0</version>
27   </dependency>
28   <dependency>
29     <groupId>junit</groupId>
30     <artifactId>junit</artifactId>
31     <version>4.11</version>
32     <scope>test</scope>
33   </dependency>
34   <dependency>
```

1 在pom文件中加入此依赖
然后重载pom.xml文件

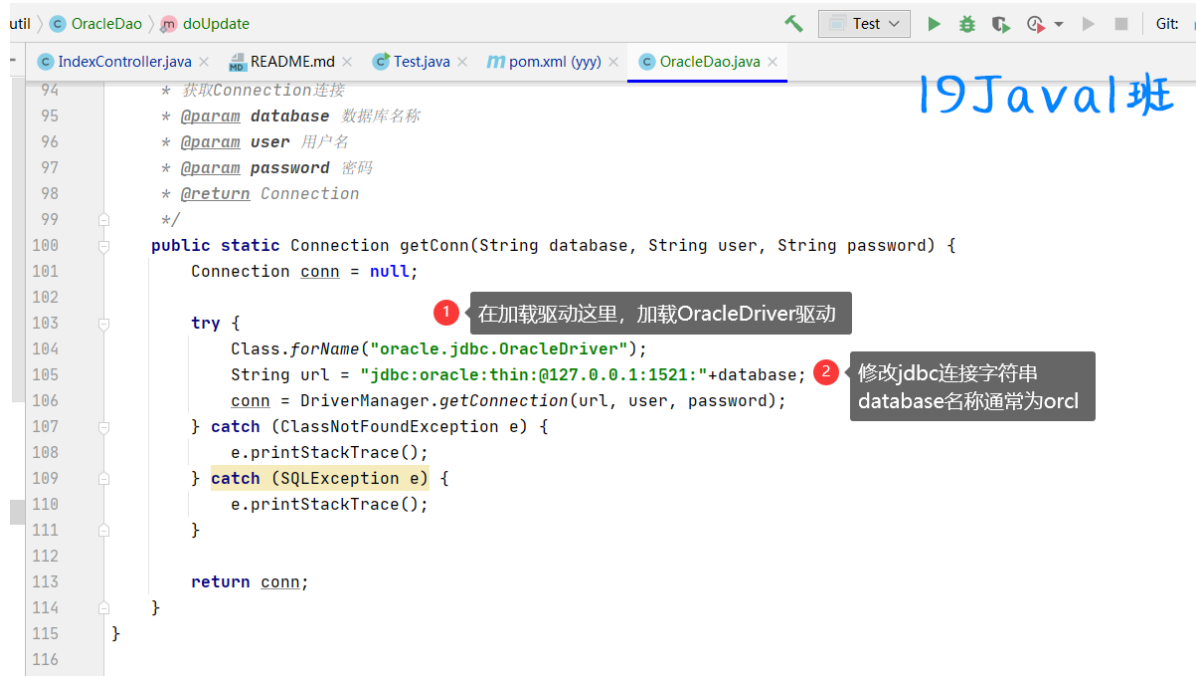
4. 修改BaseDao工具类

4.1 前言

我们此前学习MySQL连接驱动时，已经将数据库常用操作封装到BaseDao.java类中，以便于调用，同样，Oracle数据库也可以进行类似的封装操作。

4.2 修改驱动类文件和连接字符串

```
Class.forName("oracle.jdbc.OracleDriver");
String url = "jdbc:oracle:thin:@127.0.0.1:1521:"+database;
```

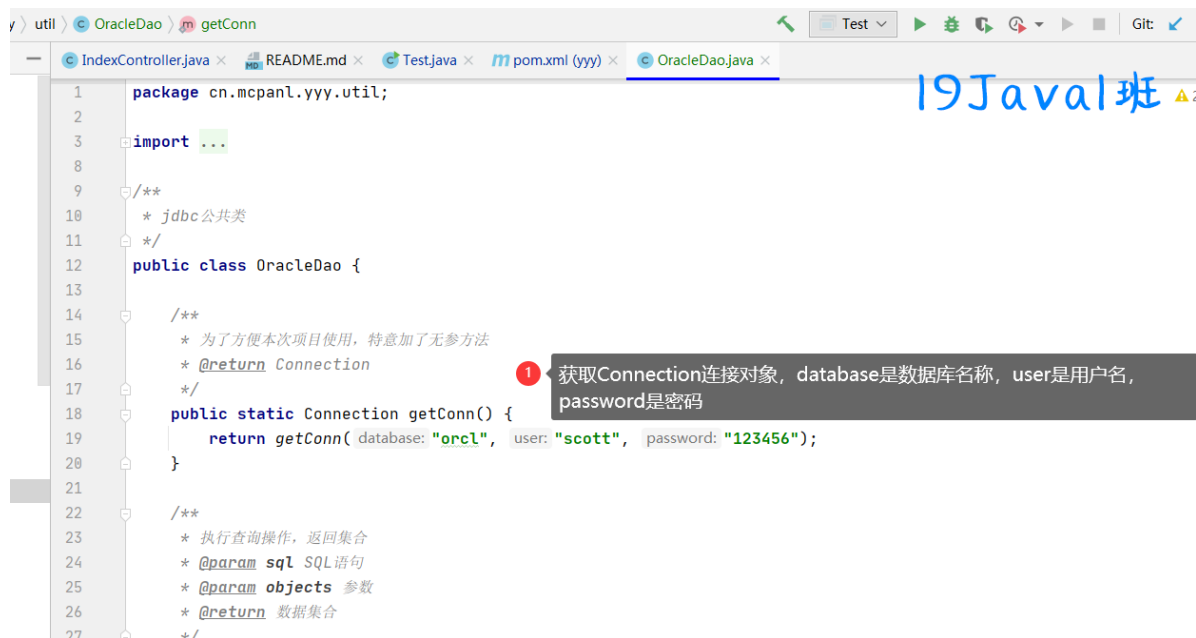


```
94  * 获取Connection连接
95  * @param database 数据库名称
96  * @param user 用户名
97  * @param password 密码
98  * @return Connection
99  */
100 public static Connection getConn(String database, String user, String password) {
101     Connection conn = null;
102
103     try {
104         Class.forName("oracle.jdbc.OracleDriver");
105         String url = "jdbc:oracle:thin:@127.0.0.1:1521:"+database;
106         conn = DriverManager.getConnection(url, user, password);
107     } catch (ClassNotFoundException e) {
108         e.printStackTrace();
109     } catch (SQLException e) {
110         e.printStackTrace();
111     }
112
113     return conn;
114 }
115
116 }
```

19Java1班

1 在加载驱动这里，加载OracleDriver驱动

2 修改jdbc连接字符串
database名称通常为orcl



```
1  package cn.mcpnl.yyy.util;
2
3  import ...
4
5  /**
6   * jdbc公共类
7   */
8  public class OracleDao {
9
10     /**
11      * 为了方便本次项目使用，特意加了无参方法
12      * @return Connection
13      */
14     public static Connection getConn() {
15         return getConn(database: "orcl", user: "scott", password: "123456");
16     }
17
18     /**
19      * 执行查询操作，返回集合
20      * @param sql SQL语句
21      * @param objects 参数
22      * @return 数据集合
23      */
24 }
```

19Java1班

1 获取Connection连接对象，database是数据库名称，user是用户名，password是密码

5. 测试sql语句

```

List<Map<String, Object>> emp_list = OracleDao.doQuery("select * from emp",
null);

for (Map<String, Object> emp : emp_list) {
    system.out.println(emp);
}

```

