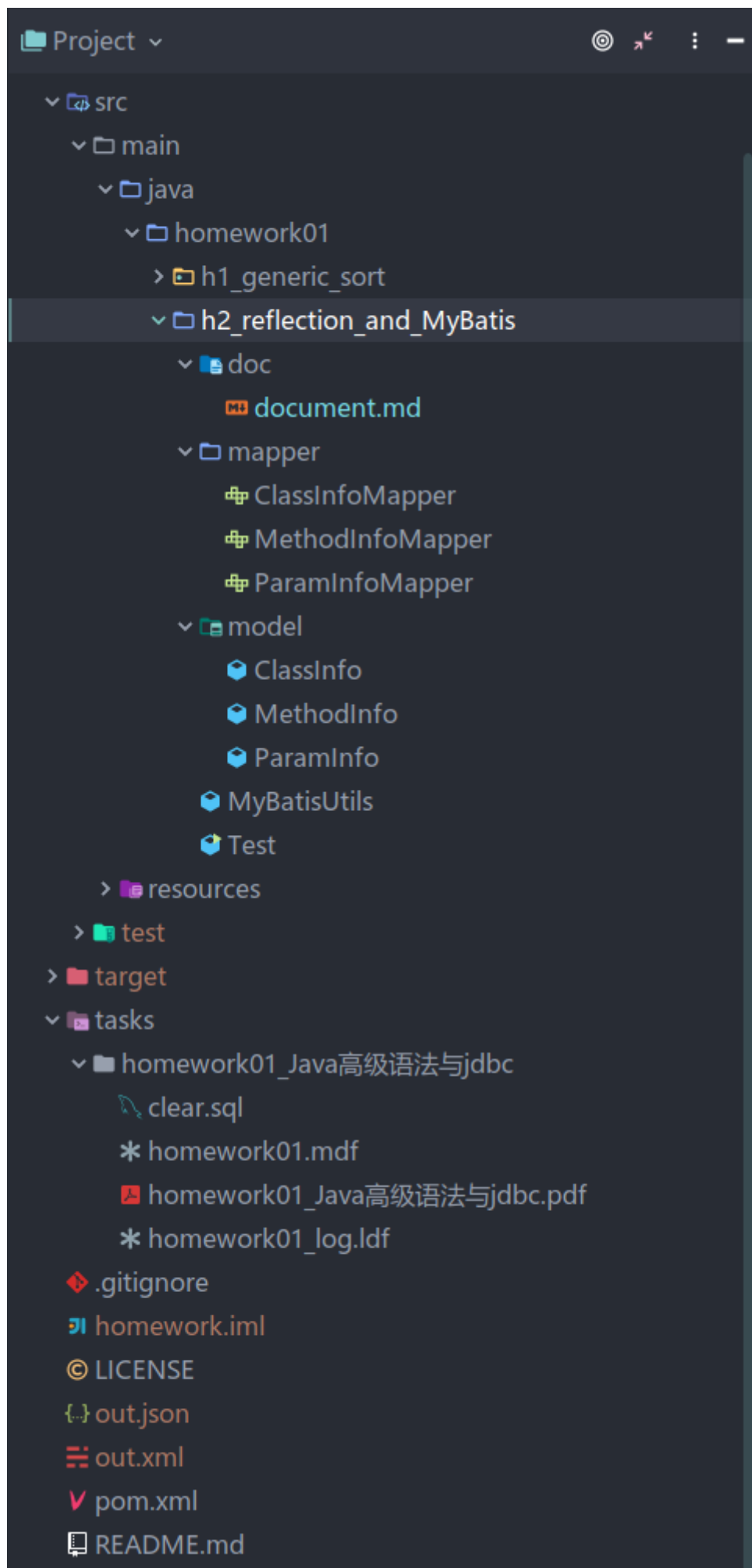


反射与MyBatis

本项目使用 Maven 进行构建，步骤参考 [Maven 入门](#)，其它相关内容均参考课堂 PPT。

项目结构如图所示：



主要 Java 类设计

model 包

model 包有以下 JavaBean，作用是映射数据库中对应的三张表。

- ClassInfo
- MethodInfo
- ParamInfo

mapper 包

mapper 包有以下接口及方法，作用是提供映射接口（参考 PPT），方法名顾名思义，不进行讲解。

- ClassInfoMapper
 - int selectClassIdByClassName(String ClassName)
 - void insertClassName(String className)
 - ClassInfo[] selectAllClassInfo()
- MethodInfoMapper
 - void insertMethodInfo(MethodInfo methodInfo)
 - MethodInfo[] selectAllMethodInfo()
 - int selectMaxMethodId()
- ParamInfoMapper
 - void insertParamInfo(ParamInfo paramInfo)
 - ParamInfo[] selectAllParamInfo()

MyBatisUtils 类

与 PPT 定义的 MyBatisUtils 相同

Test 类

Test 类为包含 main 方法的测试类，主要的逻辑也在其中，实现的功能有：

- 接收用户从控制台输入任意字符串作为类的完整名称，装载成功则打印方法信息到控制台
- 将查询到的类、方法和方法参数信息新增到 ClassInfo、MethodInfo 和 ParamInfo 表。
- 将 3 张表数据导出到一个 json 文件
- 将 3 张表数据导出到一个 xml 文件

Test 类包含的方法如下，方法名顾名思义，不进行讲解：

- private static void showMethods(Class classFromInput)
- private static void addToClassInfo(Class cls)
- private static void addToMethodInfo(int classId, Method method)
- private static void addToParamInfo(int methodId, Parameter[] parameters)
- private static void exportToJSON()
- private static void exportToXML()
- private static void writeFile(StringBuilder content, String fileExtension)

程序输入输出方式

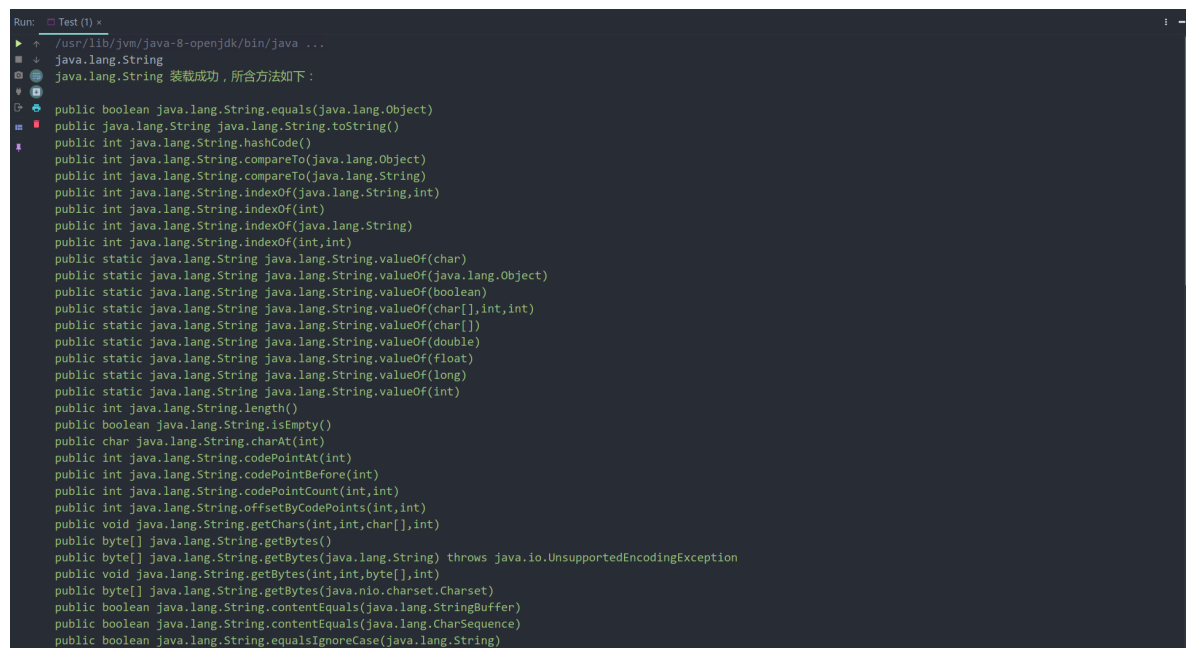
输入

直接输入要处理的 java 类（如：java.lang.String）即可。

输出

- 控制台输出装载成功的类的方法信息
- 输出 3 张表数据到项目目录下的 out.json 文件
- 输出 3 张表数据到项目目录下的 out.xml 文件

输出样例图：



```
Run: Test (1) x
+ /usr/lib/jvm/java-8-openjdk/bin/java ...
+ java.lang.String
+ java.lang.String 装载成功，所含方法如下：
+ public boolean java.lang.String.equals(java.lang.Object)
+ public java.lang.String java.lang.String.toString()
+ public int java.lang.String.hashCode()
+ public int java.lang.String.compareTo(java.lang.Object)
+ public int java.lang.String.compareTo(java.lang.String)
+ public int java.lang.String.indexOf(java.lang.String,int)
+ public int java.lang.String.indexOf(int)
+ public int java.lang.String.indexOf(java.lang.String)
+ public int java.lang.String.indexOf(int,int)
+ public static java.lang.String java.lang.String.valueOf(char)
+ public static java.lang.String java.lang.String.valueOf(java.lang.Object)
+ public static java.lang.String java.lang.String.valueOf(boolean)
+ public static java.lang.String java.lang.String.valueOf(char[],int,int)
+ public static java.lang.String java.lang.String.valueOf(char[])
+ public static java.lang.String java.lang.String.valueOf(double)
+ public static java.lang.String java.lang.String.valueOf(float)
+ public static java.lang.String java.lang.String.valueOf(long)
+ public static java.lang.String java.lang.String.valueOf(int)
+ public int java.lang.String.length()
+ public boolean java.lang.String.isEmpty()
+ public char java.lang.String.charAt(int)
+ public int java.lang.String.codePointBefore(int)
+ public int java.lang.String.codePointCount(int,int)
+ public int java.lang.String.offsetByCodePoints(int,int)
+ public void java.lang.String.getChars(int,int,char[],int)
+ public byte[] java.lang.String.getBytes()
+ public byte[] java.lang.String.getBytes(java.lang.String) throws java.io.UnsupportedEncodingException
+ public void java.lang.String.getBytes(int,int,byte[],int)
+ public byte[] java.lang.String.getBytes(java.nio.charset.Charset)
+ public boolean java.lang.String.contentEquals(java.lang.StringBuffer)
+ public boolean java.lang.String.contentEquals(java.lang.CharSequence)
+ public boolean java.lang.String.equalsIgnoreCase(java.lang.String)
```

```
Run: Test (1) x
public boolean java.lang.String.contains(java.lang.CharSequence)
public java.lang.String java.lang.String.replaceFirst(java.lang.String,java.lang.String)
public java.lang.String java.lang.String.replaceAll(java.lang.String,java.lang.String)
public java.lang.String[] java.lang.String.split(java.lang.String,int)
public java.lang.String[] java.lang.String.split(java.lang.String)
public static java.lang.String java.lang.String.join(java.lang.CharSequence,java.lang.CharSequence[])
public static java.lang.String java.lang.String.join(java.lang.CharSequence,java.lang.Iterable)
public java.lang.String java.lang.String.toLowerCase(java.util.Locale)
public java.lang.String java.lang.String.toLowerCase()
public java.lang.String java.lang.String.toUpperCase()
public java.lang.String java.lang.String.toUpperCase(java.util.Locale)
public java.lang.String java.lang.String.trim()
public char[] java.lang.String.toCharArray()
public static java.lang.String java.lang.String.format(java.util.Locale,java.lang.String,java.lang.Object[])
public static java.lang.String java.lang.String.format(java.lang.String,java.lang.Object[])
public static java.lang.String java.lang.String.copyValueOf(char[],int,int)
public static java.lang.String java.lang.String.copyValueOf(char[])
public native java.lang.String java.lang.String.intern()
public final void java.lang.Object.wait(long,int) throws java.lang.InterruptedException
public final native void java.lang.Object.wait(long) throws java.lang.InterruptedException
public final void java.lang.Object.wait() throws java.lang.InterruptedException
public final native java.lang.Class java.lang.Object.getClass()
public final native void java.lang.Object.notify()
public final native void java.lang.Object.notifyAll()
public default java.util.stream.IntStream java.lang.CharSequence.chars()
public default java.util.stream.IntStream java.lang.CharSequence.codePoints()

正在添加至数据库。..
添加完毕！

导出 json 文件中。..
成功导出至 out.json

导出 xml 文件中。..
成功导出至 out.xml

Process finished with exit code 0
```

输出的 json 文件和 xml 文件（已手动格式化）：

```
1  {}
2  {
3    "classId": 123546,
4    "className": "java.lang.String"
5  },
6  {
7    "methodId": 2642,
8    "classId": 123546,
9    "methodName": "equals",
10   "returnType": "boolean"
11 },
12 {
13   "methodId": 2643,
14   "classId": 123546,
15   "methodName": "toString",
16   "returnType": "class java.lang.String"
17 },
18 {
19   "methodId": 2644,
20   "classId": 123546,
21   "methodName": "hashCode",
22   "returnType": "int"
23 },
24 {
25   "methodId": 2645,
26   "classId": 123546,
27   "methodName": "compareTo",
28   "returnType": "int"
29 },
30 {
31   "methodId": 2646,
32   "classId": 123546,
33   "methodName": "compareTo",
34   "returnType": "int"
35 },
36 {
37   "methodId": 2647,
38   "classId": 123546,
39   "methodName": "indexOf"
```

```
1 <?xml>
2 <homework01.h2_reflection_and_MyBatis.model.ClassInfo-array>
3 <homework01.h2_reflection_and_MyBatis.model.ClassInfo>
4 <classId>123546</classId>
5 <className>java.lang.String</className>
6 </homework01.h2_reflection_and_MyBatis.model.ClassInfo>
7 </homework01.h2_reflection_and_MyBatis.model.ClassInfo-array>
8 <homework01.h2_reflection_and_MyBatis.model.MethodInfo-array>
9 <homework01.h2_reflection_and_MyBatis.model.MethodInfo>
10 <methodId>2642</methodId>
11 <classId>123546</classId>
12 <methodName>equals</methodName>
13 <returnType>boolean</returnType>
14 </homework01.h2_reflection_and_MyBatis.model.MethodInfo>
15 <homework01.h2_reflection_and_MyBatis.model.MethodInfo>
16 <methodId>2643</methodId>
17 <classId>123546</classId>
18 <methodName>toString</methodName>
19 <returnType>class java.lang.String</returnType>
20 </homework01.h2_reflection_and_MyBatis.model.MethodInfo>
21 <homework01.h2_reflection_and_MyBatis.model.MethodInfo>
22 <methodId>2644</methodId>
23 <classId>123546</classId>
24 <methodName>hashCode</methodName>
25 <returnType>int</returnType>
26 </homework01.h2_reflection_and_MyBatis.model.MethodInfo>
27 <homework01.h2_reflection_and_MyBatis.model.MethodInfo>
28 <methodId>2645</methodId>
29 <classId>123546</classId>
30 <methodName>compareTo</methodName>
31 <returnType>int</returnType>
32 </homework01.h2_reflection_and_MyBatis.model.MethodInfo>
33 <homework01.h2_reflection_and_MyBatis.model.MethodInfo>
34 <methodId>2646</methodId>
35 <classId>123546</classId>
36 <methodName>compareTo</methodName>
37 <returnType>int</returnType>
38 </homework01.h2_reflection_and_MyBatis.model.MethodInfo>
39 <homework01.h2_reflection_and_MyBatis.model.MethodInfo>
```

崔书豪