

该文档是关于代码自动测试的说明文档，将说明如何对自动测试程序进行配置。

1. 指定被测试源代码所在包

为了使得测试代码能顺利加载每个同学的 class，第 11-13 章作业通过 Javadoc API 规定了每个类的类名、每个类的数据成员的类型和变量名、每个方法的方法名、形参类型、返回类型必须一致。除此之外，还必须**统一包名**。

- 11-13 章编程题第二题，包名统一为 `package homework.ch11_13.p3`;
- 11-13 章编程题第三题，包名统一为 `package homework.ch11_13.p4`;

在 IDEA 操作演示 3.mp4 里，演示了如果通过 Refactor 改变代码所在的包，是很简单的操作，但是一定要通过 Refactor 改变代码所在的包。如果直接修改包名，相关代码修改起来比较麻烦。

2. 测试代码的目录结构

测试代码的目录结构如图 1 所示：

2023-2024春季PPT > 作业 > 第11-13章作业 (2023-03-24更新) > AutoTest_update3

名称	修改日期	类型	大小
betest	2023/3/25 17:51	文件夹	
lib	2021/2/26 15:14	文件夹	
test	2021/2/26 15:14	文件夹	
test-output	2021/2/26 15:14	文件夹	
test.bat	2023/3/25 18:04	Windows 批处理...	1 KB
test.sh	2023/3/25 17:56	SH 文件	1 KB
testng.xml	2020/3/22 7:19	XML 文档	1 KB
testng2.xml	2020/3/21 6:21	XML 文档	4 KB
testng3.xml	2020/3/22 6:33	XML 文档	3 KB
自动测试Readme.docx	2024/3/28 0:29	Microsoft Word ...	1,245 KB
~\$测试Readme.docx	2024/4/1 0:52	Microsoft Word ...	1 KB

图 1 测试代码的目录结构

测试代码的根目录为 AutoTest_update3。在该目录中，各文件和目录内容说明如下：

- testng.xml，testng2.xml，testng3.xml 为测试的配置信息；
- test.bat 是 Windows 运行测试程序的脚本文件，**大家需要修改**，如何修改见第 3 节；
- test.sh 是 Linux/Mac 运行测试程序的脚本文件，**大家需要修改**，如何修改见第 3 节；
- test-output 目录是测试结果报告的输出目录，**该目录名不要修改，目录也别删除**；
- lib 目录是测试程序运行所依赖的 jar 包，**该目录名不要修改，目录也别删除**；
- betest 目录是最后提交测试代码的目录，学生在本机测试时，可以不使用这个目录。

但提交作业时，需要把作业的 class 文件拷贝到这个目录。具体说明见第 5 节。

请保持 AutoTest_update3 下的目录结构不变，AutoTest_update3 目录可以位于任何位置。

大家进入到该目录，运行 **test.bat** 就可以启动测试程序。如果是在 Linux 下或者 Mac 下面，目录结构和要求一样。

3. test.bat,test.sh,testing.xml 的修改

用于 Windows 的 test.bat 脚本文件内容如图 2 所示：

```
test.bat - 记事本
文件(F)  编辑(E)  格式(O)  查看(V)  帮助(H)

@echo off
set JAVA_HOME=D:\jdk-17_windows-x64\jdk-17.0.3.1
set PATH=%JAVA_HOME%\bin;%PATH%

rem 设置运行TestNG的依赖Jar包所在目录 (当前目录的lib子目录)
set LIB_DIR=.\lib

rem 设置被测测试类的目录 (顶级包目录的父目录)
set TO_BE_TEST_CLASSPAH=.\betest

rem 设置测试类的目录(顶级包目录的父目录) ,为当前目录的test子目录
set TEST_SUITE_CLASSPATH=.\test

rem 将运行TestNG的依赖Jar包放进CLASSPATH
setlocal EnableDelayedExpansion
set DEP_JAR=""
for /r %LIB_DIR% %%i in (*.jar) do if !DEP_JAR!==" (set DEP_JAR=%%i) else (set DEP_JAR=!DEP_JAR;!%%i)

set CLASSPATH=%CLASSPATH%;%TO_BE_TEST_CLASSPAH%;%TEST_SUITE_CLASSPATH%;%DEP_JAR%

java --add-opens java.base/java.lang=ALL-UNNAMED --add-opens java.base/java.math=ALL-UNNAMED --add-opens java.base/sun.net.util=ALL-UNNAMED org.testng.TestNG testng.xml -d test-output
```

图 2 Windows 下运行脚本内容及所需修改地方

在 test.bat 里学生只需要修改环境变量 JAVA_HOME 和 TO_BE_TEST_CLASSPAH，其他的环境变量如 LIB_DIR、TEST_SUITE_CLASSPATH 都是用的相对路径，所以无需修改。对于 Linux 或 Mac 的脚本文件的修改一样，如图 3 所示：

```
#!/bin/bash
#设置 JAVA_HOME
JAVA_HOME=/home/guixu/jdk/courrentjdk

#设置运行TestNG的依赖Jar包所在目录(当前目录的lib子目录)
LIB_DIR=../lib

#设置学生代码的目录 (顶级包目录的父目录)
TO_BE_TEST_CLASSPAH=../betest

#设置测试类的目录,为当前目录的test子目录
TEST_SUITE_CLASSPATH=../test

#将运行TestNG的依赖Jar包放进CLASSPATH
DEP_JAR=""
for jarfile in $(find $LIB_DIR | grep '\.jar$')
do
    if [ ! -n "$DEP_JAR" ];then
        DEP_JAR=$jarfile
    else
        DEP_JAR=$DEP_JAR:$jarfile
    fi
done
CLASSPATH=$CLASSPATH:$TO_BE_TEST_CLASSPAH:$TEST_SUITE_CLASSPATH:$DEP_JAR
export JAVA_HOME
export CLASSPATH
PATH=$JAVA_HOME/bin:$PATH
export PATH

java --add-opens java.base/java.lang=ALL-UNNAMED --add-opens java.base/java.math=ALL-UNNAMED --add-opens java.base/sun.net.util=ALL-UNNAMED org.testng.TestNG testng.xml -d test-output
```

图 3 Linux/Mac 下运行脚本内容及所需修改地方

由于测试代码同时测试二个编程练习，因此有三个 xml 文件对测试过程进行配置。其中 testng2.xml 是对编程第 2 题的测试配置，testng3.xml 是对编程第 3 题的测试配置。testng.xml 则是将多个测试组合起来，其内容如下所示：

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="All Test Suite">
    <suite-files>
        <suite-file path="testng2.xml"/></suite-file>
        <suite-file path="testng3.xml"/></suite-file>
    </suite-files>
</suite>
```

一般情况下这三个 xml 配置文件不需要修改。但是有的同学第 3 题没有完成，只能进行第 2 题的测试，这时需要修改 testng.xml 文件，将第 3 题的测试配置文件 testng3.xml 注释起来，xml 文件注释用<!-- -->，如下图所示：

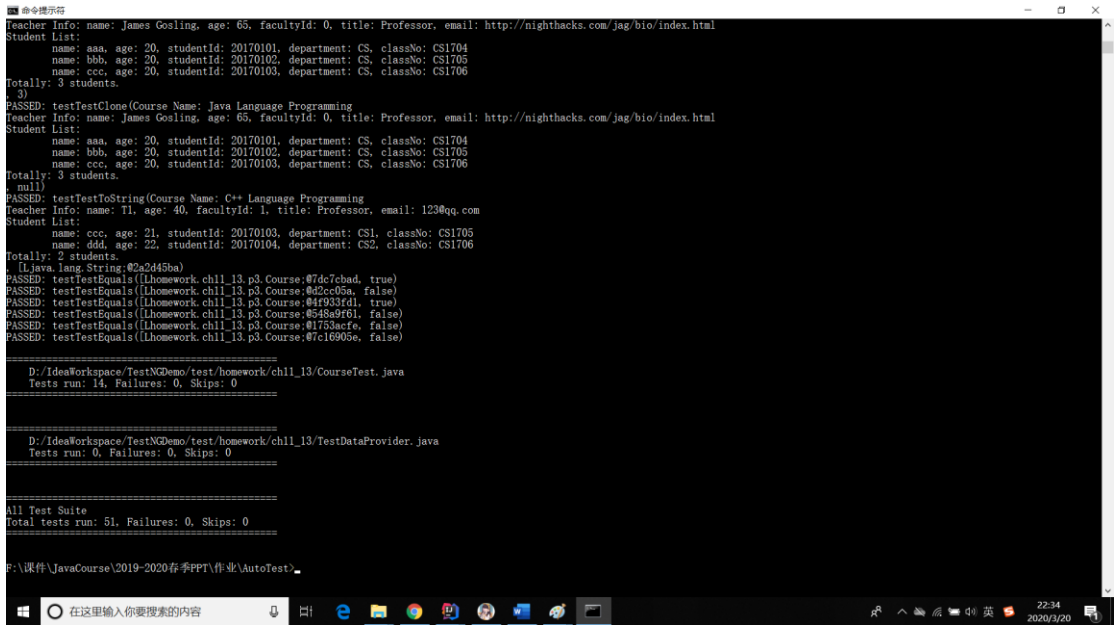
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="All Test Suite">
  <suite-files>
    <suite-file path="testng2.xml"></suite-file>
    <!--
    <suite-file path="testng3.xml"></suite-file>
    -->
  </suite-files>
</suite>
```

这个时候测试代码只会执行第 testng2.xml 里面的测试。

测试代码是在 JDK17 下编译的，因此最好保持 JDK 版本一致。运行测试脚本时，测试目录不要包含中文路径。

4. 测试程序运行及查看测试结果

在命令行进入到 AutoTest_update3 目录，运行 test.bat。运行结果如图 3 所示：



```
teacher Info: name: James Gosling, age: 65, facultyId: 0, title: Professor, email: http://nighthacks.com/jag/bio/index.html
Student List:
  name: aaa, age: 20, studentId: 20170101, department: CS, classNo: CS1704
  name: bbb, age: 20, studentId: 20170102, department: CS, classNo: CS1705
  name: ccc, age: 20, studentId: 20170103, department: CS, classNo: CS1706
Totally: 3 students.
PASSED: testTestClone(Course Name: Java Language Programming
Teacher Info: name: James Gosling, age: 65, facultyId: 0, title: Professor, email: http://nighthacks.com/jag/bio/index.html
Student List:
  name: aaa, age: 20, studentId: 20170101, department: CS, classNo: CS1704
  name: bbb, age: 20, studentId: 20170102, department: CS, classNo: CS1705
  name: ccc, age: 20, studentId: 20170103, department: CS, classNo: CS1706
Totally: 3 students.
PASSED: testTestToString(Course Name: C++ Language Programming
Teacher Info: name: TI, age: 40, facultyId: 1, title: Professor, email: 123@qq.com
Student List:
  name: ccc, age: 21, studentId: 20170103, department: CS1, classNo: CS1705
  name: ddd, age: 22, studentId: 20170104, department: CS2, classNo: CS1706
Totally: 2 students.
PASSED: testTestEquals(ILhomework.ch11_13.p3.Course:07dc7ebad, true)
PASSED: testTestEquals(ILhomework.ch11_13.p3.Course:042cc05a, false)
PASSED: testTestEquals(ILhomework.ch11_13.p3.Course:04f933fd1, true)
PASSED: testTestEquals(ILhomework.ch11_13.p3.Course:0548a9f61, false)
PASSED: testTestEquals(ILhomework.ch11_13.p3.Course:07f33cfe, false)
PASSED: testTestEquals(ILhomework.ch11_13.p3.Course:07c16905e, false)

=====
D:/IdeaWorkspace/TestNGDemo/test/homework/ch11_13/CourseTest.java
Tests run: 14, Failures: 0, Skips: 0

=====
D:/IdeaWorkspace/TestNGDemo/test/homework/ch11_13/TestDataProvider.java
Tests run: 0, Failures: 0, Skips: 0

=====
All Test Suite
Total tests run: 51, Failures: 0, Skips: 0

=====
E:\课件\JavaCourse\2019-2020春季PPT\作业\AutoTest\
```

图 3 命令行运行测试程序

这时进入到 AutoTest_update3 目录下的子目录 test-output，里面内容如图 4 所示：

test-output

共享 查看

、 此电脑 > 新加卷 (F:) > 课件 > JavaCourse > 2019-2020春季PPT > 作业 > AutoTest > test-output				
	名称	修改日期	类型	大小
	All Test Suite	2020/3/20 22:32	文件夹	
	junitreports	2020/3/20 22:32	文件夹	
	old	2020/3/20 22:32	文件夹	
	bullet_point.png	2020/3/20 22:32	PNG 文件	1 KB
	collapseall.gif	2020/3/20 22:32	GIF 文件	1 KB
	emailable-report.html	2020/3/20 22:32	Chrome HTML D...	32 KB
	failed.png	2020/3/20 22:32	PNG 文件	1 KB
	index.html	2020/3/20 22:32	Chrome HTML D...	103 KB
	jquery-1.7.1.min.js	2020/3/20 22:32	JavaScript 文件	92 KB
	navigator-bullet.png	2020/3/20 22:32	PNG 文件	1 KB
	passed.png	2020/3/20 22:32	PNG 文件	1 KB
	skipped.png	2020/3/20 22:32	PNG 文件	1 KB
vs (C:)	testng.css	2020/3/20 22:32	CSS 文件	1 KB
D:)	testng-reports.css	2020/3/20 22:32	CSS 文件	6 KB
E:)	testng-reports.js	2020/3/20 22:32	JavaScript 文件	4 KB
F:)	testng-results.xml	2020/3/20 22:32	XML 文档	40 KB

图 5 test-output 目录内容

打开 index.html，可以看到测试结果，如图 6 所示：

5. 提交测试包

为了方便作业提交后助教对学生的代码测试，学生需要把作业编译后的 class 文件拷贝到 AutoTest_update3 下的 betest 目录下。以 11-13 章编程题第二题为例，由于规定了包名必须是 homework.ch11_13.p3，因此 betest 的子目录结构应该如图 8 所示：

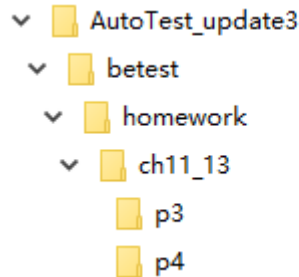


图 8 betest 的目录结构

学生则将工程编译好的这个包下面的 class 文件拷贝到 p3 子目录下。同时，将测试脚本里 TO_BE_TEST_CLASSPAH 的目录设置成相对目录。

对于 Windows 脚本 test.bat，相应语句改为：

```
set TO_BE_TEST_CLASSPAH=.\betest
```

对于 Linux/Mac 脚本 test.bat，相应语句改为：

```
TO_BE_TEST_CLASSPAH=./betest
```

注意斜杠前面有一个.，表示当前目录。

类似地，第三题作业拷贝到 betest\homework\ch11_13\p4 包下面。

最后把 AutoTest_update3 打包，随同作业源代码(.java 文件)、作业客观题答案一起提交。在将 AutoTest_update3 打包提交前，学生应该在本机运行修改后的脚本，确保测试代码可以测试 betest 下面的 class 文件。

在提交 AutoTest_update3 包前，测试代码脚本的 TO_BE_TEST_CLASSPAH 的目录可以指向工程目录里的包，如第 3 节所示，这样可以方便学生根据测试结果修改调试代码。

6. 测试运行的 JDK 版本问题

由于测试代码是在控制台下运行而不是在 IDEA 里运行，**所以要保证从控制台运行时启动 java 程序的 JDK 版本为 JDK17**。验证方法是在控制台运行 `java -version`，如下图所示：

```
E:\JavaCourse\2023-2024春季PPT\作业\第11-13章作业（2023-03-24更新）\AutoTest_update3>java -version
java version "17.0.3.1" 2022-04-22 LTS
Java(TM) SE Runtime Environment (build 17.0.3.1+2-LTS-6)
Java HotSpot(TM) 64-Bit Server VM (build 17.0.3.1+2-LTS-6, mixed mode, sharing)
```