2024-1 Software Engineering

HW2

22100579 Jinju Lee

1. (1p) Select two open-source projects from ASF java projects in Github.

List the selected project name here:

1. Common-jexl <https://github.com/apache/commons-jexl.git>
2. Cassandra: <https://github.com/apache/cassandra>

2. (1p) What build tools were used for your selected projects? (You can identify build tools for the selected projects by reading their project web sites. For Java projects, Maven (pom.xml), Gradle (build.graldle), and Ant (\*.xml) are popular ones. You may check other build tools.

> I’d like to choose maven for common-jexl because the project has pom.xml file.

3. (2p) In a build configuration file, find a section for the automated/regression testing part. Find a build command that conducts the testing only. 

|  |
| --- |
| Surefire plugin is using for unit testing. This section is setting about surefire.    This section is for test building, I guess. |

4. (2p) Build the selected projects. Put and explain the steps you have done for successful builds (including the steps installing the build tools). Capture the screens that show your build is successful. (Depending on each project, you may study basic commands of build tools by yourself. Some projects may provide guidelines or steps for how to build.)

1. install maven.

I am Windows user, so I downloaded zip file from <https://maven.apache.org/download.cgi> and unzip it in the directory I made.

After that, I need to add maven path into environment variable for using maven anywhere in my computer.

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

In here, click 환경변수 button. It means environment variable.

텍스트, 스크린샷, 디스플레이, 폰트이(가) 표시된 사진

자동 생성된 설명

Select Path variable and add new maven’s bin path on that.

텍스트, 전자제품, 스크린샷, 번호이(가) 표시된 사진

자동 생성된 설명

Now I can check if I can use maven anywhere by using this command

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

Ok. I am ready now!

1. Build using maven.

In this project, there is pom.xml file for build. So, I used ‘mvn clean install’ command to build it. I used sudo shell for let maven download what I need.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

…

텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

5. (1p) One of the useful functions of build tools is automatically importing or downloading the external libraries/packages/components. Find and put (copy and paste or a screenshot) (1) a section for external libraries from the configuration file and (2) the part of build execution logs that import or download the external libraries.

1. Section for external libraries are <dependencies> part in pom.xml

텍스트, 문서, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

2. and the part of build execution logs that import or download the external libraries are… almost whole part in logs, I guess. Its too long to capture all the logs, but it’s like this.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

6. (1p) Find the manual of build tools and execute more functionalities such as creating an executable or distributable package, documents, generating various reports, etc. Then, show the commands you used and explain their results.

for create executable JAR, maven-assembly-plugin function can be used like this.   
텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명  
and when use the ‘mvn package’ command, JAR file on target directory is created!

텍스트, 스크린샷, 소프트웨어, 폰트이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 메뉴, 폰트이(가) 표시된 사진

자동 생성된 설명

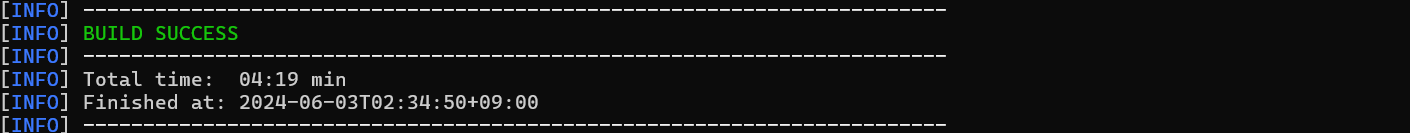
Secondly, maven create documents also. I added this part to pom.xml for let maven create documents.

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

And using command ‘mvn site’, now I got document of common-jexl project in target/site directory!





텍스트, 스크린샷, 메뉴이(가) 표시된 사진

자동 생성된 설명

7. (2p) Write what you learned from the HW tasks.

When I check the homework, I worry about doing it and afraid it so much because I am not fluent with Java and java projects. But while do the homework, I surprised a lot because there are so many well-made build tools and they allow me to build and analyze the java projects I never used before, very easily and comfortably. I think tools like that are very meaningful that they let people can use anything without domain information and skills about them. And I think Its kind of similar with the concept encapsulation. They make user don’t know about the things user don’t need to know. How sweet!

\* The following tasks are advanced and challenging ones. But these tasks are typical tasks that professional developers do in their projects.

8. (2p) Modify the original source code by intentionally introducing a bug. Then, check the existing regression test to catch the bug you introduced. (Must not be errors but purely test failures.) Capture and put a screenshot showing the failure (which means the regression test successfully detected the bug you introduced).

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

I changed this part. *this.stackOverflow* value should be *size*, but I added 1.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

And the test case catches the bug I made!

~~9.(3p) In case your modification adding a bug could not be detected by the regression test, developers must update the test case. If you face such a case, modify the test case that can detect the bug and explain your change. Capture and put a screenshot showing the test failure.~~