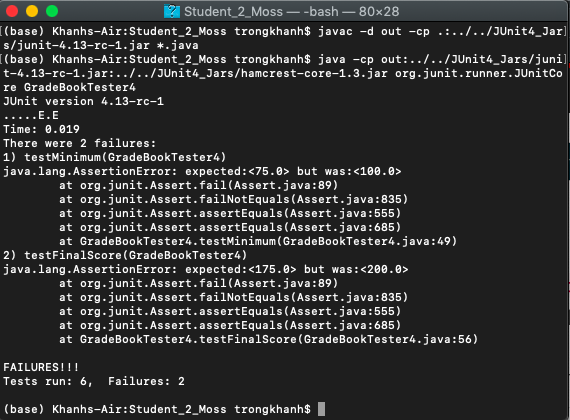
**Running Junit on command line**

1. **First approach:**

* Students can submit the tests in either Junit 4 or Junit 5. Therefore, we need to successfully run both versions of Junit.
* I thought about distinguishing Junit 4 and Junit 5 tests, and running each one with appropriate version of Junit. Junit 4 is easier. The only two jar files we need are hamcrest-core-1.3.jar and junit-4.13-rc-1.jar.

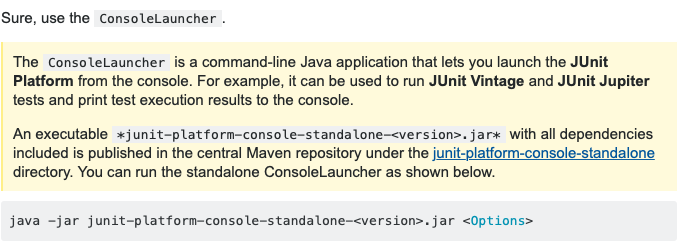


* However, this syntax was just for running Junit 4 only. We cannot run Junit 5 with this syntax and jar files. Then, I looked for a way to run Junit 5 test.

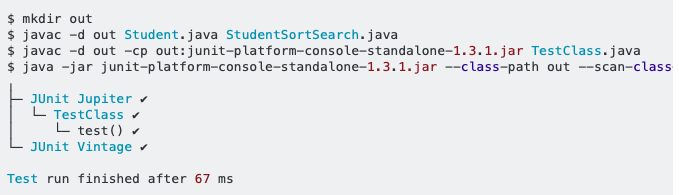
1. **Second approach:**

* I used google to look for what I needed to run Junit 5 test. Here was the link I found the answer for:

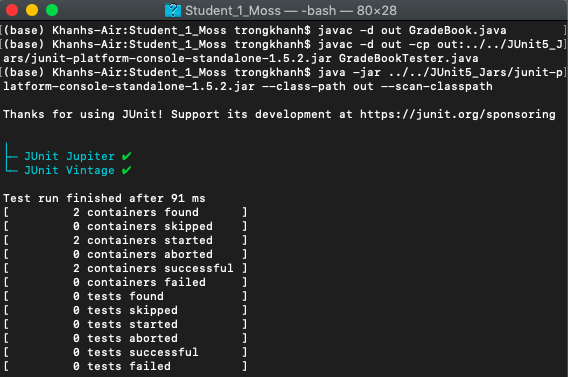
<https://stackoverflow.com/questions/52373469/how-to-launch-junit-5-platform-from-the-command-line-without-maven-gradle/52373592#52373592>



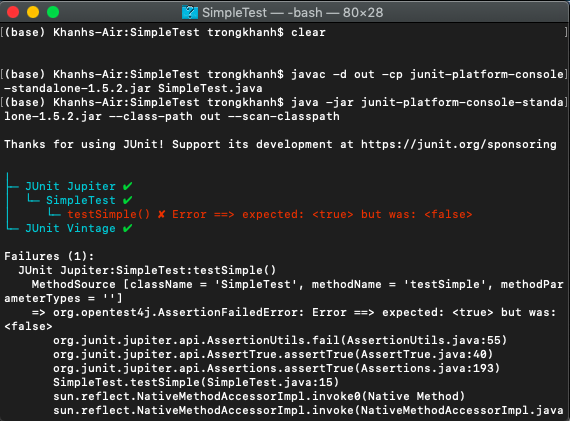
* Here was the expected output:



* Here was what I got:



* The Junit console launcher just found the two containers, but it could not find the test inside. This was not correct.
* I tried to create my own test and tested it. Here was the output:



* This made me think that there was something wrong with the –classpath. However, I tried many times with many syntaxes. Nothing worked at all.
* I tried to find the solution for this problem and got this discussion on stackoverflow:

<https://stackoverflow.com/questions/45869932/unable-to-run-tests-with-junit5-console-launcher>

* In the case they discuss, there was something wrong with the classpath. I did the same thing but nothing seemed to work.
* Therefore, I came up with another way to do it.

1. **Third approach:**

* Because I can run the test successfully on IntelliJ, I utilized their command.
* This was the command IntelliJ used to run the Junit test:



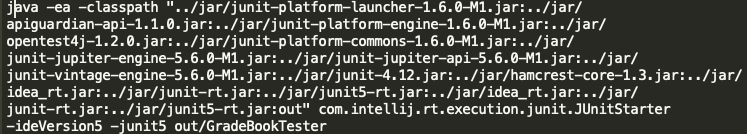
* I used some simple python code to extract jar files that IntelliJ used. After that, I eliminated the jar files from JVM, and unnecessary syntax. Here was the shortest command that can work:



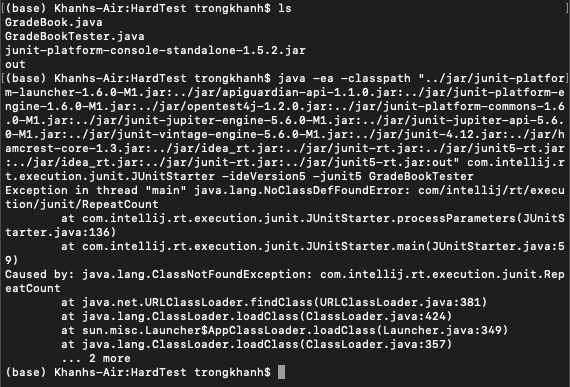
* The result of this command:



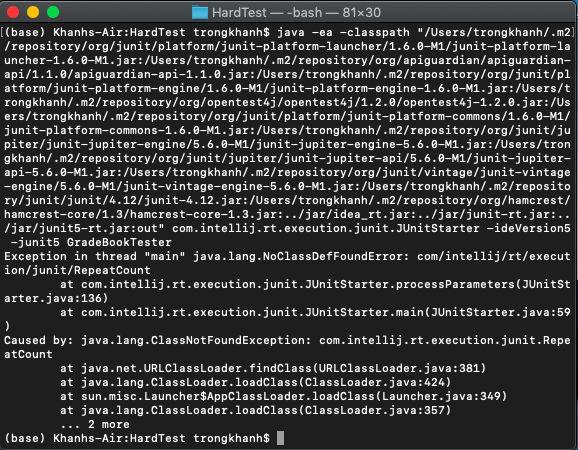
* Noticing that in the command, it used 13 jar files. Therefore, I downloaded those Junit jar files from <https://mvnrepository.com>
* There were idea\_rt.jar, junit5-rt.jar, and junit-rt.jar I could not find them online. Therefore, I went to the directory that contained those files to copy those files out.
* Then, I ran the command line which I referred those jar files to the external ones that I downloaded. Here was the new command that I used:

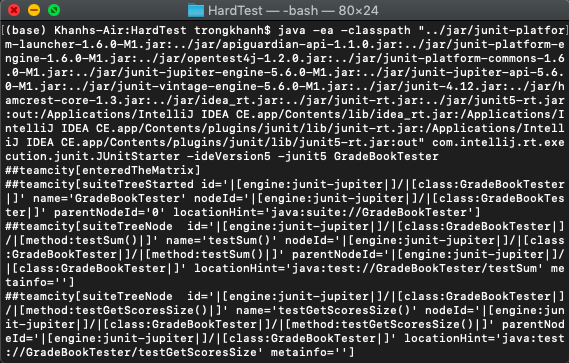


* However, the result was not what I expected:



* I doubt that the problem may come from jar files from Junit, or jar files from the runtime IntelliJ. Thus, I ran two more commands to figure it out where the problem was. One I kept the original directory of Junit, appending with the new directory for IntelliJ runtime. One I kept the original directory of IntelliJ runtime, appending with the new directory of Junit.
* I found out that the downloaded Junit jar files will work with the original IntelliJ runtime jar files. However, the original Junit jar cannot work with the IntelliJ runtime jar files that I copied to the new directory.
* Here were what happened:

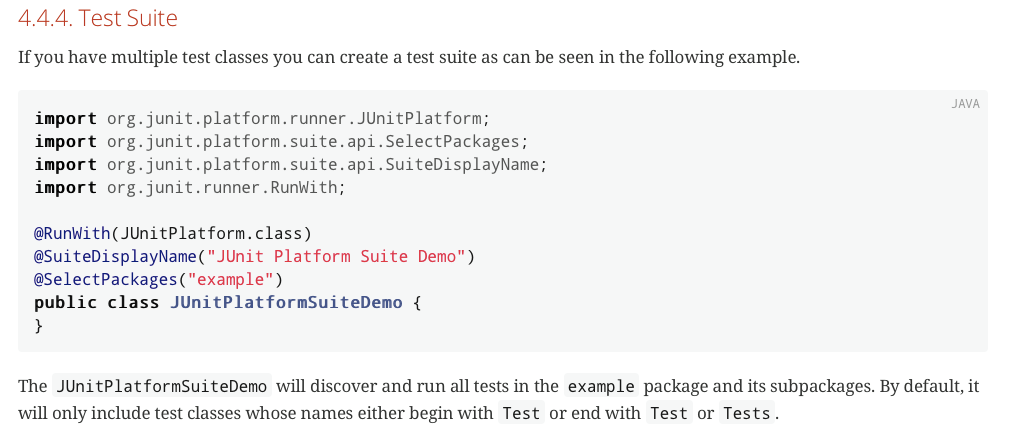




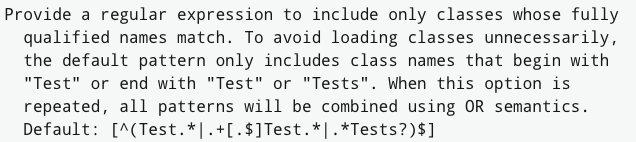
* I concluded that besides the Junit library, we also need a launcher that can run the test. In this case, I used the JUnitStarter from IntelliJ.
* My suggestion: Maybe we can run a sample test in IntelliJ with imported Junit 5 library to have the environment set up. Then, we can use that environment to run other tests.
* I am still looking for another way we can run independently with IntelliJ, which means with enough jar files, we can compile the test and run it.

1. **An accident search**

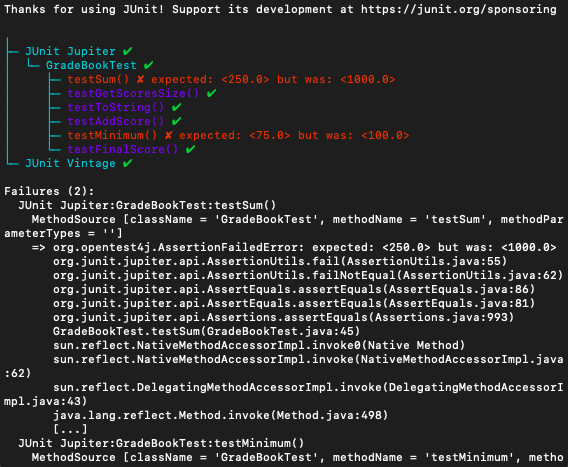
* I expected the problem I was dealing with was from the console launcher, which I called a runner. Therefore, I searched for “Junit 5 test runner”. The only thing I can refer to was the Junit 5 User Guide.
* I looked for the keyword “runner” in the guide, which somehow led me to this part:



* The last line said that the Suite can just include the classes whose names either begin with Test or end with Test or Tests. I had a feeling that this maybe the problem.
* I found the keyword “begin with Test” and “end with Test”, which led me to this:



* Then I tried to change the name of the testing class from “GradeBookTester.java” to “GradeBookTest.java” and run the same command that I found on stackoverflow:



* That was what I got. I cannot conclude 100% that by doing this, I can run with all given testing file. But in this case, it worked. I will try with other test to figure it out.

1. **Personal thinking**

* I considered the task was easy. However, I could not finish it. One of the reasons was I did not read the documentation carefully. Therefore, I knew that for tasks in the future, I will need to refer to documetation carefully before jumping into solving the problems.
* Writing a clear documentation is also important. I found this solution by accident, which means I could not find what I need directly from this User Guide. I am not blaming the development team. This is just to remind myself to write a clear documentation (e.g.: javadoc) in my future courses.