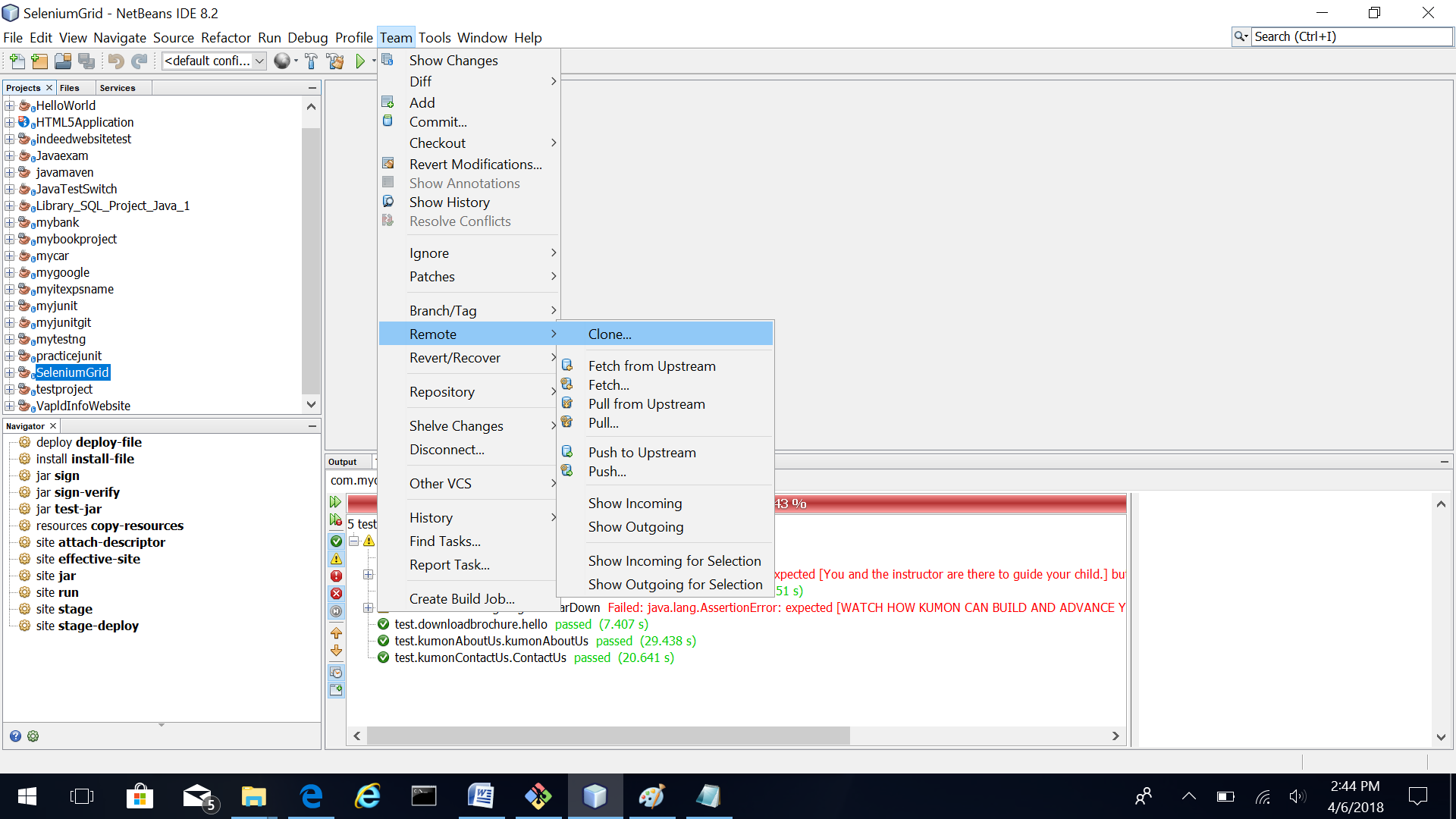
# Connecting NetBeans IDE project to GitHub repository.

These steps are used to clone and execute an already existing NetBeans Java Maven Application project in GitHub repository. If you are creating project from scratch then refer to the following subsection - Using NetBeans IDE to build and test all Selenium TestNG files and using GitBash for accessing GitHub.

1. Clone the project by clicking Team->Remote->Clone

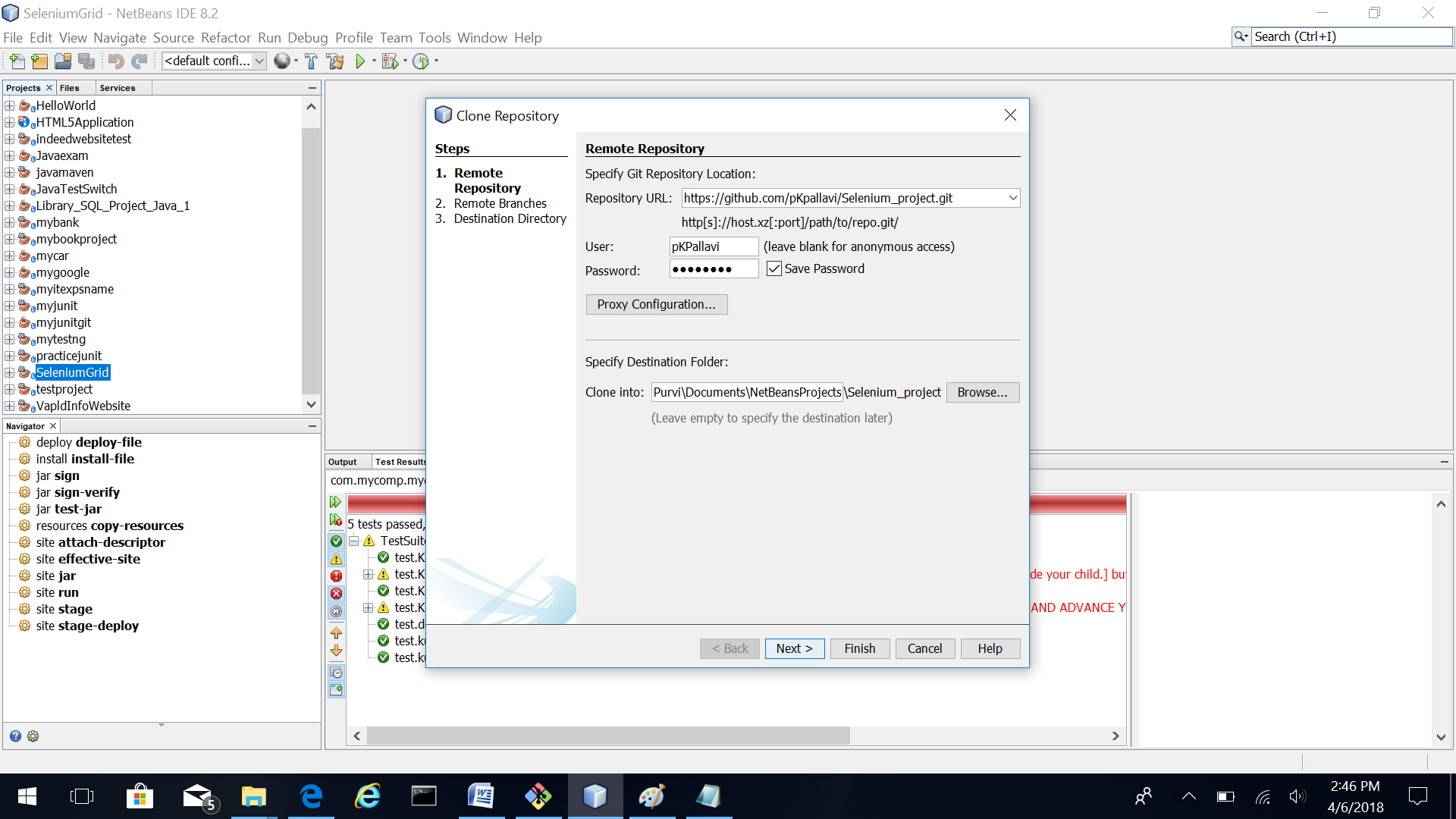


1. Provide the clone repository url.

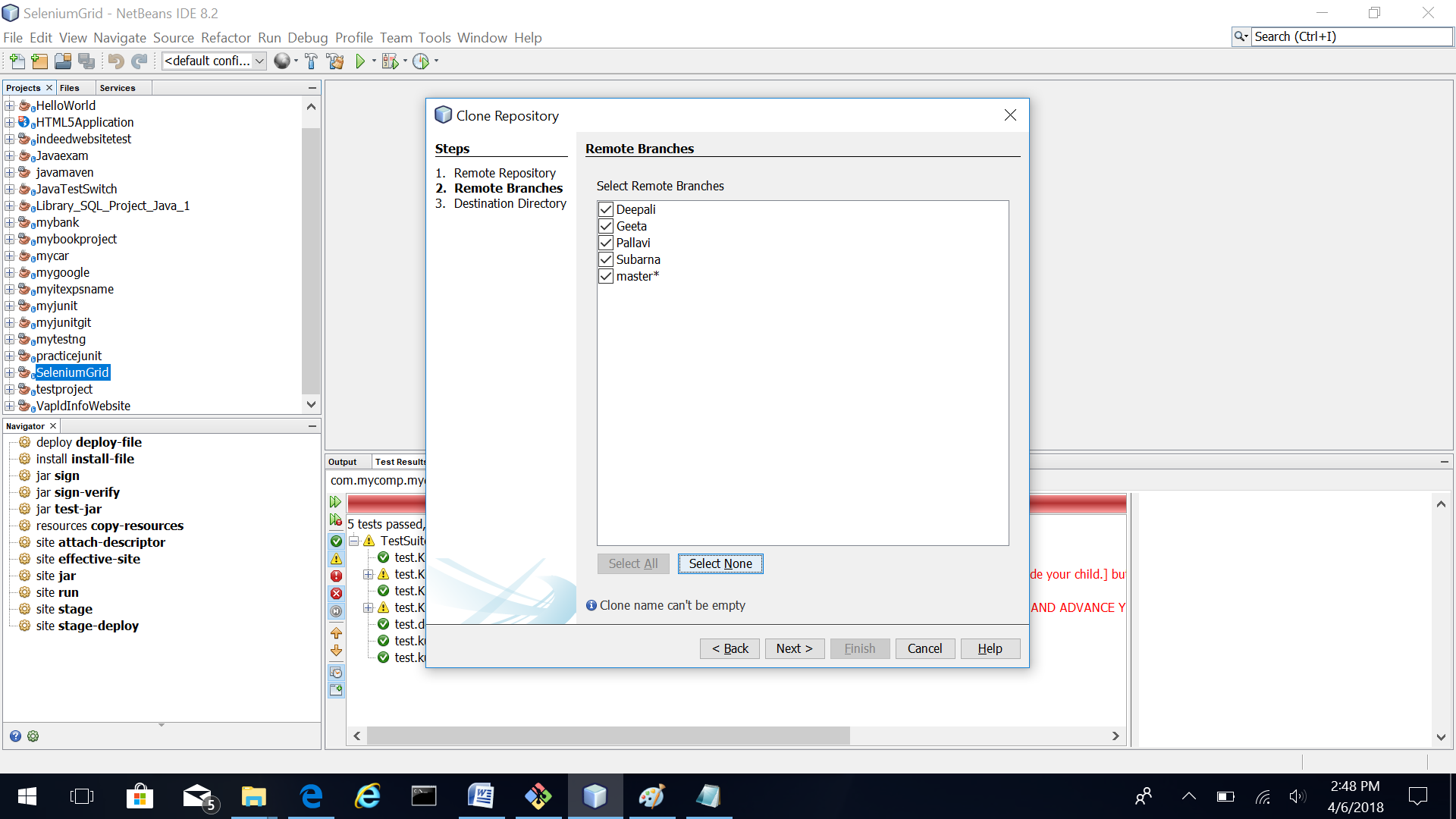
<https://github.com/pKpallavi/Selenium_project.git>

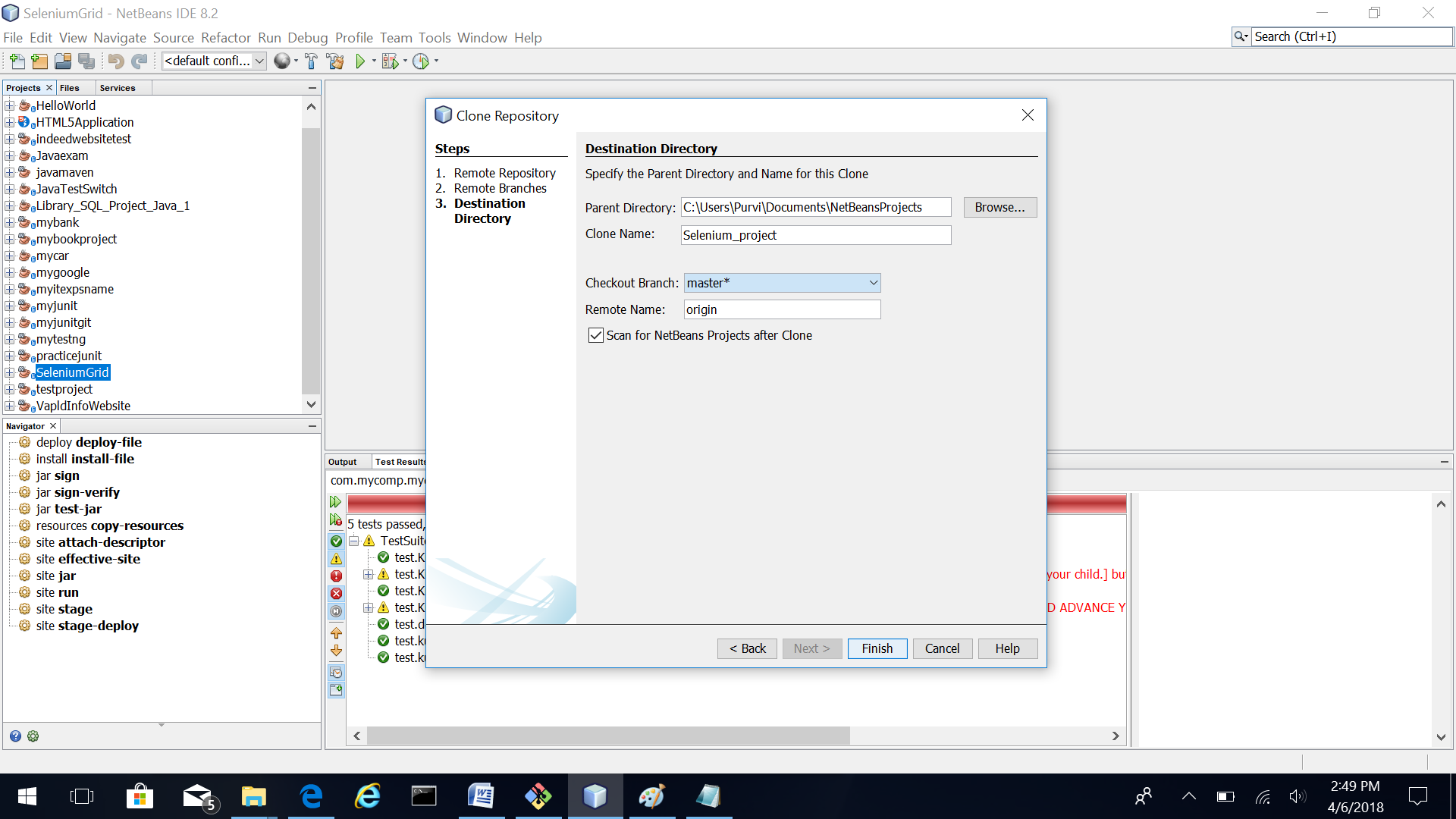
Enter your GitHub username and password which is used to login in the website.

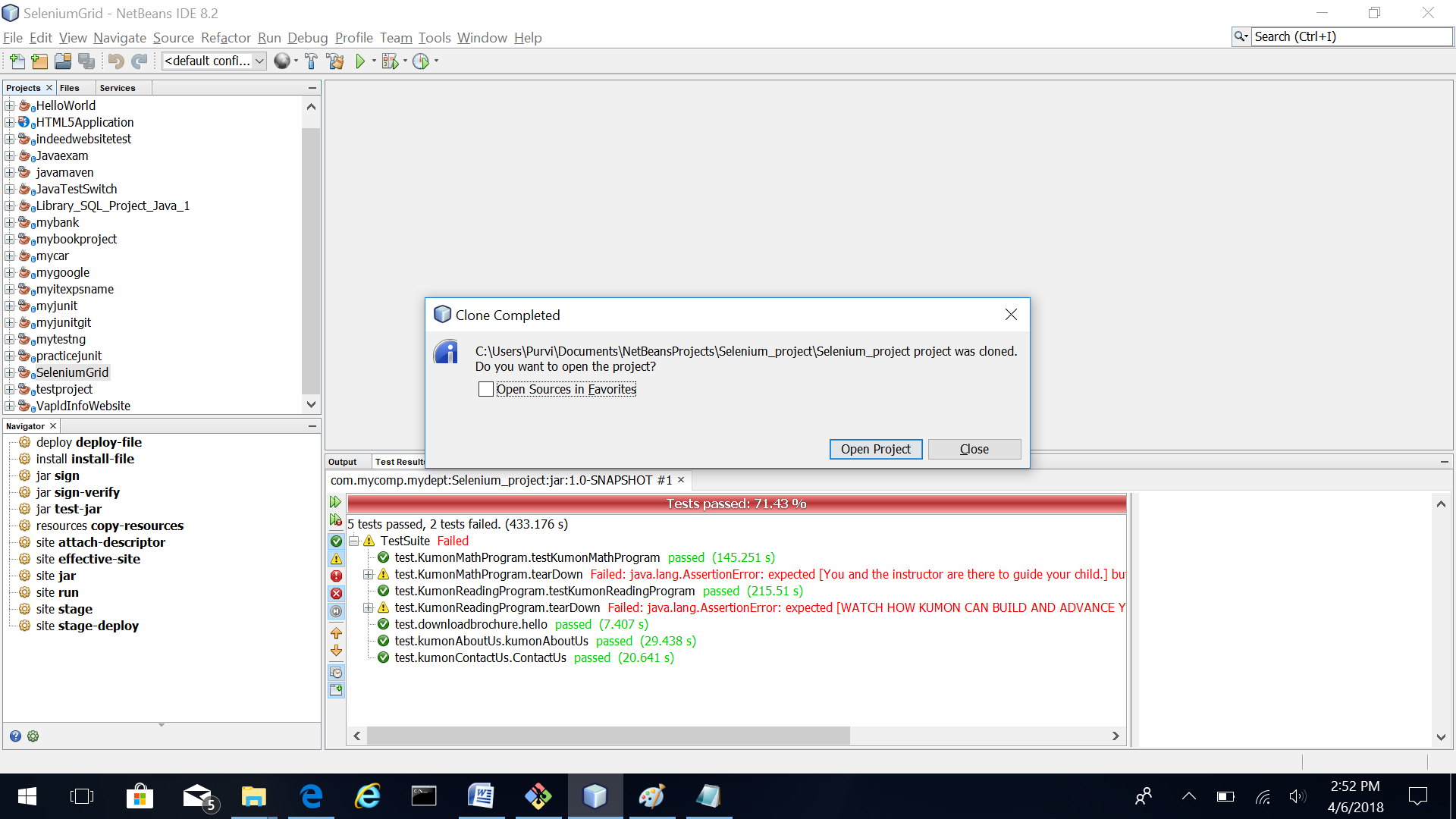
Provide the Directory path in “Clone Into” box.

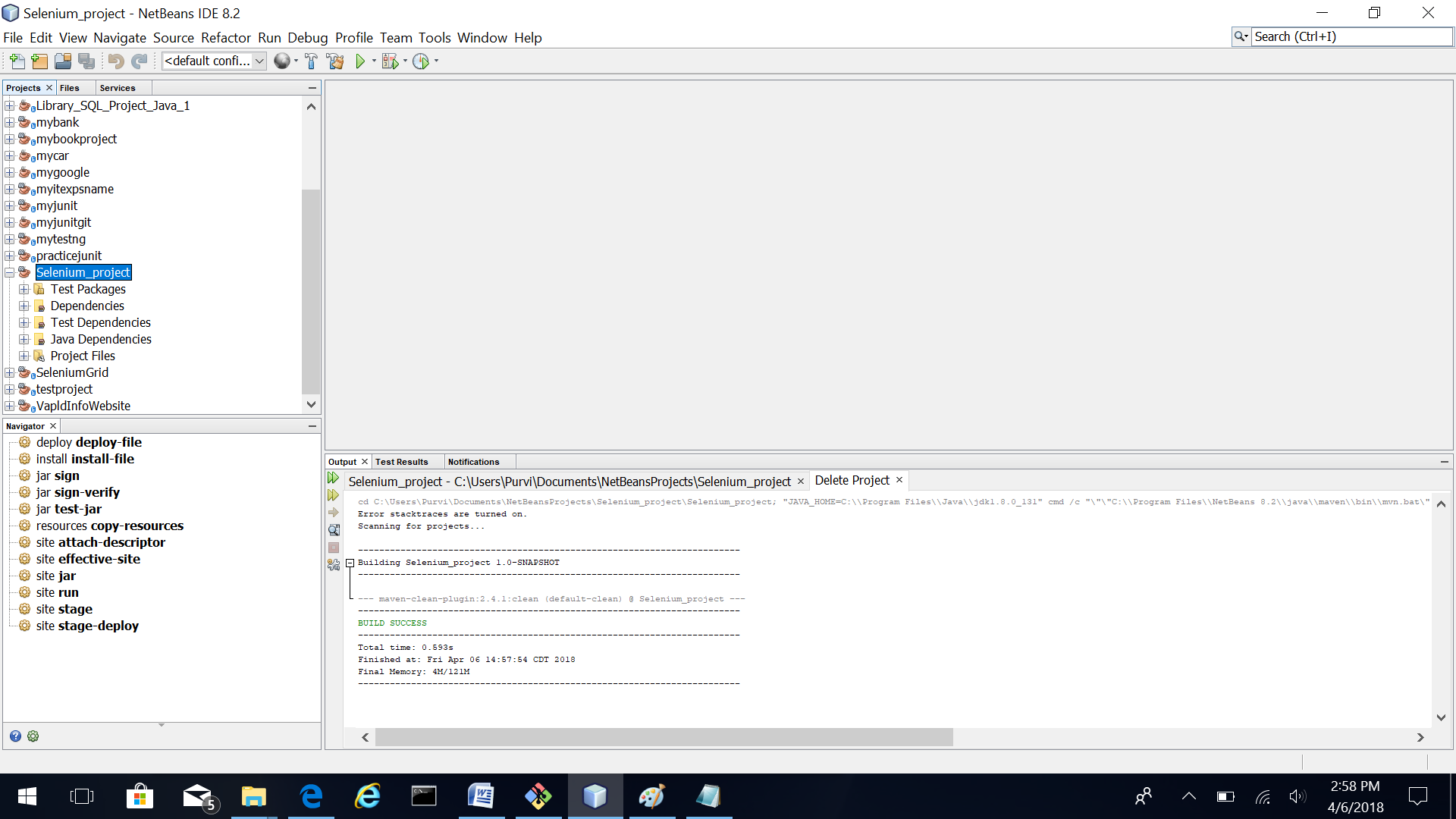


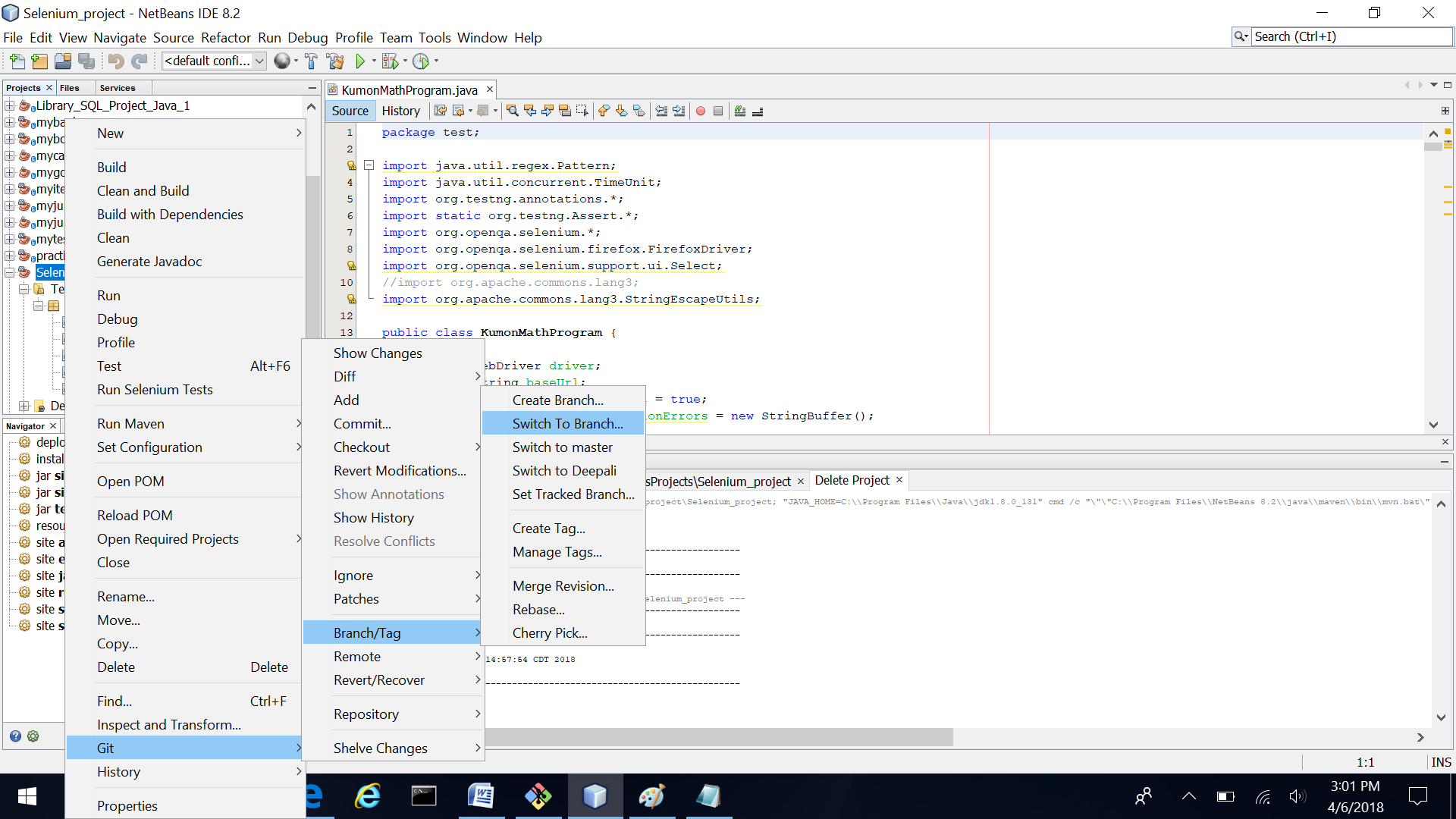
1. Select all remote branches and click next.

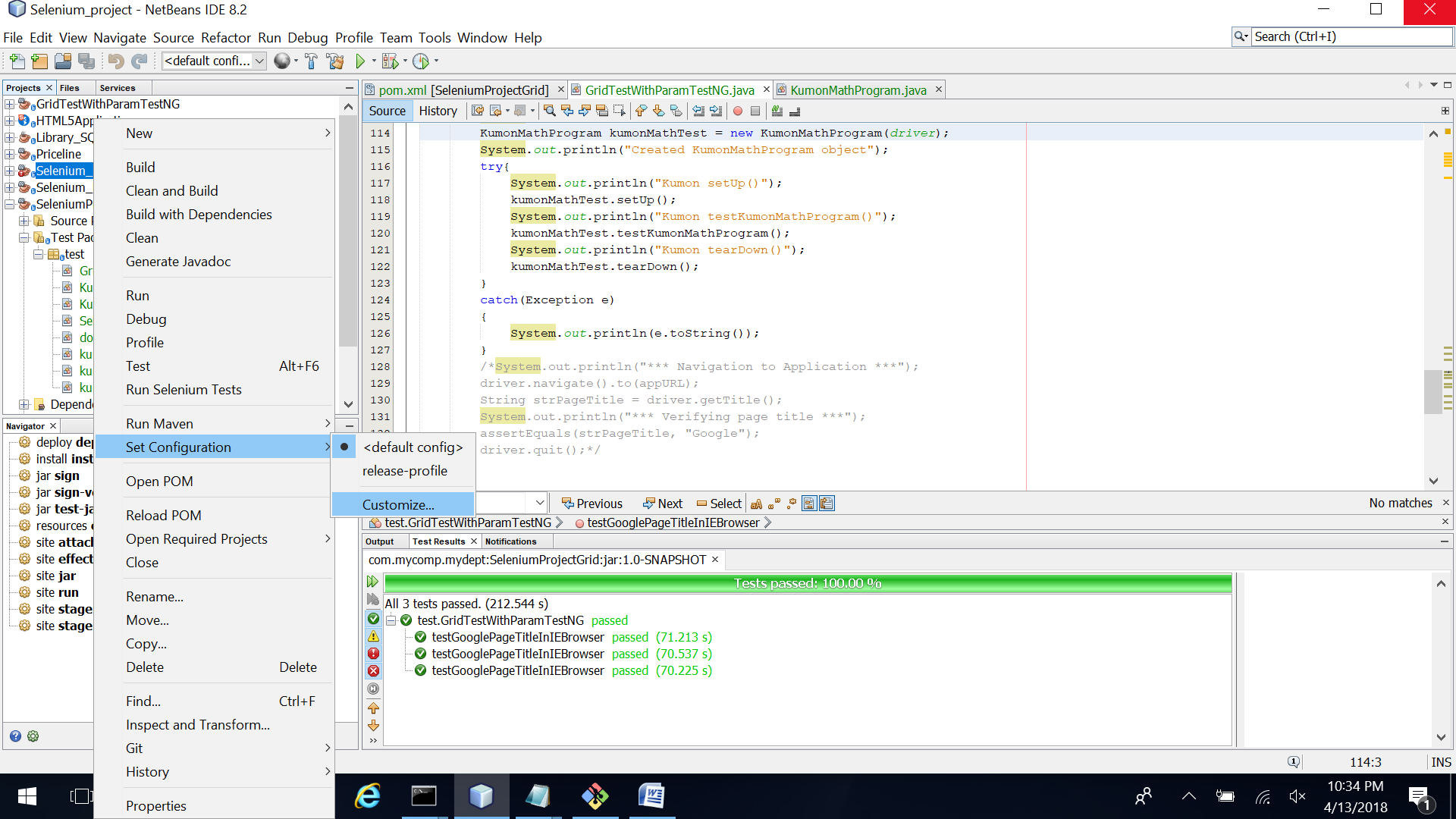


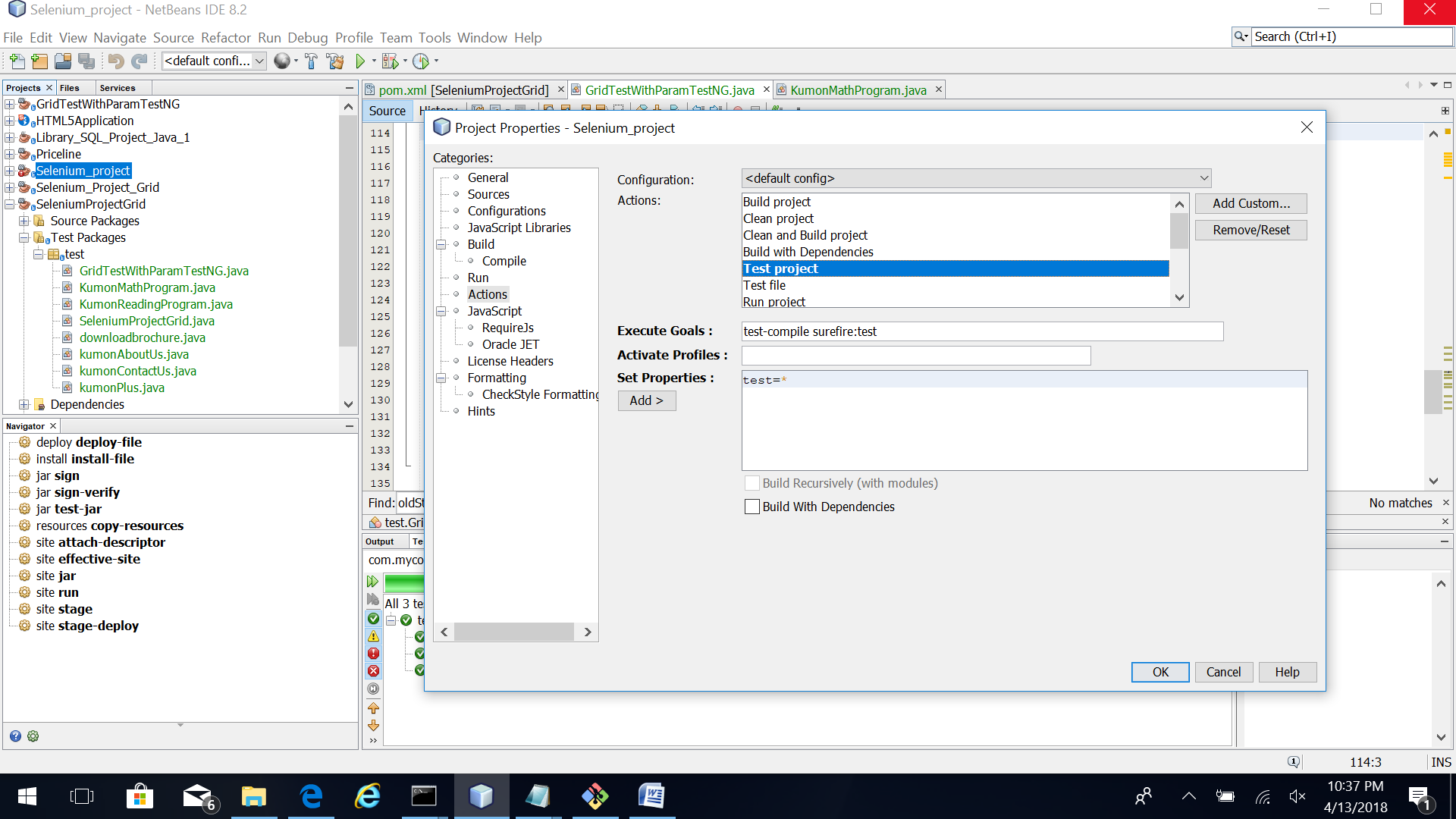
1. In the Destination Directory screen leave options as is. Later you can change the Checkout branch to your branch. Click finish button. 
2. After the repository is cloned following dialog will appear. Click open project.

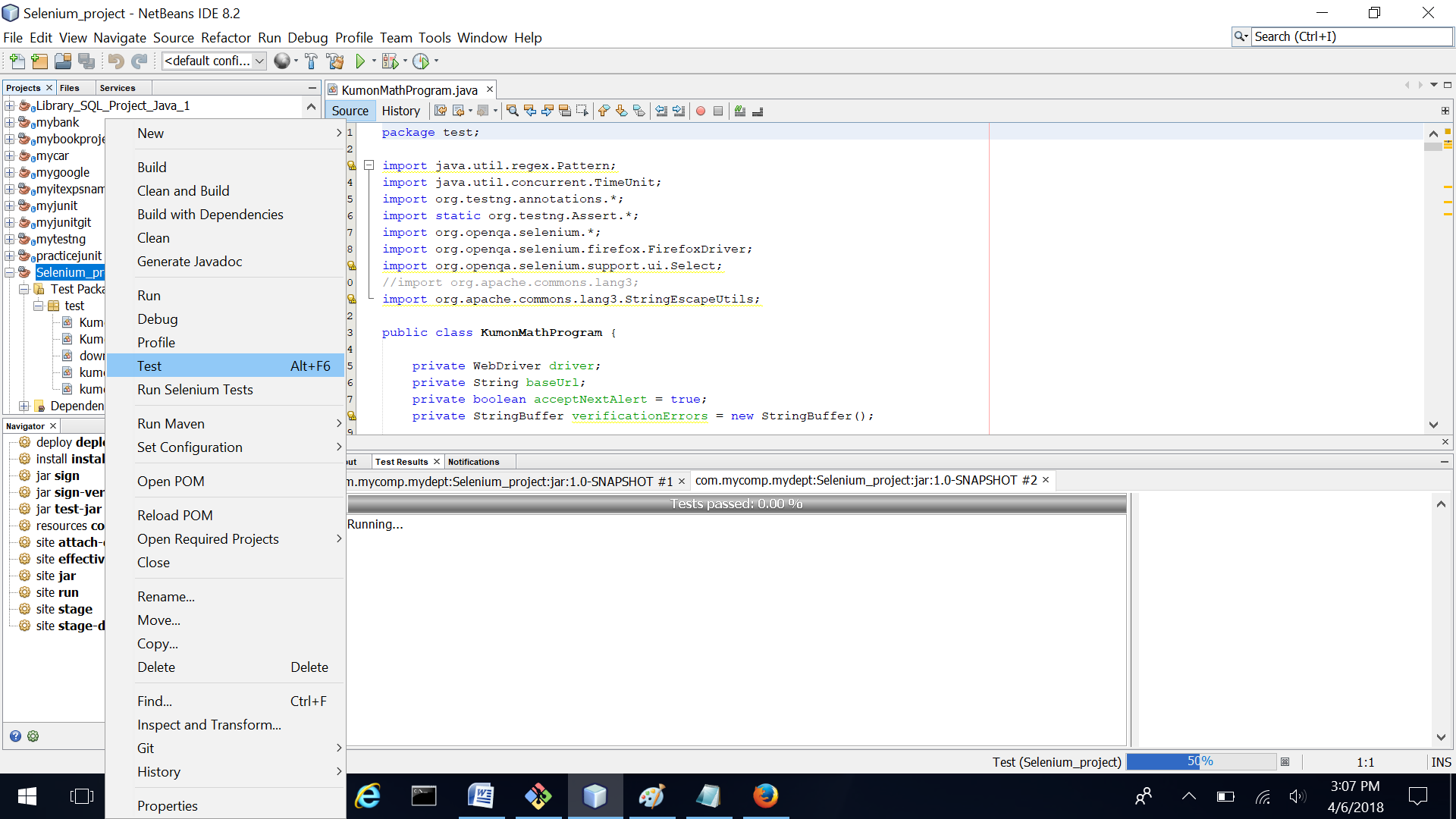
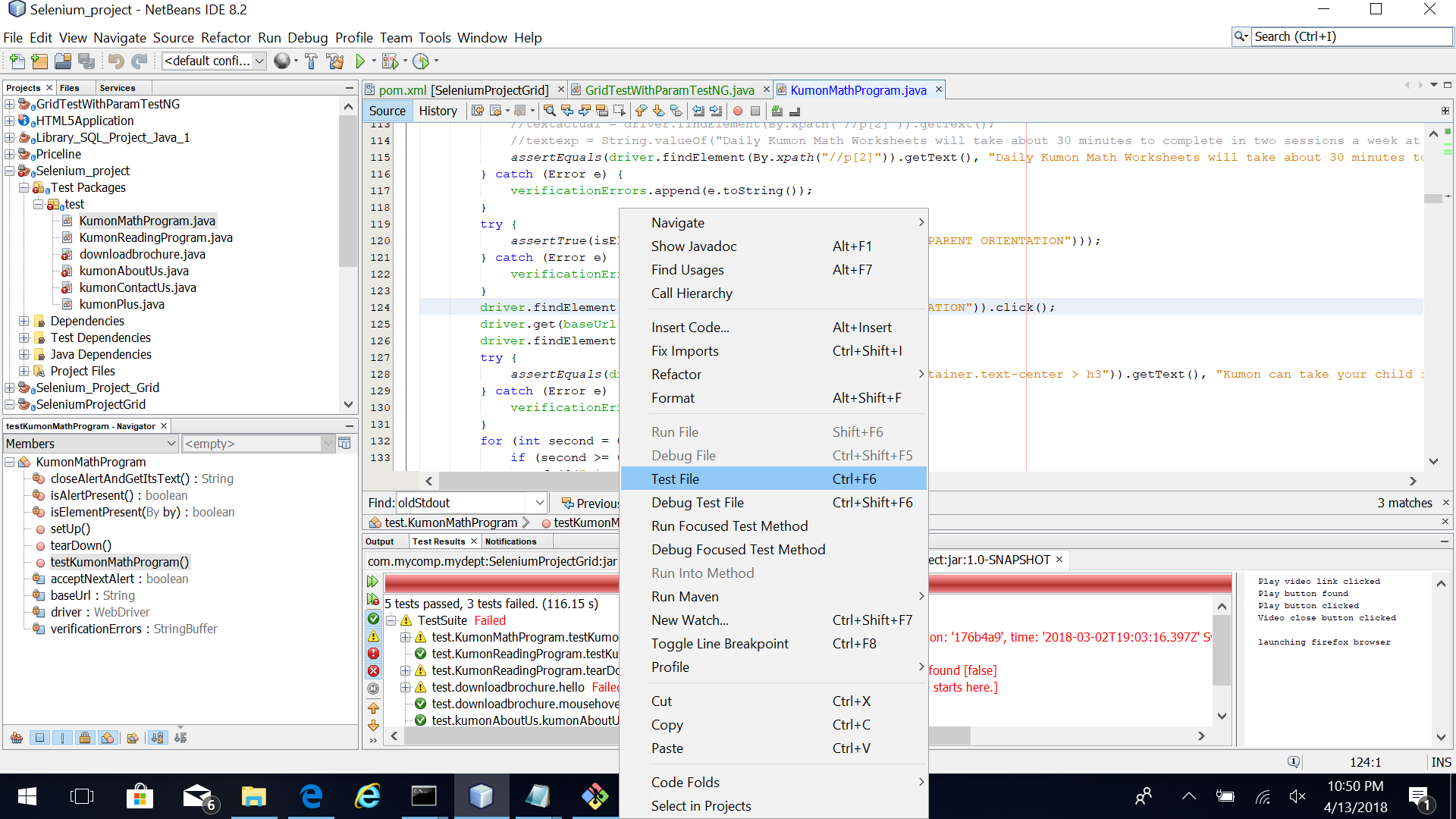
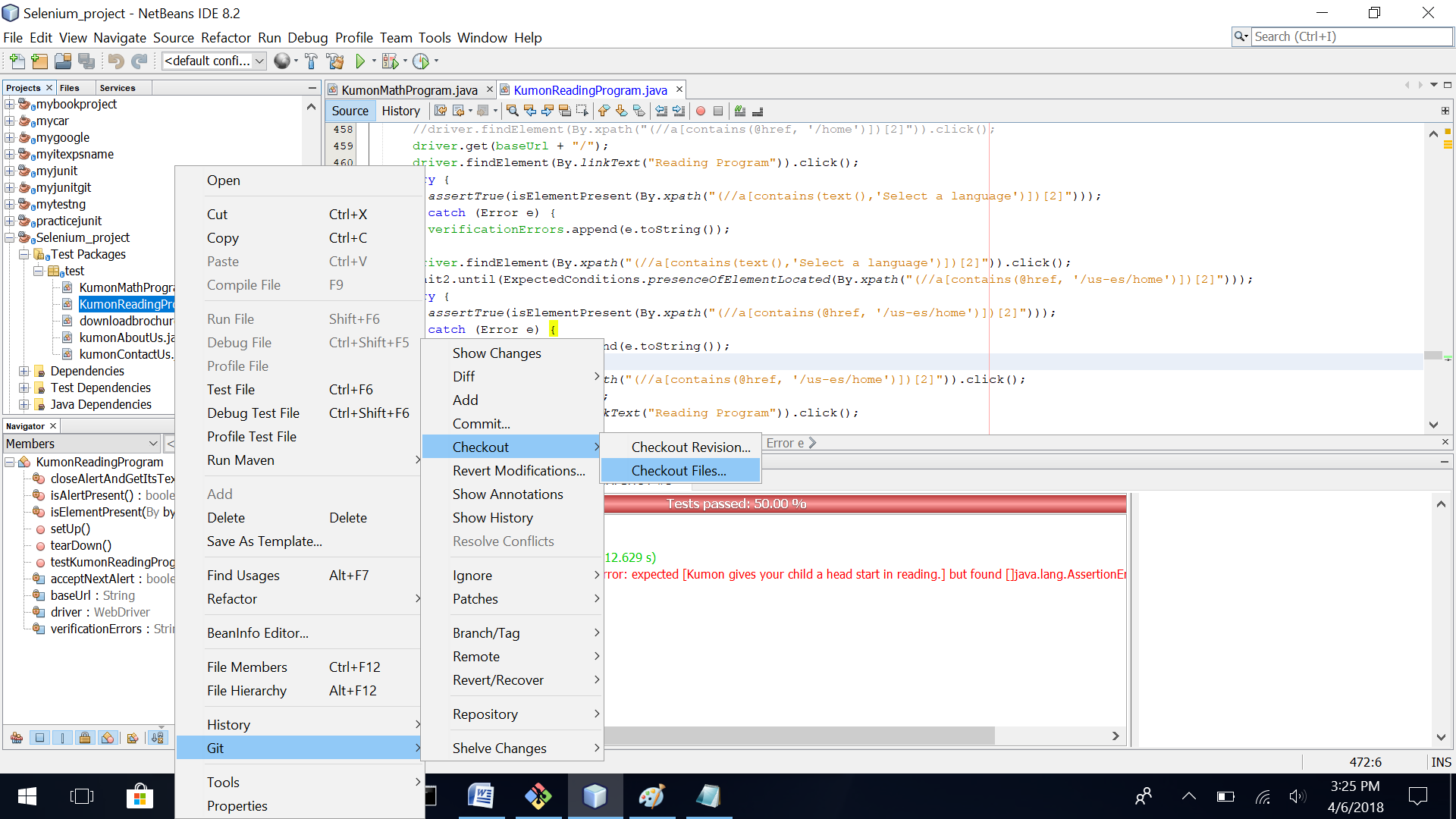
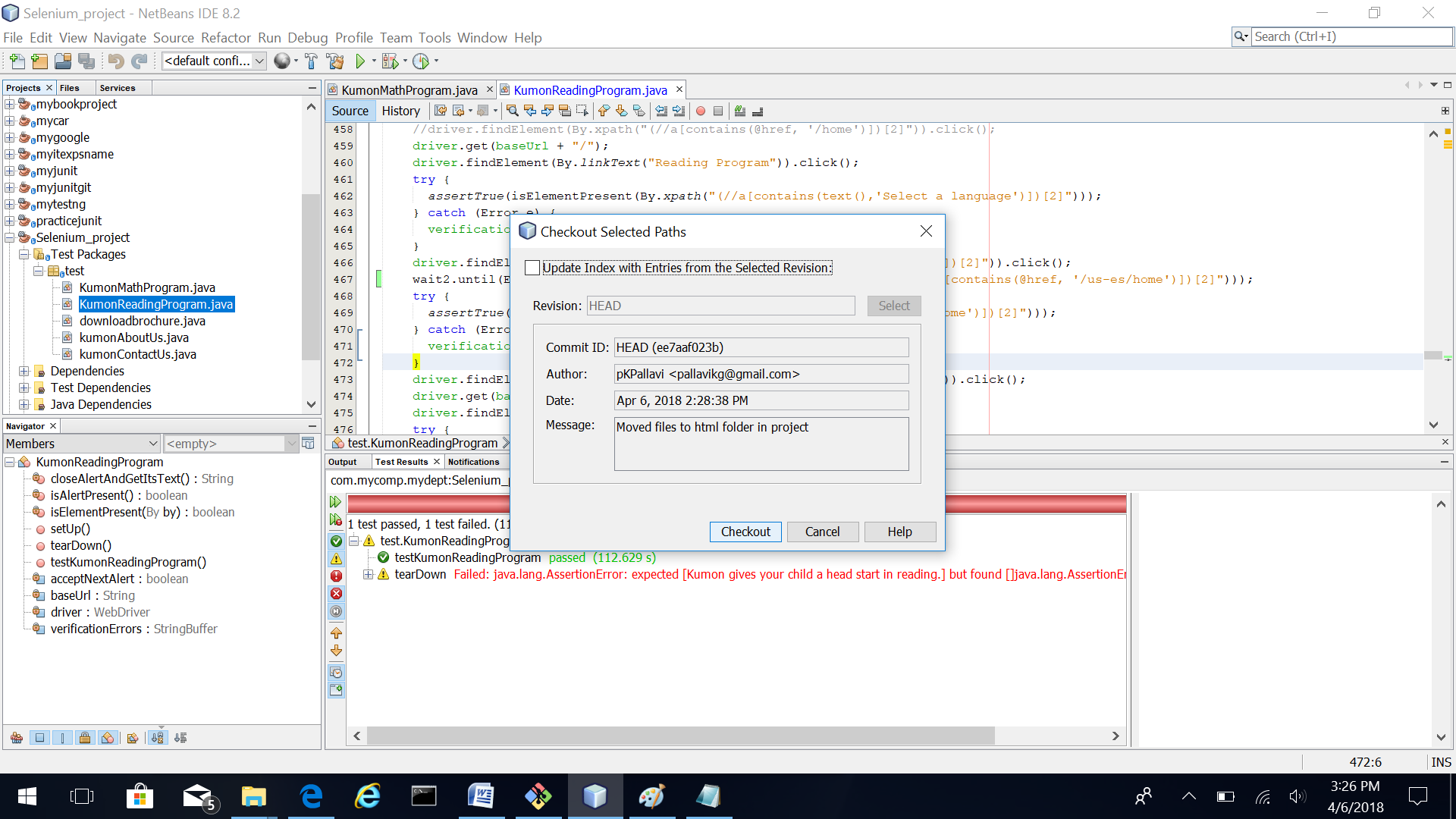
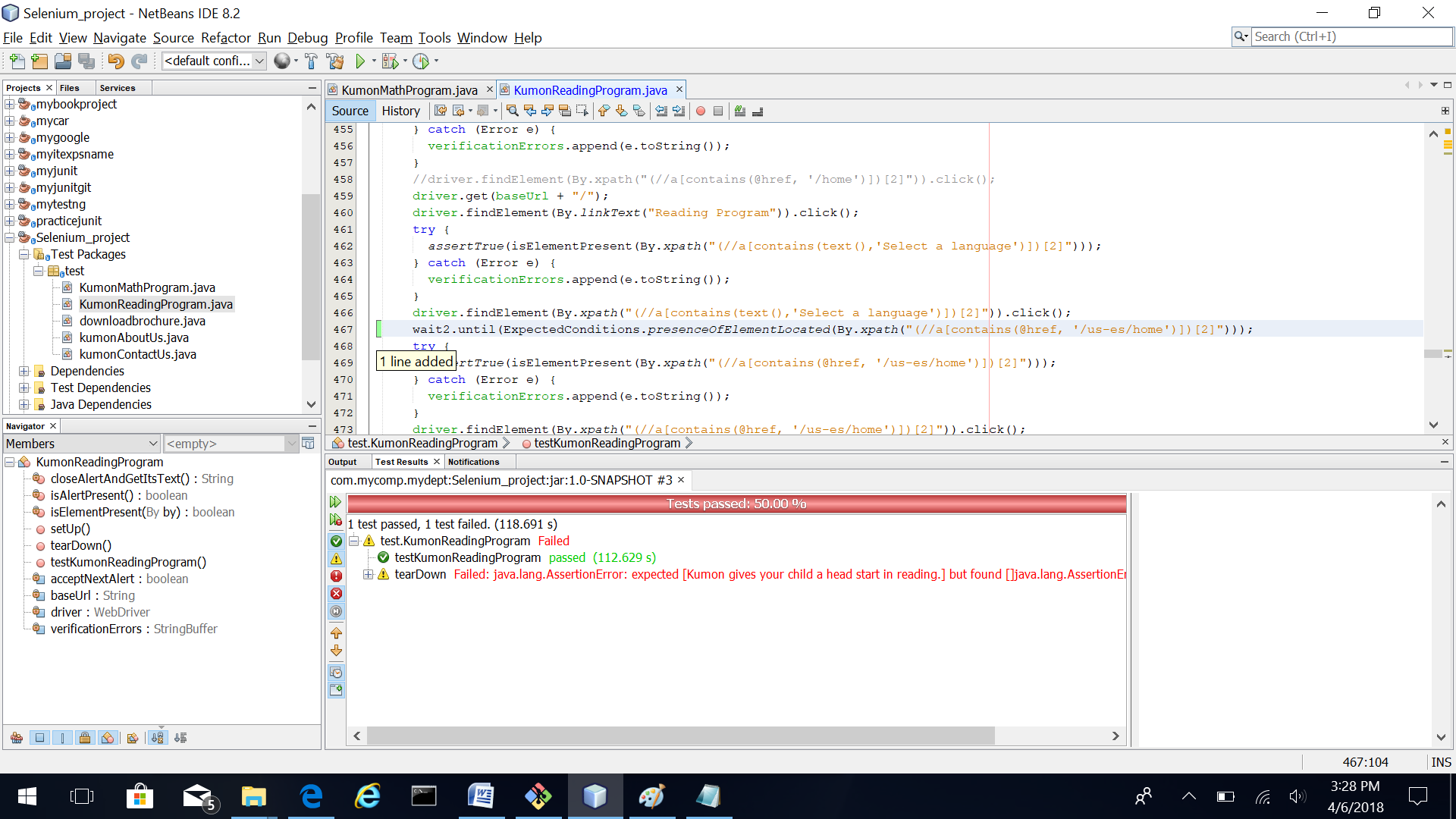
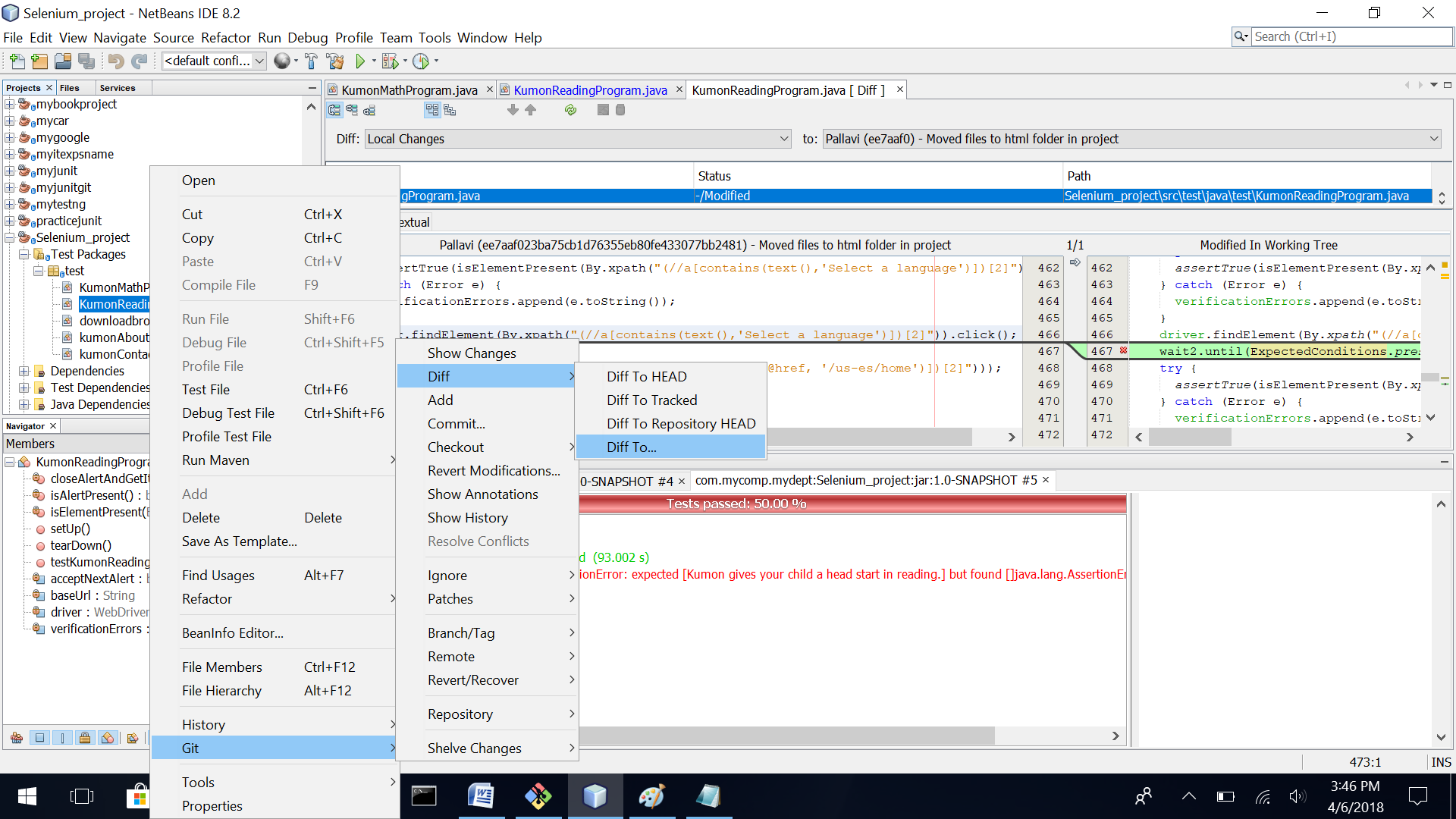


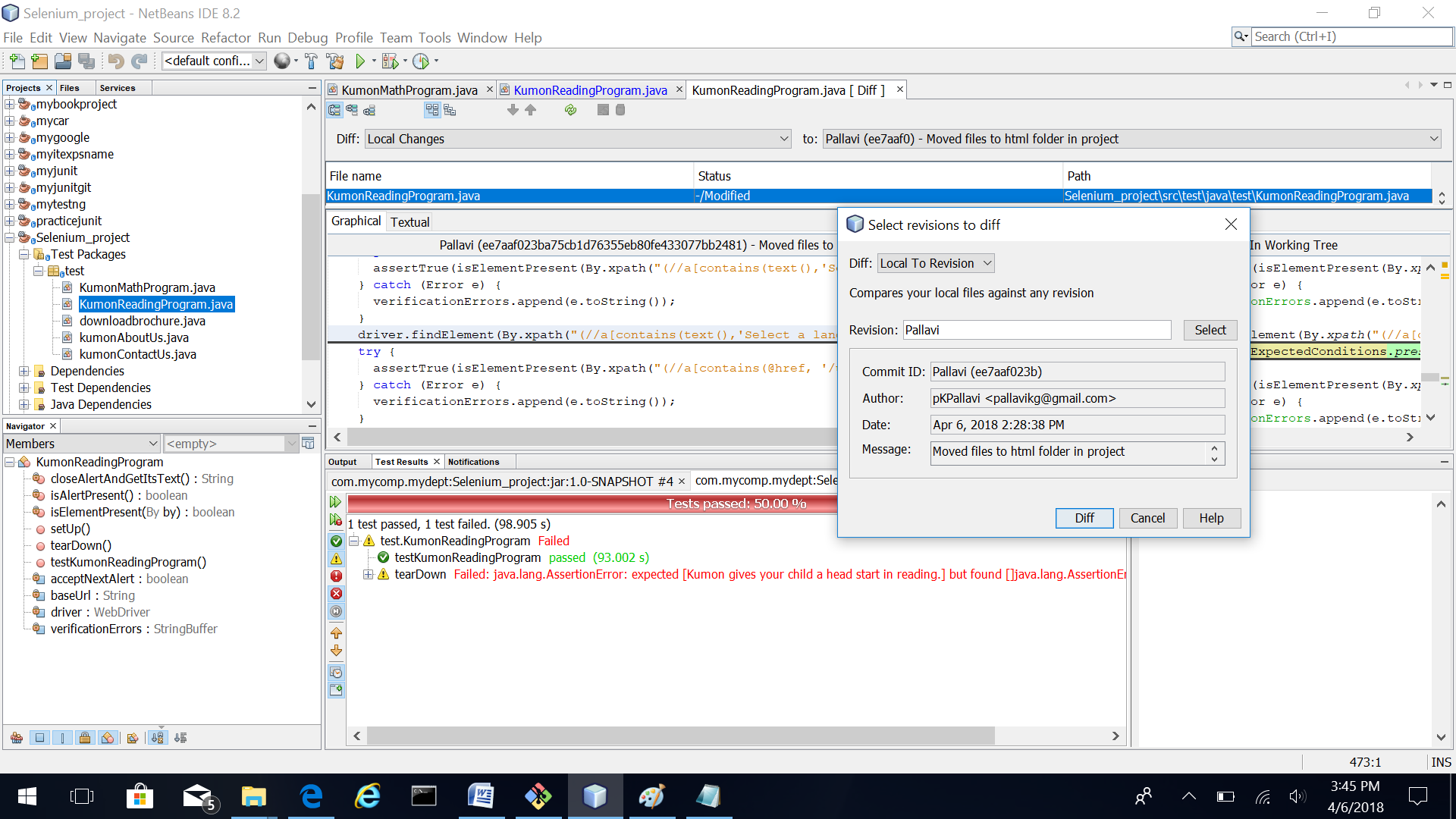
1. Project is loaded. 
2. You can switch to branches as shown in the snapshot below.

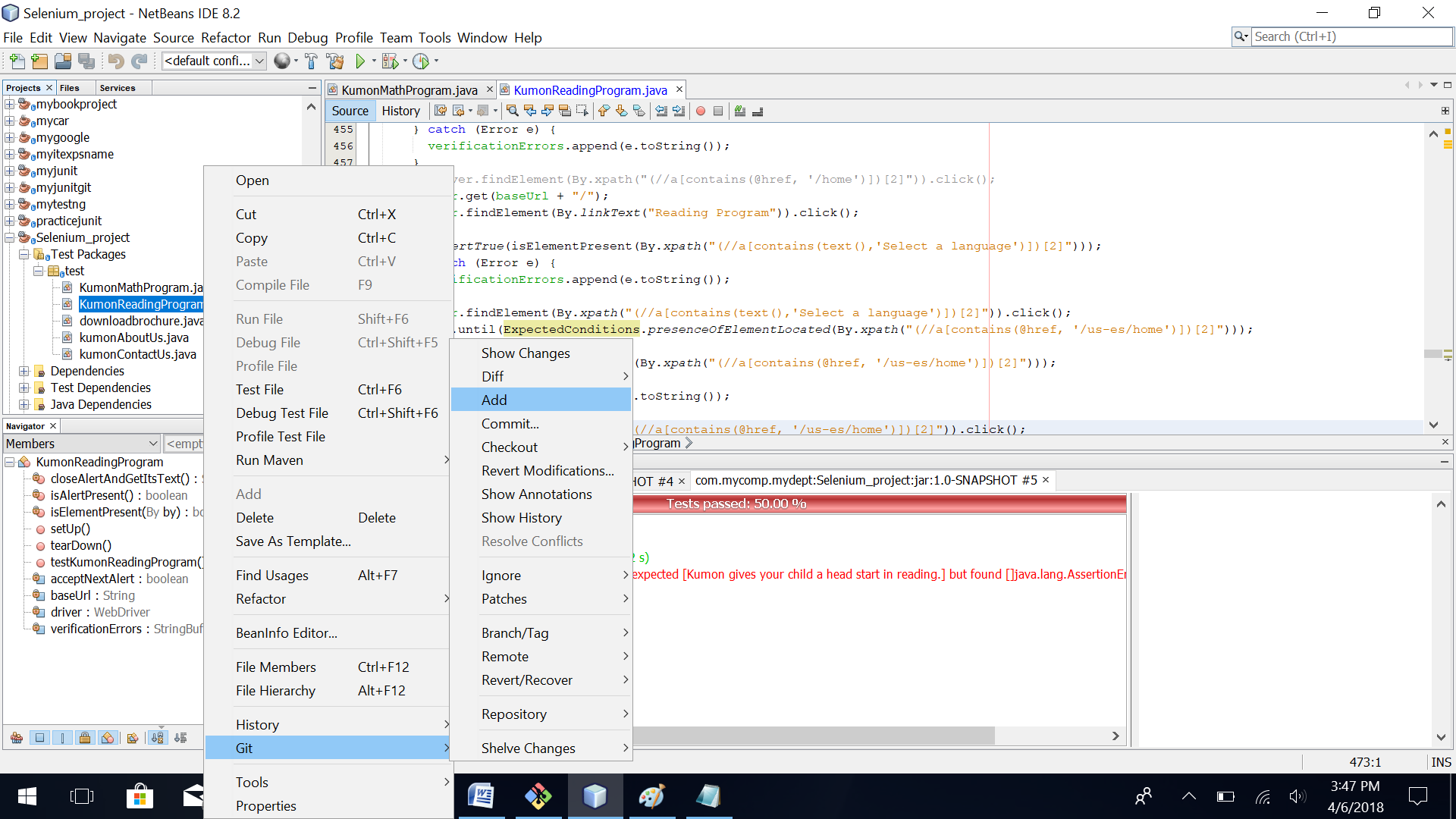
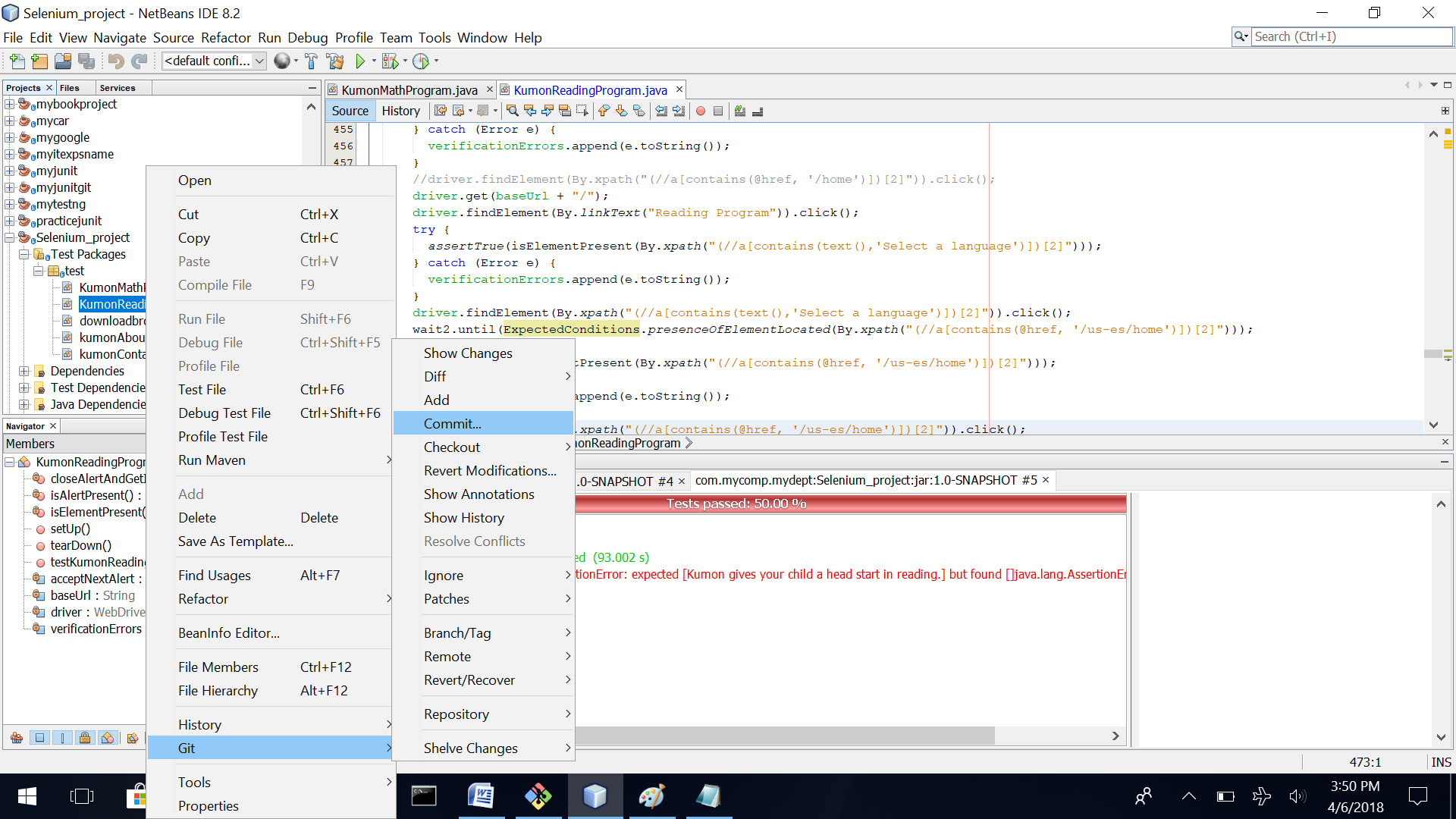
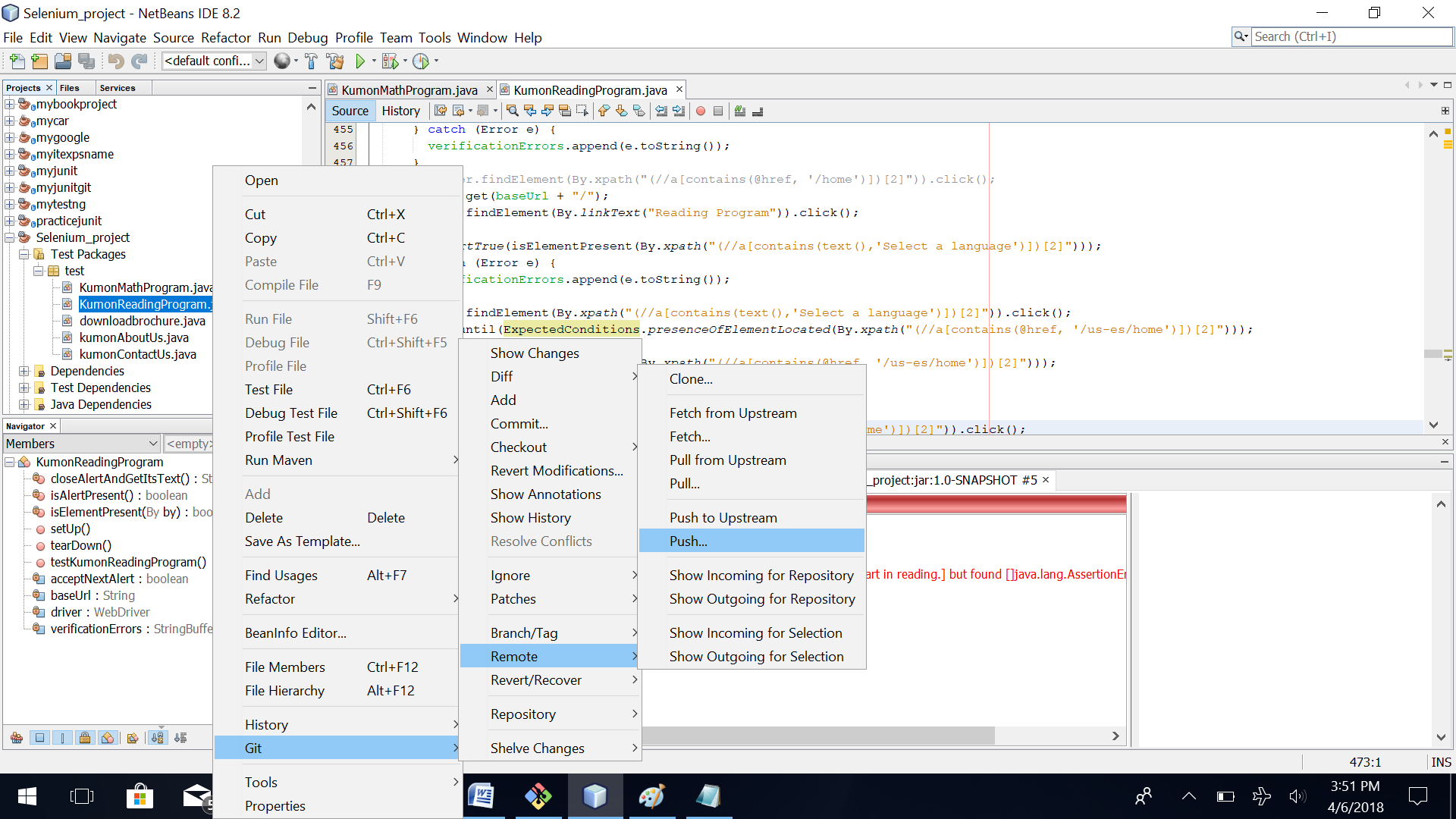


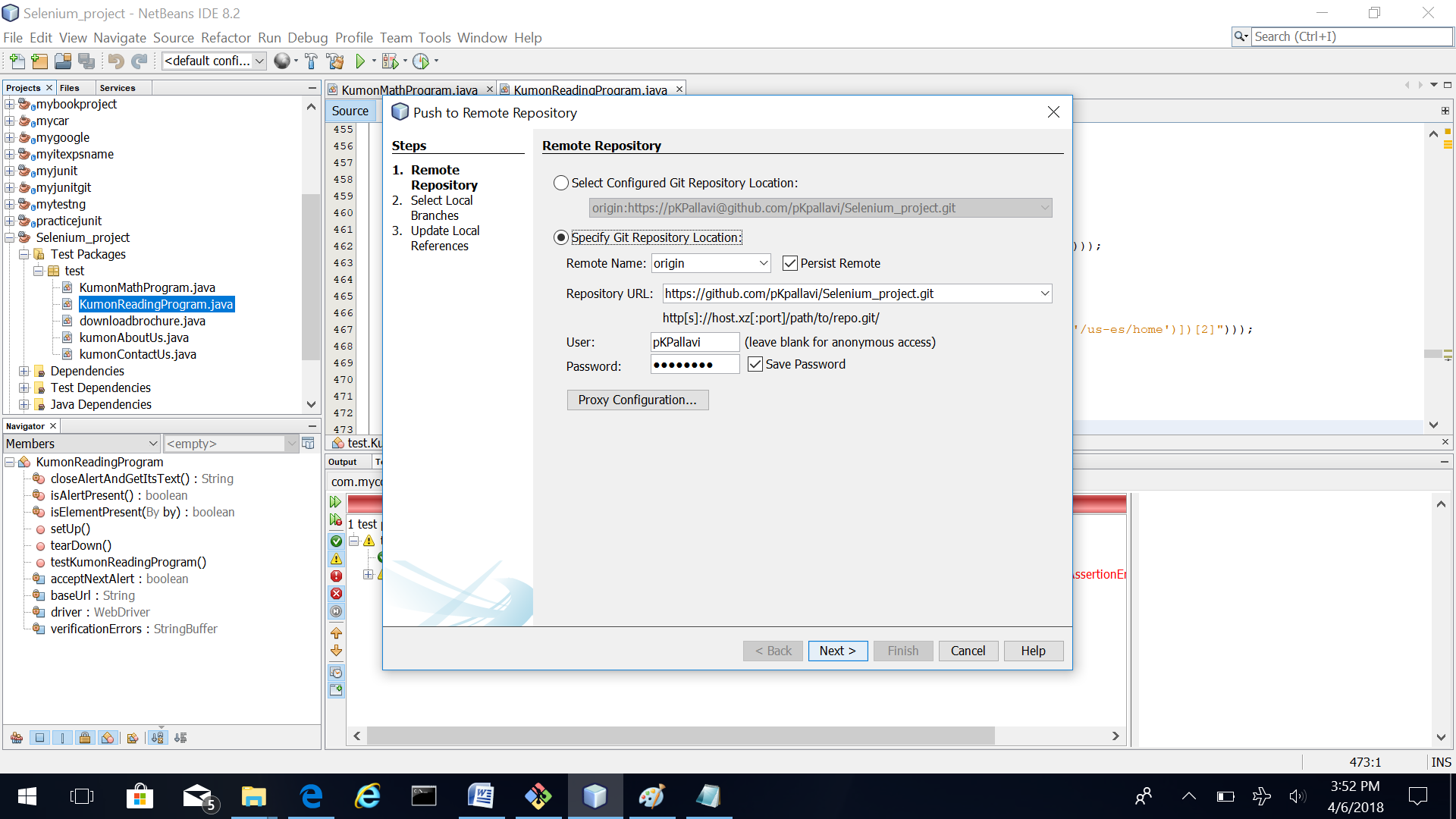
1. Select the branch you want to switch to. Provide the Branch Name in the box at the bottom. Press the Switch button. You can make your changes and additions in your branch and then check in code on GitHub through NetBeans on your branch. The online GitHub repository will be configured in future to automatically pull your changes from local branch to master branch. 
2. Verify that the Project Configuration is set to “test-compile surefire:test” by right-clicking on project “Set Configuration” -> “Customize”. Set Properties to “test=\*”. Click OK. 

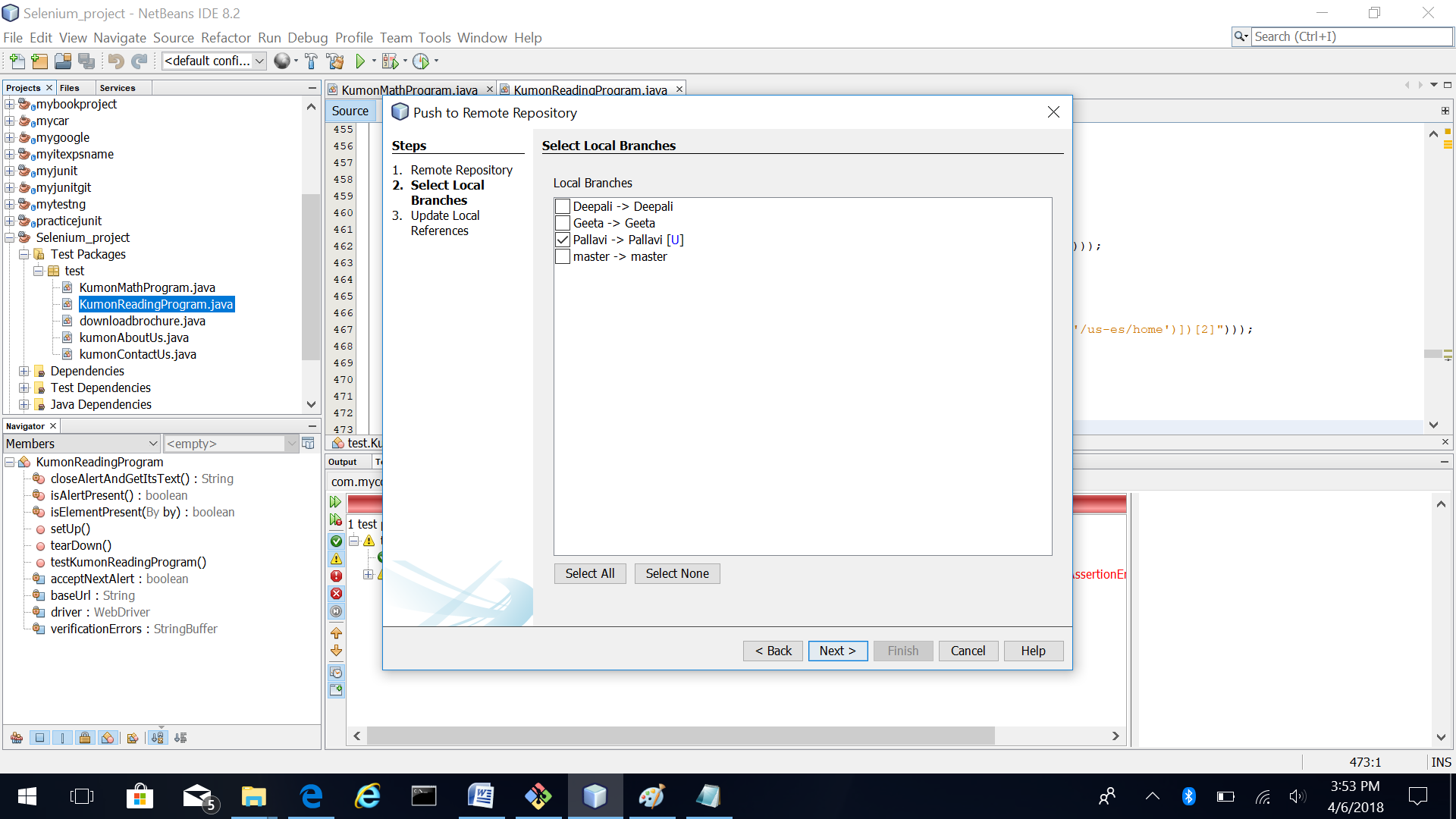


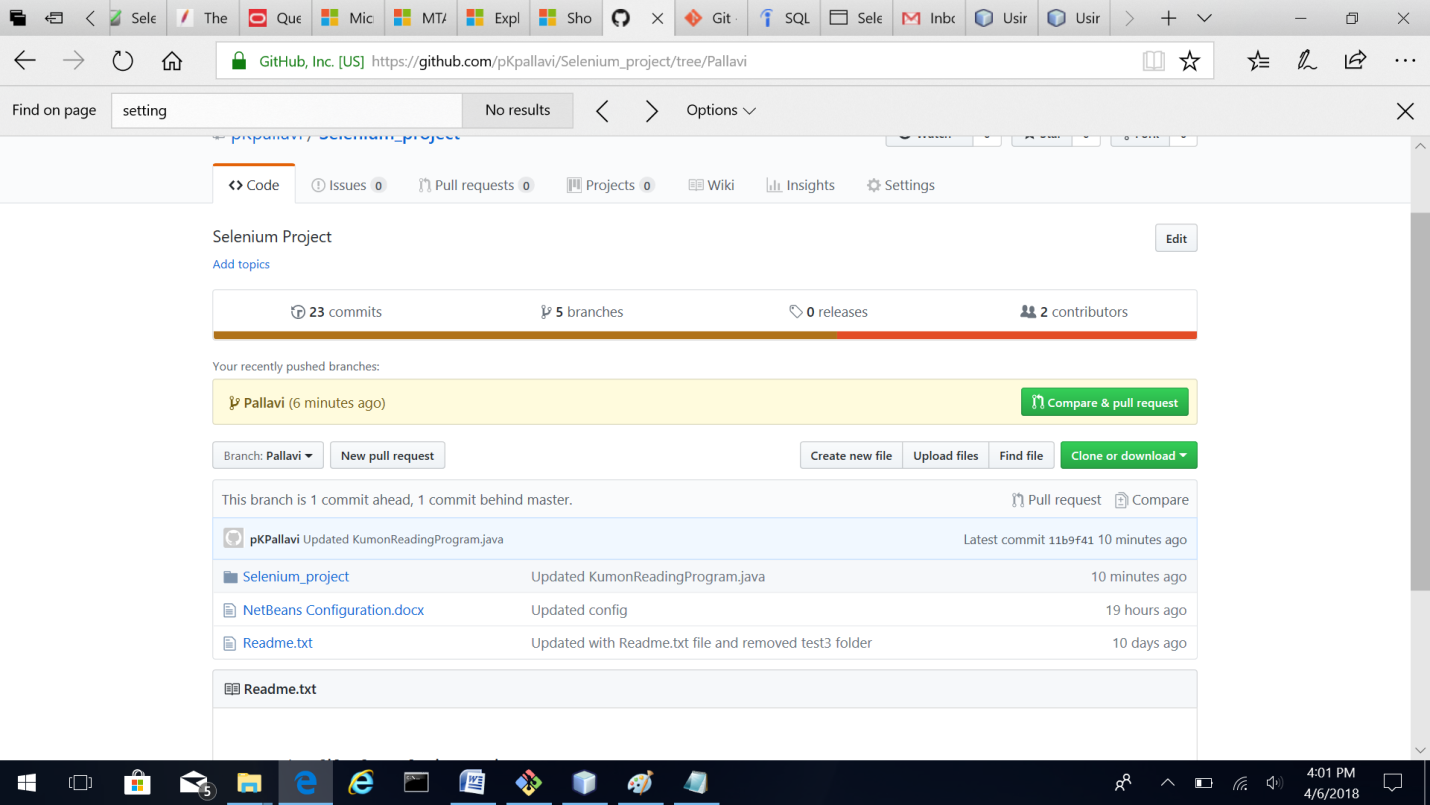
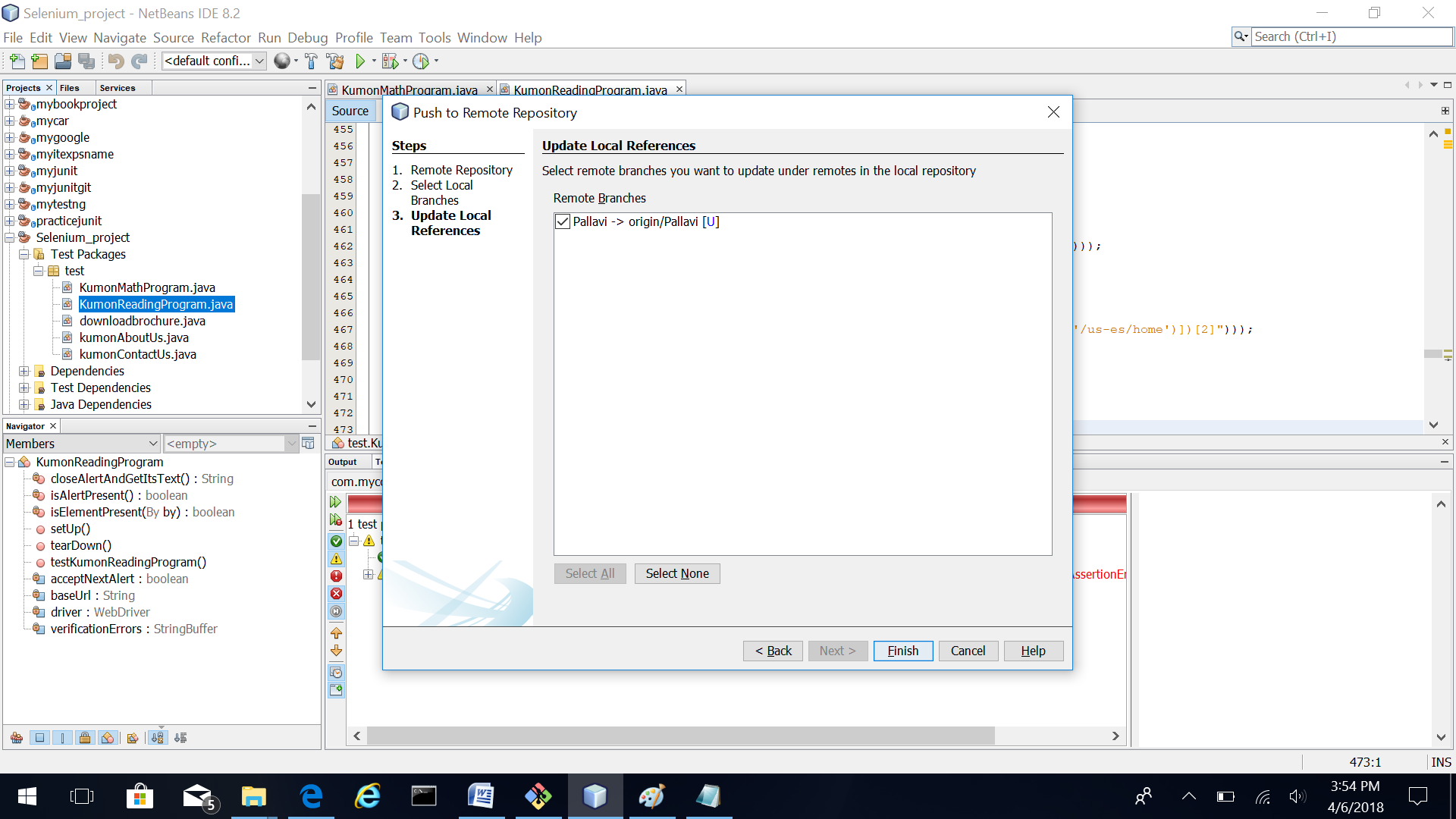
1. Right click on the project and run Test. 
2. Individual files can also be tested by right-clicking on the file and selecting “Test”. 
3. If you want to modify files, then Checkout Files as shown below.
4. Press Checkout. 
5. Modifications to file are shown in color. Addition is shown in green. File color also changes in the tab above to blue.
6. See the differences in file as shown below.



1. To add the modifications to the repository do Git Add as shown in snapshot below. 
2. Perform Git commit as shown in the snapshot. 
3. Perform Git Push as shown in the snapshot below. 



1. Select your branch to check in. Click Next.
2. Click Finish. Changes are pushed upstream and can be verified online in your branch.



# Using NetBeans IDE to build and test all Selenium TestNG files and using GitBash for accessing GitHub

These steps assume that there is already a repository in GitHub and these steps are to be followed to checkout and push updated code on local machine.

1. Get the files from GitHub.

Purvi@DESKTOP-6NDUR2E MINGW64 /c/Data/Selenium\_Project/Selenium\_project (Subarna)

$ git checkout Deepali

Switched to branch 'Deepali'

Purvi@DESKTOP-6NDUR2E MINGW64 /c/Data/Selenium\_Project/Selenium\_project (Deepali)

$ ls

Readme.txt

Purvi@DESKTOP-6NDUR2E MINGW64 /c/Data/Selenium\_Project/Selenium\_project (Deepali)

$ git pull

remote: Counting objects: 8, done.

remote: Compressing objects: 100% (7/7), done.

remote: Total 8 (delta 2), reused 7 (delta 1), pack-reused 0

Unpacking objects: 100% (8/8), done.

From https://github.com/pKpallavi/Selenium\_project

2619ab4..8a8eb11 Deepali -> origin/Deepali

ad7b409..775dc19 Subarna -> origin/Subarna

There is no tracking information for the current branch.

Please specify which branch you want to merge with.

See git-pull(1) for details.

git pull <remote> <branch>

If you wish to set tracking information for this branch you can do so with:

git branch --set-upstream-to=origin/<branch> Deepali

Purvi@DESKTOP-6NDUR2E MINGW64 /c/Data/Selenium\_Project/Selenium\_project (Deepali)

$ ls

Readme.txt

Purvi@DESKTOP-6NDUR2E MINGW64 /c/Data/Selenium\_Project/Selenium\_project (Deepali)

$ git branch --set-upstream-to=origin/Deepali Deepali

Branch Deepali set up to track remote branch Deepali from origin.

Purvi@DESKTOP-6NDUR2E MINGW64 /c/Data/Selenium\_Project/Selenium\_project (Deepali)

$ git pull

Updating 08c20ca..8a8eb11

Fast-forward

downloadbrochure.java | 79 +++++++++++++++++++++++++++++++++++++++++++++++++++

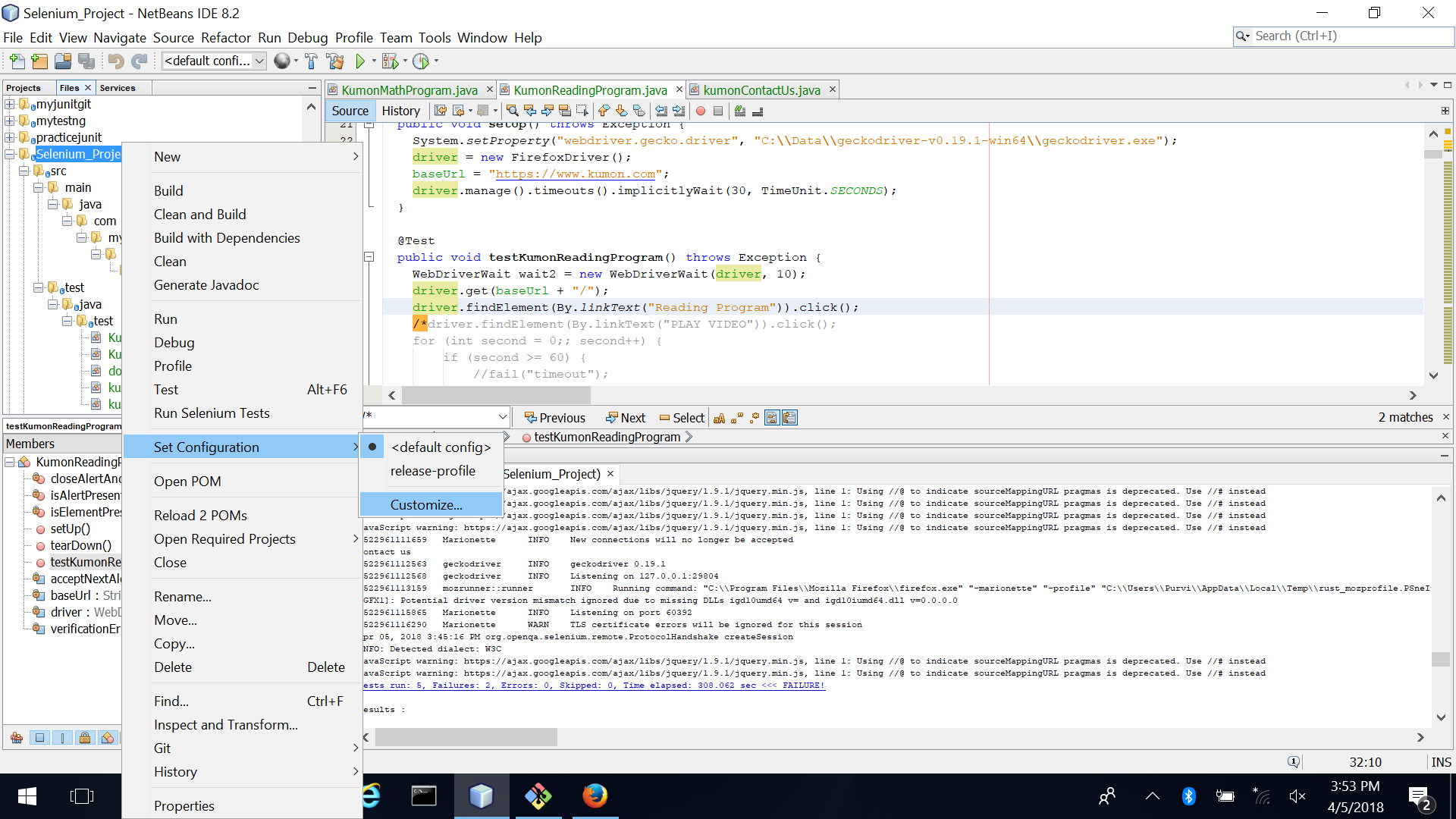
1 file changed, 79 insertions(+)

create mode 100644 downloadbrochure.java

Copy all Deepali’s files in the /c/Data/Selenium\_Project/Selenium\_project to C:\Users\Purvi\Documents\NetBeansProjects\Selenium\_Project\src\test\java\test folder

In same way set upstream to track another user’s branch to check out their files and copy in your NetBeans test folder. The files are automatically updated in NetBeans IDE as test folder is set in your IDE configuration.

1. Right click on the project in NetBeans. Select Customize option.



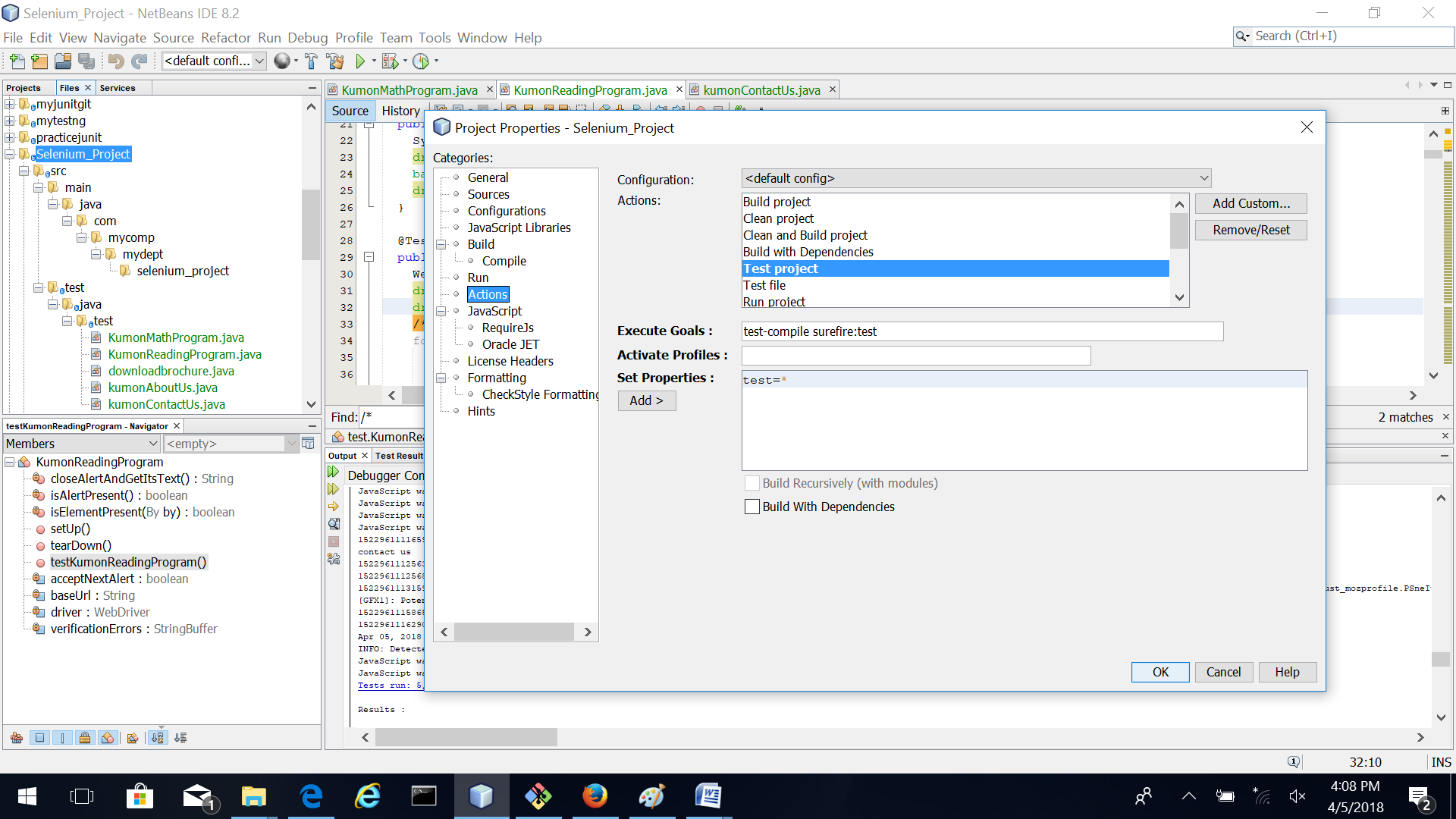
1. Select Actions and Test project in default config. Add goals as shown in snapshot below.

test-compile surefire:test

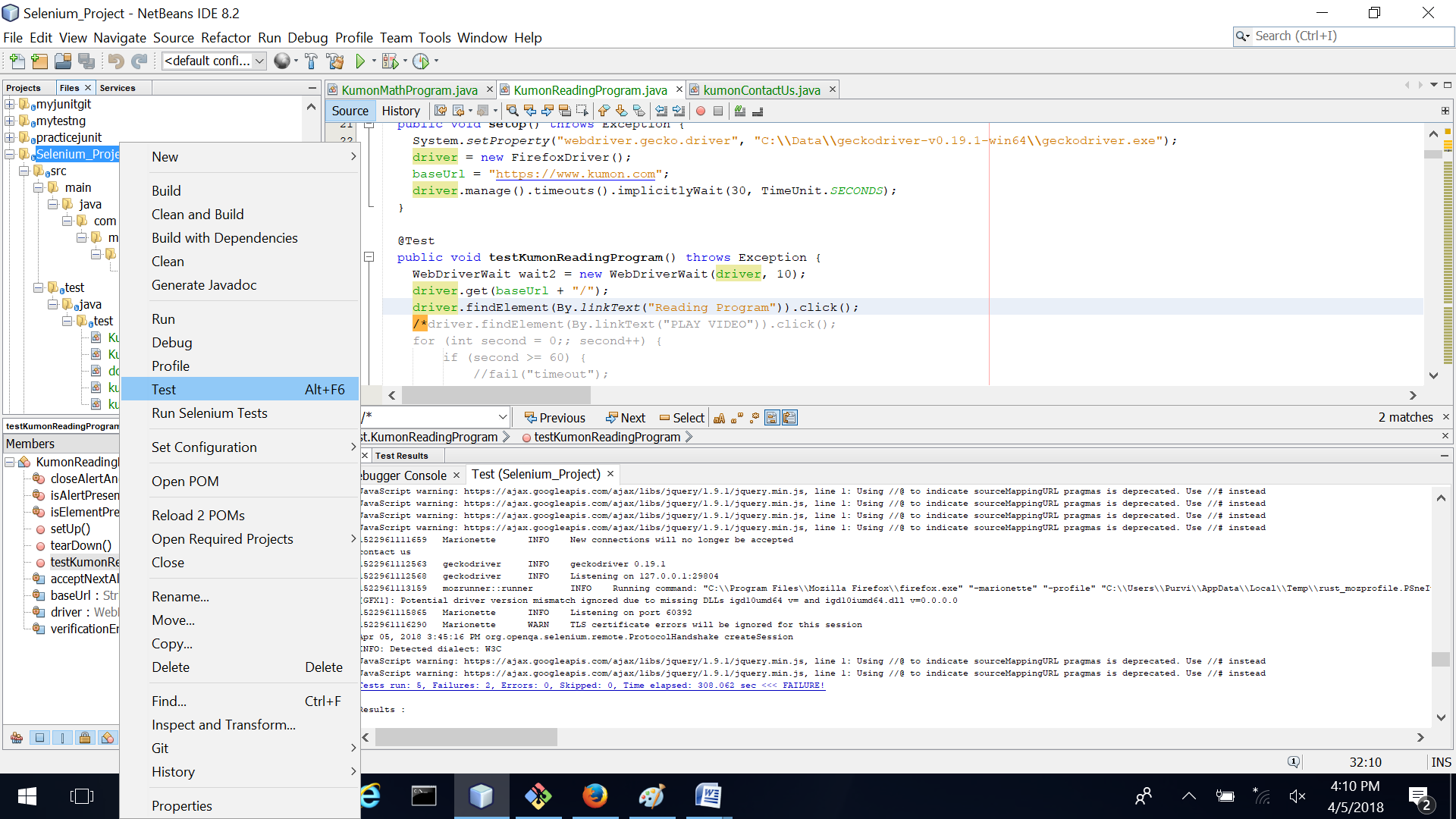
Set Properties as shown

test=\*

Click OK.



1. Right click on project and click Test



1. View the results in “Test Results” window. 