Java Assertions and Junit Tests

Assertions

For the Folio class an assertion was added in the ***addStock*** method to check that the stock that was just added does not already exist. An assertion was added to the ***buyStock*** method to check that the holding is >=0 after buying to indicate one has been bought. An assertion was added to the ***sellStock*** method to check that the stock no longer exists if all its shares have been sold. In the FolioTracker class an assertion was added to the method ***openFile*** to check if any two folios are the same after a folio has just been opened to make sure nothing is duplicated.

Junit Tests

Stock Class

The Stock test class has a coverage of 100%. Before each test a sample stock object is created with a sample TickerSymbol, name, pricePerShare, amount and change fields. A method ***stockGetterTest*** attempts to use each individual getter method, checking it against the expected sample field to ensure the correct details are returned. A second method ***stockSetterTest*** attempts to use each individual setter method. The first test attempts to lower the pricePerShare by using its setter, and checks if it has been lowered using its getter and checks if the change Boolean has been changed to false to indicate a decrease. The same is done for an increase in pricePerShare, with the change Boolean becoming true to indicate an increase. Tests are also included to set the number of shares to a value and then get them again to ensure the field was changed and to alter the change Boolean and check that it has been successfully changed by using its getter.

Folio Class

The Folio test class has a coverage of 96%. Before each test a sample folio object is created and three stocks with different sample fields are added to it. A method ***totalHoldingTest*** attempts to get the total holding of all the stocks in the folio, checking the result against the known value. The method ***getStockTest*** creates a sample stock object identical to one that’s already in the folio and uses it to check that the stock that “getStock” receives from the folio is the same for the given ticker symbol. The method ***buyStockTest*** attempts to buy shares for a stock already in the folio and for a stock not already in the folio to check if they succeed and fail respectively. The method ***sellStockTest*** attempts to sell stocks for a stock in the folio, checking that the result is true. A second test attempts to sell again for the same stock even though there are not enough shares to sell, this is to check if it throws a NotEnoughSharesException as expected. The method ***addStockTest*** attempts to add a stock that does not already exist, checking if it returns true. The method ***getNameTest*** calls the “getName” method with the given sample name of the folio to check if it has been named correctly.

FolioTracker Class

The FolioTracker test class has a coverage of 93%. Before each test a clean FolioTracker Object is created. The ***createFolioTest*** method attempts to create a folio with a sample name, attempts to “get” it with it’s given name and then attempts to create the same folio again. These three tests were employed to check that the folio is successfully created, that it has the correct name and that creating an identical folio is not allowed. The ***openFolioTest***method attempts to create a folio, add a stock to it and then save the folio. This folio is opened, and the contents of the stocks checked. This test is employed to check that the correct data is opened from a saved stock. This method also has two other tests to check that opening a secure file and opening an incorrect file fail as expected. The ***saveFolioTest*** method attempts to create a folio, add a stock to it and then save it to check that the save method returns true as expected. This method also attempts to save a folio that does not exist and to save a secure file folio without permission to see if these fail as expected. The ***deleteFolioTest*** method creates a folio, saves, deletes it and then checks that it no longer exists as expected. Another test attempts to delete a folio that doesn’t exist to check if it fails as expected while an additional test deletes a folio that was never saved to check that it still deletes as expected. The ***getFolioTest*** method creates a folio with a sample name and attempt to get it using its name to check it works successfully. It also attempts to get a folio that doesn’t exist ensuring that it fails as expected.