**Java Assertions**

StockListener

line 81, assertion used as we are only ever expecting there to be buy, add and sell actions and since we have already checked for buy and add, the last case must be sell.

PortfolioTracker

Line 62 – asserting that once the delete method has ran it will always be the case that the given folio name will not be not be a member of the portfolioList regardless if it was there in the first place or not.

Line 75 – similar assertion to above but in this case it will always be the case that by the end of the method a folio with the given name will belong to a folio within the tracker.

Line 82 - This assert is used for the creation of an object used later for an assertion.

Line 83 – this assertion is also used for setup for a later assertion

Line 99 – this assertion is used to check that the save method has not modified the list of portfolios while saving.

StockTable

Line 184 – checking that the updateField method is working by checking the field parameter is the same NumSharesField

Portfolio

Line 54 – this assertion is to create a copy of what the amount of shares before the operation is

Line 68 – used to call a private method which checks, given the amount before and the amount to sell that the number of shares after is valid.

Line 101 – testing the precondition of the private method createStock to see if our system passes this method any invalid tickers.

**Junit Tests**

There were 4 test classes created for each class within the model. They were PortfolioTest, PortfolioTrackerTest, PricesTest and StockTest. The tests within each class were used to assure that each class performed as expected. In Portfolio and Stock we override equals and hashcode so in order to test them effectively we test to the equals specification.

Junit Coverage

As can be seen in the screenshot below, this is the coverage of our model package.

