Instructions For Building and Running Software Application

Building the programme:

Please follow following instructions according to the order for completing the building phase.

1. Extract the supplied zip file to a convenient location. Then a folder called `Solution` will be created.

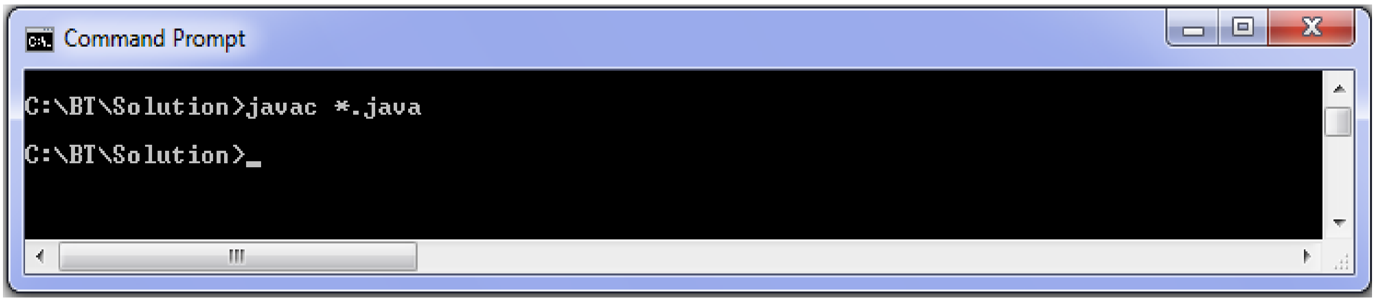
2. Open the command prompt or terminal and set the working directory to the `Solution` folder.

3. Then type `javac *.java` in the command prompt or terminal.

4. Please copy all the files containing word lists that you intend to use to current directory, in this case `Solution` directory.

5. Then the programme is ready for the execution with all the class files required to run the programme.

Following figure demonstrates the state of a windows command prompt after Building phase is completed.



```
C:\BT\Solution>javac *.java

C:\BT\Solution>_
```

Figure 1: State after completion of building phase

Running the programme:

The main execution point for the programme is provided through the `Program.class` file. For successful execution the programme requires two types of inputs.

- One or more surnames
- A list of surnames provided through the command line: This list of surnames could be directly taken from the command line/terminal or text file through input redirection.

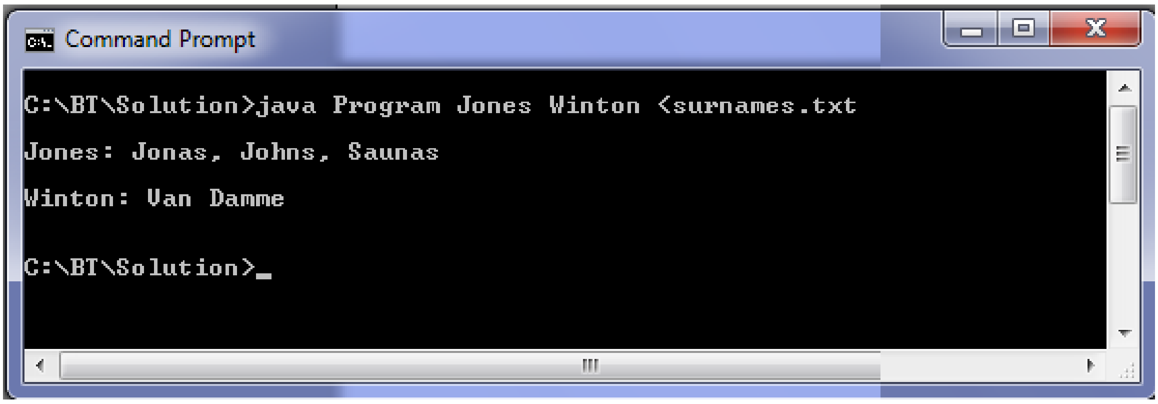
For each given surname output will be a set of surnames in the input list that phonetically match with the given surname.

Programme can be successfully executed through following below syntax.

```
java Program <one or more surnames> <Input List of names>
```

Case 1: Input List provided through the help of input redirection

Following figure clearly demonstrates the above situation.



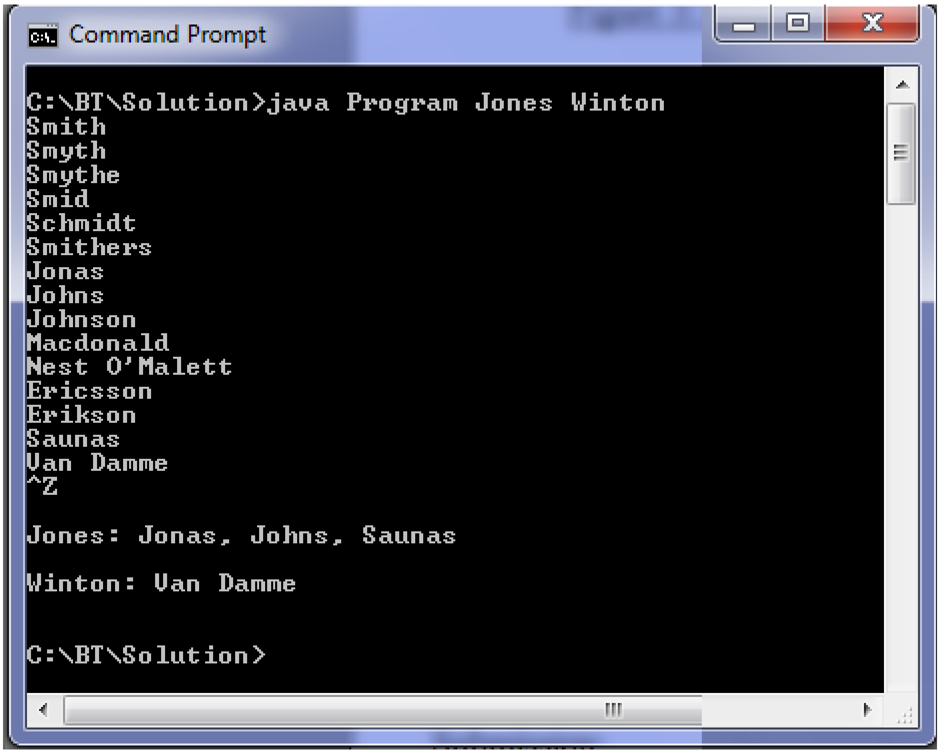
```
C:\BT\Solution>java Program Jones Winton <surnames.txt
Jones: Jonas, Johns, Saunas
Winton: Van Damme

C:\BT\Solution>_
```

Figure 2: Execution of the programme through input redirection

Case 2: List of surnames is provided only through the command line.

In this case the end of the standard input stream has to be conveyed to the programme through `Ctrl+Z` in windows environment or `Ctrl+D` in Linux environment. Following figure clearly demonstrates this case in a windows environment.



```
C:\BT\Solution>java Program Jones Winton
Smith
Smyth
Smythe
Smid
Schmidt
Smithers
Jonas
Johns
Johnson
Macdonald
Nest O'Malett
Ericsson
Erikson
Saunas
Van Damme
^Z

Jones: Jonas, Johns, Saunas

Winton: Van Damme


C:\BT\Solution>
```

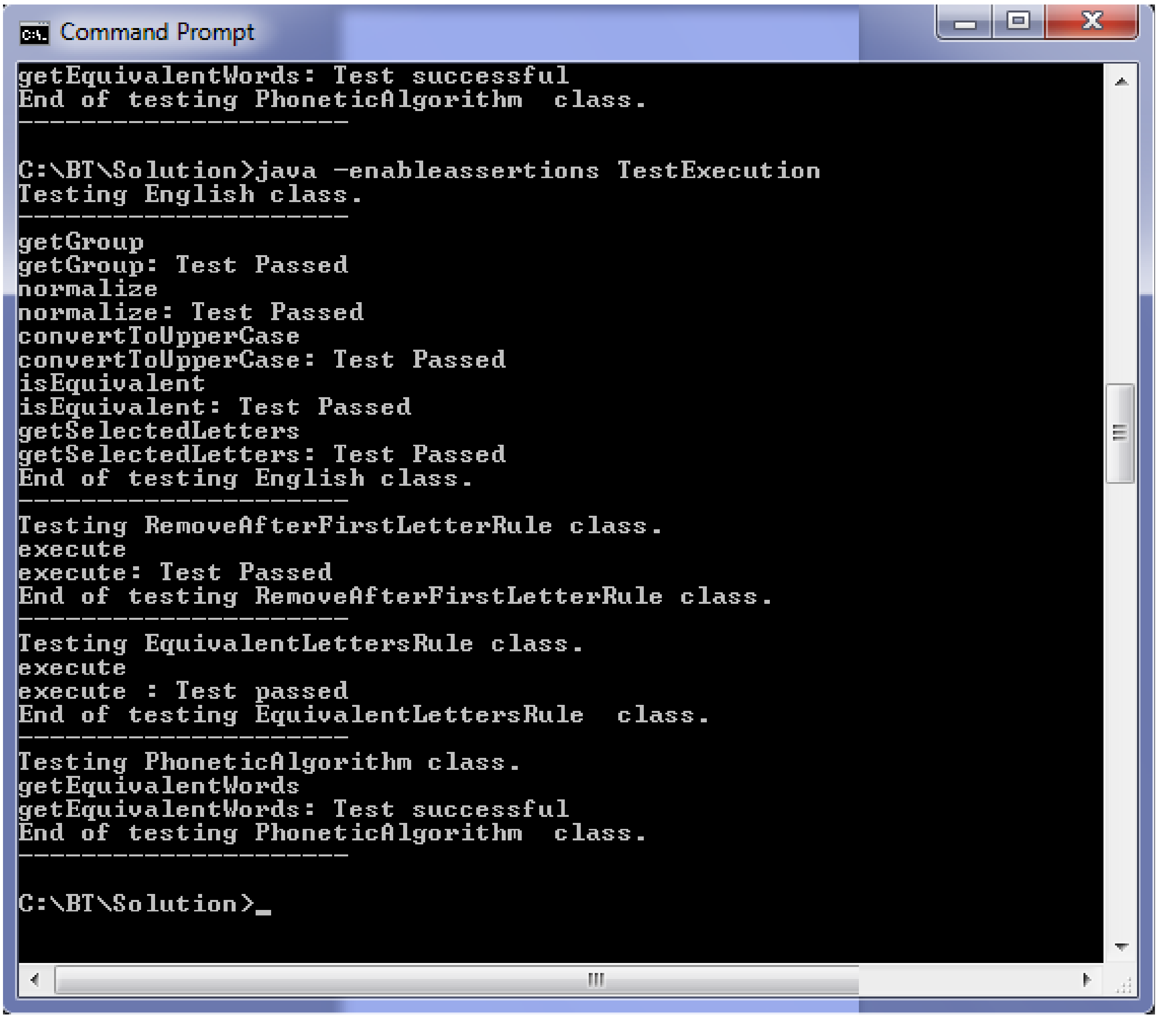
Figure 3: Execution of the programme through providing the input list solely from console.

Running test classes

Test cases could be executed using following command.

```
java -enableassertions TestExecution
```

Following figure gives a snapshot of me conducting unit tests on some of the classes.



```
getEquivalentWords: Test successful
End of testing PhoneticAlgorithm  class.
---------------------

C:\BT\Solution>java -enableassertions TestExecution
Testing English class.
---------------------
getGroup
getGroup: Test Passed
normalize
normalize: Test Passed
convertToUpperCase
convertToUpperCase: Test Passed
isEquivalent
isEquivalent: Test Passed
getSelectedLetters
getSelectedLetters: Test Passed
End of testing English class.
---------------------
Testing RemoveAfterFirstLetterRule class.
execute
execute: Test Passed
End of testing RemoveAfterFirstLetterRule class.
---------------------
Testing EquivalentLettersRule class.
execute
execute : Test passed
End of testing EquivalentLettersRule  class.
---------------------
Testing PhoneticAlgorithm class.
getEquivalentWords
getEquivalentWords: Test successful
End of testing PhoneticAlgorithm  class.
---------------------

C:\BT\Solution>_
```

Figure 4: Snapshot of test execution