**PROJECT**

Joseph Pickering

Student Number: 120181483

CSC1022 Programming II

Contents

[ArrayDirectory: 3](#_Toc346551880)

[addEntry(): 3](#_Toc346551889)

[deleteEntry(): 5](#_Toc346551890)

[lookupEntry(): 6](#_Toc346551893)

[changeTelephoneNumber(): 7](#_Toc346551905)

[toString(): 8](#_Toc346551915)

ListDirectory: [9](#_Toc346551889)

[addEntry(): 9](#_Toc346551890)

[deleteEntry(): 11](#_Toc346551893)

[lookupEntry(): 12](#_Toc346551905)

[changeTelephoneNumber(): 13](#_Toc346551915)

[toString(): 14](#_Toc346551905)

HashDirectory: [15](#_Toc346551889)

[addEntry(): 15](#_Toc346551890)

[deleteEntry(): 17](#_Toc346551893)

[lookupEntry(): 18](#_Toc346551905)

[changeTelephoneNumber(): 19](#_Toc346551915)

[toString(): 20](#_Toc346551905)

GUI: [21](#_Toc346551889)

[Inserting an entry: 21](#_Toc346551890)

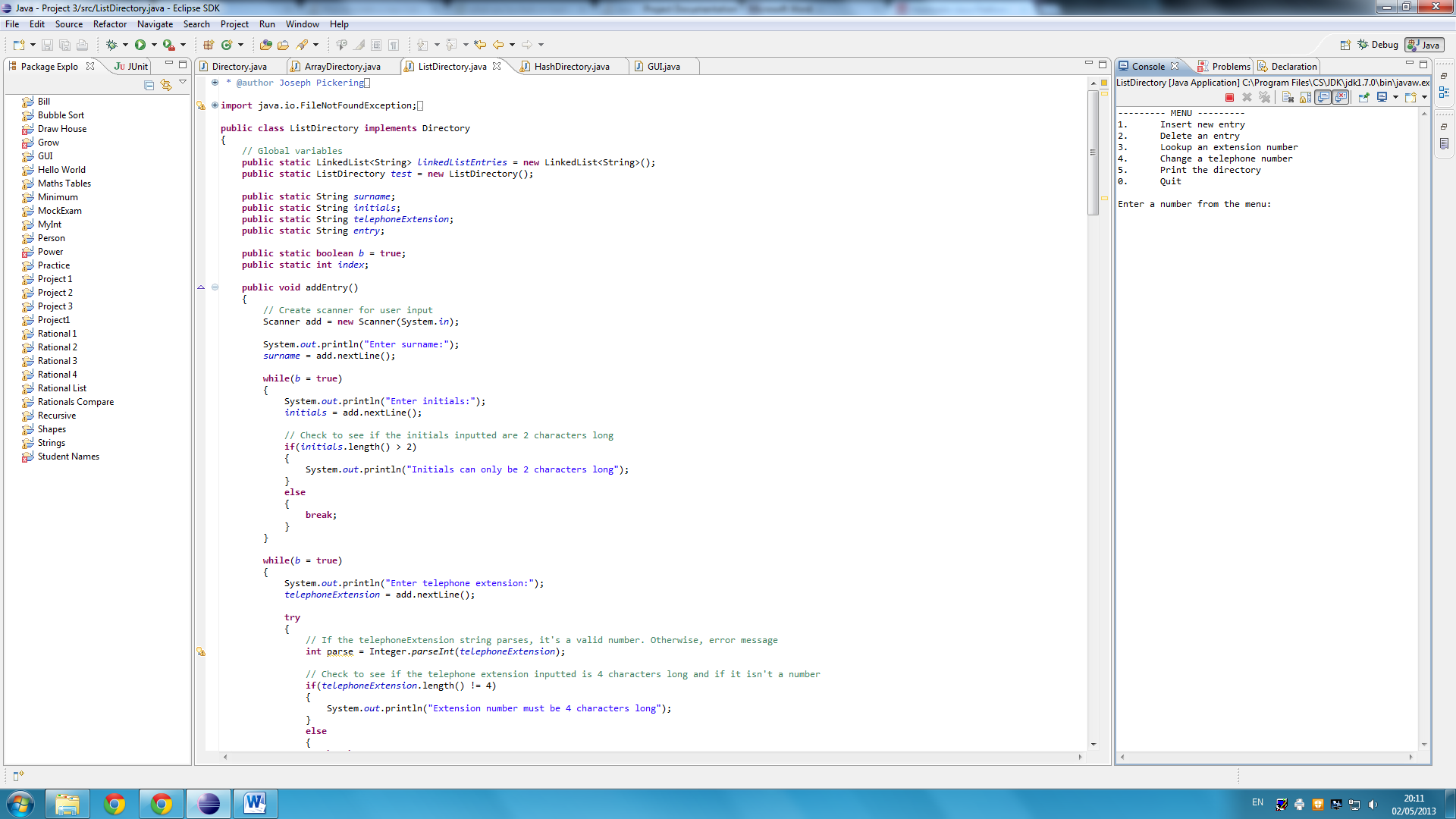
[Deleting an entry: 23](#_Toc346551893)

[Lookup an entry: 25](#_Toc346551905)

[Change extension number: 26](#_Toc346551915)

[Show directory: 27](#_Toc346551905)

**TESTING**

All three of the Directories implemented include a menu. This is for the user to choose what they’d like to do in the program. The menu looks like the following:

ArrayDirectory

addEntry():

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose of Test | Coverage | Test Values | Results Predicted | Results Actual |
| To see if the *addEntry()* method works correctly in adding a surname. | The *Directory* interface and the *ArrayDirectory* class are used. The main method and the *addEntry()* method will be used in this test. | Surname = Browning | For Enter initials: to appear | Enter initials: |
| To see if the *addEntry()* method works correctly in adding initials. | The *Directory* interface and the *ArrayDirectory* class are used. The main method and the *addEntry()* method will be used in this test. | Initials =  LB | For Enter telephone extension: to appear | Enter telephone extension: |
| To see if the *addEntry()* method works correctly in adding incorrect initials. | The *Directory* interface and the *ArrayDirectory* class are used. The main method and the *addEntry()* method will be used in this test. | Initials =  LBLBLB | For Initials can only be 2 characters long! appear | Initials can only be 2 characters long! |
| To see if the *addEntry()* method works correctly in adding a telephone extension. | The *Directory* interface and the *ArrayDirectory* class are used. The main method and the *addEntry()* method will be used in this test. | Telephone Extension = 1000 | For Entry added! to appear | Entry added! |
| To see if the *addEntry()* method works correctly in adding a word as a telephone extension. | The *Directory* interface and the *ArrayDirectory* class are used. The main method and the *addEntry()* method will be used in this test. | Telephone Extension = number | For Telephone extension can only be a number to appear | Telephone extension can only be a number |

The testing shows that the method works in preventing incorrect entries from being added to the directory and upon printing the menu after each of these tests, it showed that the method works well in adding the entries to the directory also.

deleteEntry():

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose of Test | Coverage | Test Values | Results Predicted | Results Actual |
| To see if the *deleteEntry()* method works in deleting an entry by entering a name. | The *Directory* interface and the *ArrayDirectory* class are used. The main method and the *deleteEntry()* method will be used in this test. | Name/Number =  name  Surname = Nolan | For Entry deleted! to appear and for the entry with surname Nolan to be removed | Entry deleted! |
| To see if the *deleteEntry()* method works in telling the user that they have entered and incorrect name. | The *Directory* interface and the *ArrayDirectory* class are used. The main method and the *deleteEntry()* method will be used in this test. | Name/Number =  name  Surname = Browning | For Invalid entry! to appear | Invalid entry! |
| To see if the *deleteEntry()* method works in deleting an entry by entering a telephone extension. | The *Directory* interface and the *ArrayDirectory* class are used. The main method and the *deleteEntry()* method will be used in this test. | Name/Number =  number  Telephone extension = 0009 | For Entry deleted! to appear and for the entry with telephone extension 0009 to be removed | Entry deleted! |
| To see if the *deleteEntry()* method works in telling the user that they have entered and incorrect telephone extension. | The *Directory* interface and the *ArrayDirectory* class are used. The main method and the *deleteEntry()* method will be used in this test. | Name/Number =  number  Telephone extension = 1000 | For Invalid entry! to appear | Invalid entry! |

The testing shows that the method works in preventing incorrect inputs from being removed from the directory and upon printing the menu after each of these tests, it showed that the method works well in removing correct entries to the directory also.

lookupEntry():

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose of Test | Coverage | Test Values | Results Predicted | Results Actual |
| To see if the *lookupEntry()* method works in finding an entry. | The *Directory* interface and the *ArrayDirectory* class are used. The main method and the *lookupEntry()* method will be used in this test. | Surname = Zorro | For Extension number for Zorro is: 0026 to appear | Extension number for Zorro is: 0026 |
| To see if the *lookupEntry()* method works in finding an entry. | The *Directory* interface and the *ArrayDirectory* class are used. The main method and the *lookupEntry()* method will be used in this test. | Surname = Browning | For Invalid surname! to appear | Invalid surname! |
| To see if the *lookupEntry()* method works efficiently in finding an entry. | The *Directory* interface and the *ArrayDirectory* class are used. The main method and the *lookupEntry()* method will be used in this test. | Surname = Zorro | For the method to lookup an entry 10000 times in about 150 milliseconds | Time:  136 |

The testing shows that the lookupEntry() method works correctly and efficiently in finding the extension number for an entry in the directory.

changeEntryNumber():

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose of Test | Coverage | Test Values | Results Predicted | Results Actual |
| To see if the *changeEntryNumber()* method works in changing the extension number for an entry in the directory. | The *Directory* interface and the *ArrayDirectory* class are used. The main method and the *changeEntryNumber()* method will be used in this test. | Surname = Davies  Extension Number = 1000 | For Extension number for Davies is: 0004  What would you like to change it to? to appear.  Then for Extension number changed! to appear | Extension number for Davies is: 0004  What would you like to change it to?  Extension number changed! |
| To see if the *changeEntryNumber()* method works in informing the user that an incorrect surname was inputted. | The *Directory* interface and the *ArrayDirectory* class are used. The main method and the *changeEntryNumber()* method will be used in this test. | Surname = Browning | For Invalid surname! to appear. | Extension number for Cook is: 0005  What would you like to change it to? |
| To see if the *changeEntryNumber()* method works in informing the user that an incorrect telephone extension was inputted. | The *Directory* interface and the *ArrayDirectory* class are used. The main method and the *changeEntryNumber()* method will be used in this test. | Surname = Davies  Extension Number = 10001000 | For Extension number for Davies is: 0004  What would you like to change it to? to appear.  Then for Extension number must be 4 characters long to appear | Extension number for Davies is: 0004  What would you like to change it to?  Extension number must be 4 characters long |
| To see if the *changeEntryNumber()* method works in informing the user that an incorrect telephone extension was inputted. | The *Directory* interface and the *ArrayDirectory* class are used. The main method and the *changeEntryNumber()* method will be used in this test. | Surname = Davies  Extension Number = Joseph | For Extension number for Davies is: 0004  What would you like to change it to? to appear.  Then for Extension number must be a number to appear | Extension number for Davies is: 0004  What would you like to change it to?  Extension number must be a number |

The testing shows that the changeEntryNumber() method works well in changing the extension number of an entry. However, there is a bug upon entering an incorrect surname which is sometimes recognised as one in the directory.

toString():

|  |  |  |
| --- | --- | --- |
| Purpose of Test | Coverage | Test Values |
| To see if the *toString()* method print the directory in a neat tabulated fashion. | The *Directory* interface and the *ArrayDirectory* class are used. The main method and the *toString()* method will be used in this test. | Menu =  5 |

|  |  |
| --- | --- |
| Results predicted | Results actual |
| --------- Directory ---------  Surname Initials Extension Number  Andrews AA 0001  Brown BB 0002  Cook CC 0003  Davies DD 0004  Evans EE 0005  Fox FF 0006  Green GG 0007  Hughes HH 0008  Ireland II 0009  Jones JJ 0010  King KK 0011  Lee LL 0012  Moore MM 0013  Nolan NN 0014  Owen OO 0015  Parker PP 0016  Quinn QQ 0017  Roberts RR 0018  Smith SS 0019  Taylor TT 0020  Upson UU 0021  Vine VV 0022  Wood WW 0023  Young YY 0025  Zorro ZZ 0026 | --------- Directory ---------  Surname Initials Extension Number  Andrews AA 0001  Brown BB 0002  Cook CC 0003  Davies DD 0004  Evans EE 0005  Fox FF 0006  Green GG 0007  Hughes HH 0008  Ireland II 0009  Jones JJ 0010  King KK 0011  Lee LL 0012  Moore MM 0013  Nolan NN 0014  Owen OO 0015  Parker PP 0016  Quinn QQ 0017  Roberts RR 0018  Smith SS 0019  Taylor TT 0020  Upson UU 0021  Vine VV 0022  Wood WW 0023  Young YY 0025  Zorro ZZ 0026 |

This test shows that the *toString()* method works well in printing the directory in a neat tabulated fashion.

ListDirectory

addEntry():

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose of Test | Coverage | Test Values | Results Predicted | Results Actual |
| To see if the *addEntry()* method works correctly in adding a surname. | The *Directory* interface and the *ListDirectory* class are used. The main method and the *addEntry()* method will be used in this test. | Surname = Browning | For Enter initials: to appear | Enter initials: |
| To see if the *addEntry()* method works correctly in adding initials. | The *Directory* interface and the *ListDirectory* class are used. The main method and the *addEntry()* method will be used in this test. | Initials =  LB | For Enter telephone extension: to appear | Enter telephone extension: |
| To see if the *addEntry()* method works correctly in adding incorrect initials. | The *Directory* interface and the *ListDirectory* class are used. The main method and the *addEntry()* method will be used in this test. | Initials =  LBLBLB | For to Initials can only be 2 characters long! appear | Initials can only be 2 characters long! |
| To see if the *addEntry()* method works correctly in adding a telephone extension. | The *Directory* interface and the *ListDirectory* class are used. The main method and the *addEntry()* method will be used in this test. | Telephone Extension = 1000 | For Entry added! to appear | Entry added! |
| To see if the *addEntry()* method works correctly in adding a word as a telephone extension. | The *Directory* interface and the *ListDirectory* class are used. The main method and the *addEntry()* method will be used in this test. | Telephone Extension = number | For Telephone extension can only be a number to appear | Telephone extension can only be a number |

The testing shows that the method works in preventing incorrect entries from being added to the directory and upon printing the menu after each of these tests, it showed that the method works well in adding the entries to the directory also.

deleteEntry():

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose of Test | Coverage | Test Values | Results Predicted | Results Actual |
| To see if the *deleteEntry()* method works in deleting an entry by entering a name. | The *Directory* interface and the *ListDirectory* class are used. The main method and the *deleteEntry()* method will be used in this test. | Name/Number =  name  Surname = Nolan | For Entry deleted! to appear and for the entry with surname Nolan to be removed | Entry deleted! |
| To see if the *deleteEntry()* method works in telling the user that they have entered and incorrect name. | The *Directory* interface and the *ListDirectory* class are used. The main method and the *deleteEntry()* method will be used in this test. | Name/Number =  name  Surname = Browning | For Invalid entry! to appear | Invalid entry! |
| To see if the *deleteEntry()* method works in deleting an entry by entering a telephone extension. | The *Directory* interface and the *ListDirectory* class are used. The main method and the *deleteEntry()* method will be used in this test. | Name/Number =  number  Telephone extension = 0009 | For Entry deleted! to appear and for the entry with extension number 0009 to be removed | Entry deleted! |
| To see if the *deleteEntry()* method works in telling the user that they have entered and incorrect telephone extension. | The *Directory* interface and the *ListDirectory* class are used. The main method and the *deleteEntry()* method will be used in this test. | Name/Number =  number  Telephone extension = 1000 | For Invalid entry! to appear | Invalid entry! |

The testing shows that the method works in preventing incorrect inputs from being removed from the directory and upon printing the menu after each of these tests, it showed that the method works well in removing correct entries to the directory also.

lookupEntry():

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose of Test | Coverage | Test Values | Results Predicted | Results Actual |
| To see if the *lookupEntry()* method works in finding an entry. | The *Directory* interface and the *ListDirectory* class are used. The main method and the *lookupEntry()* method will be used in this test. | Surname = Zorro | For Extension number for Zorro is: 0026 to appear | Extension number for Zorro is: 0026 |
| To see if the *lookupEntry()* method works in finding an entry. | The *Directory* interface and the *ListDirectory* class are used. The main method and the *lookupEntry()* method will be used in this test. | Surname = Browning | For Invalid surname! to appear | Invalid surname! |
| To see if the *lookupEntry()* method works efficiently in finding an entry. | The *Directory* interface and the *ListDirectory* class are used. The main method and the *lookupEntry()* method will be used in this test. | Surname = Zorro | For the method to lookup an entry 10000 times in about 150 milliseconds | Time:  144 |

The testing shows that the lookupEntry() method works correctly and efficiently in finding the extension number for an entry in the directory.

changeEntryNumber():

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose of Test | Coverage | Test Values | Results Predicted | Results Actual |
| To see if the *changeEntryNumber()* method works in changing the extension number for an entry in the directory. | The *Directory* interface and the *ListDirectory* class are used. The main method and the *changeEntryNumber()* method will be used in this test. | Surname = Davies  Extension Number = 1000 | For Extension number for Davies is: 0004  What would you like to change it to? to appear.  Then for Extension number changed! to appear | Extension number for Davies is: 0004  What would you like to change it to?  Extension number changed! |
| To see if the *changeEntryNumber()* method works in informing the user that an incorrect surname was inputted. | The *Directory* interface and the *ListDirectory* class are used. The main method and the *changeEntryNumber()* method will be used in this test. | Surname = Browning | For Invalid surname! to appear. | Invalid surname! |
| To see if the *changeEntryNumber()* method works in informing the user that an incorrect telephone extension was inputted. | The *Directory* interface and the *ListDirectory* class are used. The main method and the *changeEntryNumber()* method will be used in this test. | Surname = Davies  Extension Number = 10001000 | For Extension number for Davies is: 0004  What would you like to change it to? to appear.  Then for Extension number must be 4 characters long to appear | Extension number for Davies is: 0004  What would you like to change it to?  Extension number must be 4 characters long |
| To see if the *changeEntryNumber()* method works in informing the user that an incorrect telephone extension was inputted. | The *Directory* interface and the *ListDirectory* class are used. The main method and the *changeEntryNumber()* method will be used in this test. | Surname = Davies  Extension Number = Joseph | For Extension number for Davies is: 0004  What would you like to change it to? to appear.  Then for Extension number must be a number to appear | Extension number for Davies is: 0004  What would you like to change it to?  Extension number must be a number |

The testing shows that the changeEntryNumber() method works well in changing the extension number of an entry and also works in preventing the user from inputting incorrect surnames and telephone extensions.

toString():

|  |  |  |
| --- | --- | --- |
| Purpose of Test | Coverage | Test Values |
| To see if the *toString()* method print the directory in a neat tabulated fashion. | The *Directory* interface and the *ListDirectory* class are used. The main method and the *toString()* method will be used in this test. | Menu =  5 |

|  |  |
| --- | --- |
| Results predicted | Results actual |
| --------- Directory ---------  Surname Initials Extension Number  Andrews AA 0001  Brown BB 0002  Cook CC 0003  Davies DD 0004  Evans EE 0005  Fox FF 0006  Green GG 0007  Hughes HH 0008  Ireland II 0009  Jones JJ 0010  King KK 0011  Lee LL 0012  Moore MM 0013  Nolan NN 0014  Owen OO 0015  Parker PP 0016  Quinn QQ 0017  Roberts RR 0018  Smith SS 0019  Taylor TT 0020  Upson UU 0021  Vine VV 0022  Wood WW 0023  Young YY 0025  Zorro ZZ 0026 | --------- Directory ---------  Surname Initials Extension Number  Andrews AA 0001  Brown BB 0002  Cook CC 0003  Davies DD 0004  Evans EE 0005  Fox FF 0006  Green GG 0007  Hughes HH 0008  Ireland II 0009  Jones JJ 0010  King KK 0011  Lee LL 0012  Moore MM 0013  Nolan NN 0014  Owen OO 0015  Parker PP 0016  Quinn QQ 0017  Roberts RR 0018  Smith SS 0019  Taylor TT 0020  Upson UU 0021  Vine VV 0022  Wood WW 0023  Young YY 0025  Zorro ZZ 0026 |

This test shows that the *toString()* method works well in printing the directory in a neat tabulated fashion.

HashDirectory

addEntry():

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose of Test | Coverage | Test Values | Results Predicted | Results Actual |
| To see if the *addEntry()* method works correctly in adding a surname. | The *Directory* interface and the *HashDirectory* class are used. The main method and the *addEntry()* method will be used in this test. | Surname = Browning | For Enter initials: to appear | Enter initials: |
| To see if the *addEntry()* method works correctly in adding initials. | The *Directory* interface and the *HashDirectory* class are used. The main method and the *addEntry()* method will be used in this test. | Initials =  LB | For Enter telephone extension: to appear | Enter telephone extension: |
| To see if the *addEntry()* method works correctly in adding incorrect initials. | The *Directory* interface and the *HashDirectory* class are used. The main method and the *addEntry()* method will be used in this test. | Initials =  LBLBLB | For to Initials can only be 2 characters long! appear | Initials can only be 2 characters long! |
| To see if the *addEntry()* method works correctly in adding a telephone extension. | The *Directory* interface and the Hash*Directory* class are used. The main method and the *addEntry()* method will be used in this test. | Telephone Extension = 1000 | For Entry added! to appear | Entry added! |
| To see if the *addEntry()* method works correctly in adding a word as a telephone extension. | The *Directory* interface and the *HashDirectory* class are used. The main method and the *addEntry()* method will be used in this test. | Telephone Extension = number | For Telephone extension can only be a number to appear | Telephone extension can only be a number |

The testing shows that the method works in preventing incorrect entries from being added to the directory and upon printing the menu after each of these tests, it showed that the method works well in adding the entries to the directory also.

deleteEntry():

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose of Test | Coverage | Test Values | Results Predicted | Results Actual |
| To see if the *deleteEntry()* method works in deleting an entry by entering a name. | The *Directory* interface and the *HashDirectory* class are used. The main method and the *deleteEntry()* method will be used in this test. | Name/Number =  name  Surname = Nolan | For Entry deleted! to appear and for the entry with surname Nolan to be removed | Entry deleted! |
| To see if the *deleteEntry()* method works in telling the user that they have entered and incorrect name. | The *Directory* interface and the *HashDirectory* class are used. The main method and the *deleteEntry()* method will be used in this test. | Name/Number =  name  Surname = Browning | For Invalid entry! to appear | Invalid entry! |
| To see if the *deleteEntry()* method works in deleting an entry by entering a telephone extension. | The *Directory* interface and the *HashDirectory* class are used. The main method and the *deleteEntry()* method will be used in this test. | Name/Number =  number  Telephone extension = 0009 | For Entry deleted! to appear and for the entry with extension number 0009 to be removed | Entry deleted! |
| To see if the *deleteEntry()* method works in telling the user that they have entered and incorrect telephone extension. | The *Directory* interface and the *HashDirectory* class are used. The main method and the *deleteEntry()* method will be used in this test. | Name/Number =  number  Telephone extension = 1000 | For Invalid entry! to appear | Invalid entry! |

The testing shows that the method works in preventing incorrect inputs from being removed from the directory and upon printing the menu after each of these tests, it showed that the method works well in removing correct entries to the directory also.

lookupEntry():

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose of Test | Coverage | Test Values | Results Predicted | Results Actual |
| To see if the *lookupEntry()* method works in finding an entry. | The *Directory* interface and the *HashDirectory* class are used. The main method and the *lookupEntry()* method will be used in this test. | Surname = Zorro | For Extension number for Zorro is: 0026 to appear | Extension number for Zorro is: 0026 |
| To see if the *lookupEntry()* method works in finding an entry. | The *Directory* interface and the *HashDirectory* class are used. The main method and the *lookupEntry()* method will be used in this test. | Surname = Browning | For Invalid surname! to appear | Invalid surname! |
| To see if the *lookupEntry()* method works efficiently in finding an entry. | The *Directory* interface and the *HashDirectory* class are used. The main method and the *lookupEntry()* method will be used in this test. | Surname = Zorro | For the method to lookup an entry 10000 times in about 150 milliseconds | Time:  121 |

The testing shows that the lookupEntry() method works correctly and efficiently in finding the extension number for an entry in the directory.

changeEntryNumber():

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose of Test | Coverage | Test Values | Results Predicted | Results Actual |
| To see if the *changeEntryNumber()* method works in changing the extension number for an entry in the directory. | The *Directory* interface and the *HashDirectory* class are used. The main method and the *changeEntryNumber()* method will be used in this test. | Surname = Davies  Extension Number = 1000 | For Extension number for Davies is: 0004  What would you like to change it to? to appear.  Then for Extension number changed! to appear | Extension number for Davies is: 0004  What would you like to change it to?  Extension number changed! |
| To see if the *changeEntryNumber()* method works in informing the user that an incorrect surname was inputted. | The *Directory* interface and the *HashDirectory* class are used. The main method and the *changeEntryNumber()* method will be used in this test. | Surname = Browning | For Invalid surname! to appear. | Invalid surname! |
| To see if the *changeEntryNumber()* method works in informing the user that an incorrect telephone extension was inputted. | The *Directory* interface and the *HashDirectory* class are used. The main method and the *changeEntryNumber()* method will be used in this test. | Surname = Davies  Extension Number = 10001000 | For Extension number for Davies is: 0004  What would you like to change it to? to appear.  Then for Extension number must be 4 characters long to appear | Extension number for Davies is: 0004  What would you like to change it to?  Extension number must be 4 characters long |
| To see if the *changeEntryNumber()* method works in informing the user that an incorrect telephone extension was inputted. | The *Directory* interface and the *HashDirectory* class are used. The main method and the *changeEntryNumber()* method will be used in this test. | Surname = Davies  Extension Number = Joseph | For Extension number for Davies is: 0004  What would you like to change it to? to appear.  Then for Extension number must be a number to appear | Extension number for Davies is: 0004  What would you like to change it to?  Extension number must be a number |

The testing shows that the changeEntryNumber() method works well in changing the extension number of an entry and also works in preventing the user from inputting incorrect surnames and telephone extensions.

toString():

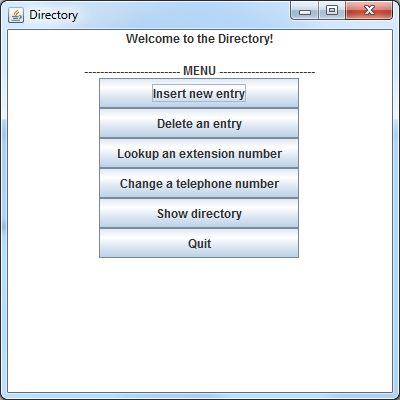
|  |  |  |
| --- | --- | --- |
| Purpose of Test | Coverage | Test Values |
| To see if the *toString()* method print the directory in a neat tabulated fashion. | The *Directory* interface and the *HashDirectory* class are used. The main method and the *toString()* method will be used in this test. | Menu =  5 |

|  |  |
| --- | --- |
| Results predicted | Results actual |
| --------- Directory ---------  Surname Initials Extension Number  Andrews AA 0001  Brown BB 0002  Cook CC 0003  Davies DD 0004  Evans EE 0005  Fox FF 0006  Green GG 0007  Hughes HH 0008  Ireland II 0009  Jones JJ 0010  King KK 0011  Lee LL 0012  Moore MM 0013  Nolan NN 0014  Owen OO 0015  Parker PP 0016  Quinn QQ 0017  Roberts RR 0018  Smith SS 0019  Taylor TT 0020  Upson UU 0021  Vine VV 0022  Wood WW 0023  Young YY 0025  Zorro ZZ 0026 | --------- Directory ---------  Surname Initials Extension Number  Zorro ZZ 0026  Young YY 0025  Wood WW 0023  Vine VV 0022  Upson UU 0021  Taylor TT 0020  Smith SS 0019  Roberts RR 0018  Quinn QQ 0017  Parker PP 0016  Owen OO 0015  Nolan NN 0014  Moore MM 0013  Lee LL 0012  King KK 0011  Jones JJ 0010  Ireland II 0009  Hughes HH 0008  Green GG 0007  Fox FF 0006  Evans EE 0005  Davies DD 0004  Cook CC 0003  Brown BB 0002  Andrews AA 0001 |

This test shows that the *toString()* method works well in printing the directory in a neat tabulated fashion but doesn’t print it in order.

GUI

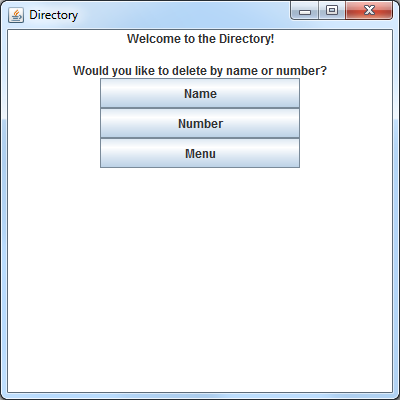
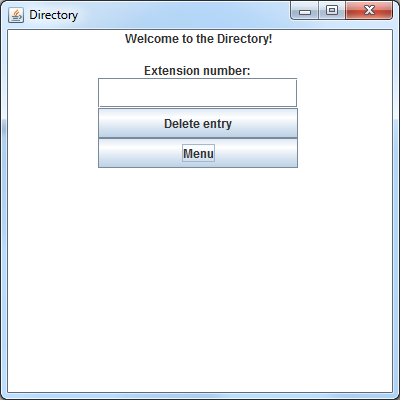
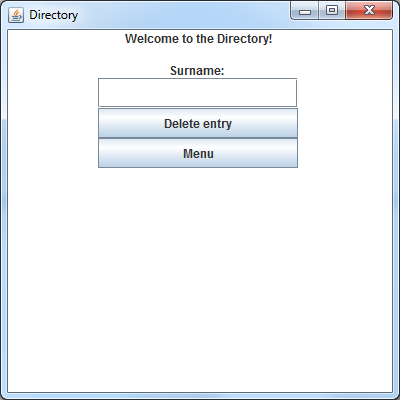
The class that I decided to implement a graphical user interface with was the ListDirectory class. The screen when you run the program looks like the following:



Inserting an entry:

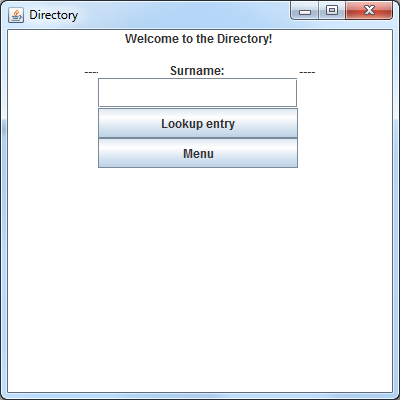
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose of Test | Coverage | Test Values | Results Predicted | Results Actual |
| To see if the *Insert new entry* section of the GUI works correctly | The *Directory* interface, *ListDirectory* class and the GUI class are used. | Surname = Browningson | For a message box with Surnames must be between 1 and characters long to appear | Surnames must be between 1 and characters long |
| To see if the *Insert new entry* section of the GUI works correctly | The *Directory* interface, *ListDirectory* class and the GUI class are used. | Surname = Browning  Initials = BLBLBL | For a message box with Initials must be 2 characters long to appear | Initials must be 2 characters long |
| To see if the *Insert new entry* section of the GUI works correctly | The *Directory* interface, *ListDirectory* class and the GUI class are used. | Surname = Browning  Initials =  BL  Extension number =  10001000 | For a message box with Extension number must be 4 characters long to appear | Extension number must be 4 characters long |
| To see if the *Insert new entry* section of the GUI works correctly | The *Directory* interface, *ListDirectory* class and the GUI class are used. | Surname = Browning  Initials =  BL  Extension number =  1000 | For a message box with Entry added! to appear | Entry added! |

The testing shows that the *Insert new entry* section works correctly and prevents incorrect inputs to be added to the directory.

Deleting an entry:

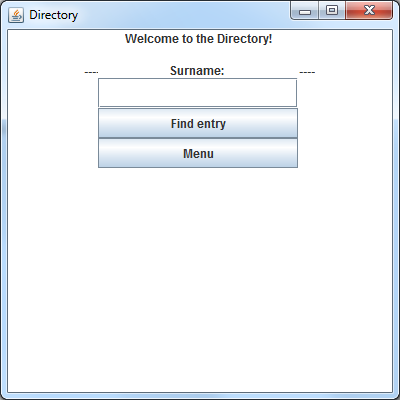
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose of Test | Coverage | Test Values | Results Predicted | Results Actual |
| To see if the *Delete an entry - name section* of the GUI works correctly | The *Directory* interface, *ListDirectory* class and the GUI class are used. | Surname =  Browning | For a message box with Surname not found to appear | Surname not found |
| To see if the *Delete an entry - name section* of the GUI works | The *Directory* interface, *ListDirectory* class and the GUI class are used. | Surname =  Zorro | For a message box with Entry deleted! to appear | Entry deleted! |
| To see if the *Delete an entry -number section* of the GUI works correctly | The *Directory* interface, *ListDirectory* class and the GUI class are used. | Extension number =  1000 | For a message box with Extension number not found to appear | Extension number not found |
| To see if the *Delete an entry -number section* of the GUI works correctly | The *Directory* interface, *ListDirectory* class and the GUI class are used. | Extension number =  0001 | For a message box with Entry deleted! to appear | Entry deleted! |

The testing shows that the *Delete an entry* section works correctly and tells the user if there is an incorrect input.

Lookup an entry:

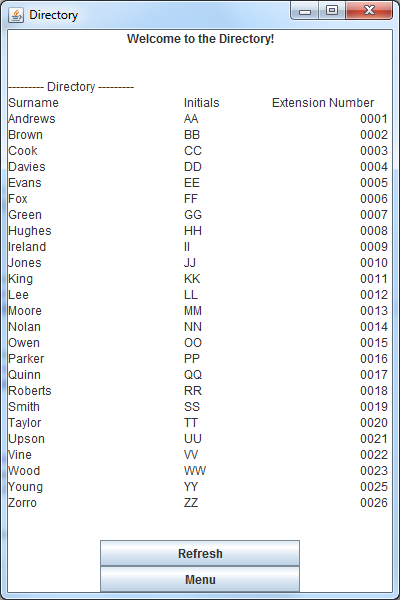
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose of Test | Coverage | Test Values | Results Predicted | Results Actual |
| To see if the *Lookup an entry* section of the GUI works correctly | The *Directory* interface, *ListDirectory* class and the GUI class are used. | Surname =  Browning | For a message box with Surname not found to appear | Surname not found |
| To see if the *Lookup an entry* sectionof the GUI works | The *Directory* interface, *ListDirectory* class and the GUI class are used. | Surname =  Zorro | For a message box with Extension number for Zorro is: 0026 to appear | Extension number for Zorro is: 0026 |

The testing shows that the *Lookup an entry* section works correctly and efficiently.

Changing an extension number:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose of Test | Coverage | Test Values | Results Predicted | Results Actual |
| To see if the *Changing an extension number* section of the GUI works correctly | The *Directory* interface, *ListDirectory* class and the GUI class are used. | Surname =  Browning | For a message box with Surname not found to appear | Surname not found |
| To see if the *Changing an extension number* section of the GUI works correctly | The *Directory* interface, *ListDirectory* class and the GUI class are used. | Surname =  Zorro  Extension number =  1000 | For a label with Extension number for Zorro is: 0026 to appear, a question asking What would you like to change it to? and a text field to appear for the user to enter the new extension number | Extension number for Zorro is: 0026  What would you like to change it to?  Extension number changed! |

The testing shows that the *Changing an extension number* section works correctly and tells the user if the input is incorrect.

Showing the directory:

The screenshot shows that the Show directory section works correctly. The refresh is used for when a new entry has been added or an entry has been deleted, to update the directory.

**TIMING**

The timing results for all of the directories were fairly similar meaning that they all were efficient in looking up an entry and obtaining it’s extension number. The following table shows the results for each directory implementation:

|  |  |
| --- | --- |
| **Directory** | **Time for lookup** |
| ArrayDirectory | 136 |
| ListDirectory | 144 |
| HashDirectory | 121 |

If the directories were to be used with a lot of entries, then the HashDirectory implementation would be the most efficient because of the use of Hashtable.