**Faculty of Engineering & Environment**

**Module**: **CM0718–Program Design and Implementation**

**Module tutor**: **Michael Brockway**

**Title- Assignment 1(Address Book)**

**16/03/2015**

**Pritesh Bhole– w14043450**

**Computing & Information technology**

**Report on Address book System**

**Introduction**

Address book system has three main functions to add a person, to remove a person and to find a person. To achieve above functionality four classes has been used. A brief explanation of functions and responsibilities of each class is explained below

* Name class

This class is made up of variables and methods which are used for temporary storing of persons details. In this class we have initialized variables used for person name and details. It also has getMethods which is used to get data when a user enters a value and display it if called. getMethods include getFname to get first name, getLName to get last name, getNameData to get all details of person, getName to get only person’s name.

* AddressBook class

This class consist of array list declaration, initialization and supporting methods, which is necessary for system functionality. It consists of enrolling method which is used to add a person. getAddressBook method to provide all details of a person when full command is called by user. Similarly there are removeName, matchName, showName method which are used to remove a person from array list, to check uniqueness of a person, to find a person in address book (array list) respectively, according to the input from user.

* AddressBookTextUI class

This class consist of program which create interface between system and user. Scanner variable is initialized and used to accept input from user. It consists of various methods. Void menu is used to accept valid input from user. Void displayMenu displays option available to input for user. Void execute is used to activate proper method if user enters a valid input. Void fullCommand is used to get all persons along with details present in address book. Void findCommand, void removeCommand, void addCommand are used to activate methods in AddressBook class to remove, find or add a person. Various methods like firstName, lastName, stName, stNumber, ctName, zpCode are used to check that user doesn't leave any field blank and also to check a valid UK post code format.

* AddressBookDriver class

This class contains the static void main method which runs the program and forms a system. It calls menu method in addressBookTextUI class to run display.

**JUnit**

As constructors are used to create a new object it should not be null hence assertNotNull command of JUnit has been used to check it. Get methods can be checked by both assertTrue and assertEquals as both of them will compare user input with return values except assertTrue will return true if value match and assertEquals will just check values. To test array list if it stores the value assertNotNull is used and to check if its empty assertTrue is used so that there is no garbage value stored in it. For testing find and remove methods assertEqual with boolean has been used as it will match user input name with stored array list name and if found or removed will return true or else false. For testing add method assertEqual is used as only one person name has been checked if it is added correctly.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Unique ID | Description of the purpose of the test | Pre-condition for running test | Test data | Expected results |
| testAllConstructor() | To test all constructor present in Name class | Create a object with person details | Person details as up | After entering the data constructors should not be null |
| testGetFirstNameMethod() | To test if method is returning the value user has inputted | Create a object with person details | Person details as up | It should return true if first name of person is same as inputted by user |
| testGetLastNameMethod() | To test if method is returning the value user has inputted | Create a object with person details | Person details as up | It should return true if last name of person is same as inputted by user |
| testGetNameDataMethod() | To test if method is returning entire details of person | Create a object with person details | Person details as up | It should return the same person details as inputted by user |
| testGetNameMethod() | To test if method is returning Full Name of person | Create a object with person details | Person details as up | It should return Full name of person as inputted by user |
| testArrayListIsCreated() | To test persons details are getting stored in array list | Create a object addressB1 and call enrolName method | Enter person details by creating a new instance | It will check if user entry is present |
| testIfArrayListIsEmpty() | To check if array list is empty or not | Create a object addressB1 | Do not enter anything | It should give result as true if array list is empty |

**Test Plan**

Test Data includes person details

**First name- pritesh, Last name- bhole, Street name- gowland, Street number-5, City name-newcastle, Zip code- NE4 9NE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Unique ID | Description of the purpose of the test | Pre-condition for running test | Test data | Expected results |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Unique ID | Description of the purpose of the test | Pre-condition for running test | Test data | Expected results |
| testForFindMethod() | To check if user inputted person is present in list | Create a object addressB1 and call enrolName method | Enter person details by creating a new instance | It should work if “priteshbhole” is present in array list as it is added in test data. It will show an error if another person name is added. Error would be Expected<priteshbhole>  But was found<jamespitt> |
| testRemoveMethod() | To check if entered person is found and deleted from array list and if not found in array list | Create a object addressB1 and call void  addCommand method after adding call  void  remove  Command | Enter  persons details  in user interface  and then enter person to be removed | After adding person details if “priteshbhole” is entered to remove than test will execute correctly. If some other person name is entered to remove it will show an error expected<priteshbhole>  Found<> |
| testAddMethod() | To check user inputted person is stored in array list | Create a object addressB1 and call void  addCommand method | Enter  persons details  in user interface | If person is added to array list test will execute correctly for reference purpose only person with name “pritesh bhole” can be added |

Critical evaluation of program

Weakness

System lacks in checking if user enters a valid name and not any speical character.

System lacks in checking if user input unnecessary space

Future development recommendation

Program should eliminate unwanted spaces