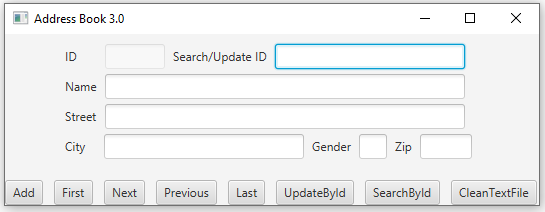
**ADNAN MENDERES UNIVERSITY**

**CSE 203 Object-Oriented Programming**

**Project Address Book**

(2019)



This program is capable of adding, recording, and updating addresses, and shows records. Addresses are recording to a “.dat” file.

AddressBookObject Class:

In AddressBookObject class, BorderPane is extended and alerts, buttons, text fields, and AddressBookObject constructer are created. These buttons added pane2(HBox) and text fields added pane(FlowPane). The dimensions of the text fields are determined. Pane set in the center with setCenter method and pane2 set in the bottom with setBottom method.

AddressBook Class:

In AddressBook class, we declared private int data fields named numberOfPeople and locationInArray(default 0) and final static int data fields named ID(4), NAME(32), STREET(32), CITY(20), GENDER(1), ZIP(5), RECORD\_SIZE(ID+NAME + STREET + CITY + GENDER + ZIP). Later, a private AddressBookObject object named pane, a public RandomAccessFile named raf, and a Person object array named peopleArray are created.

User should enter gender ‘M’ or ‘F’ otherwise IsGenderValueValid function returns false. isZipValueInteger is a Boolean type function and it returns true if the entered zip can parse to integer. areTextFieldsEmpty returns false if user try to leave empty 'name', 'street' and 'city' text fields. Otherwise, the function returns true. isTfSUIdEmpty checks if user try to do not enter integer value in the 'Search/Update Id' text field. The function returns true if the entered argument can parse to integer. Otherwise, it returns false. The functions in the above should returns true at the same time. The isDataAlreadyExist function checks is the tried to entered data already exist. It works by using an if-else condition.

EventBtFirst:

The eventBtFirst is a void typed function. This function works to display the first data in address.dat file. Program will show an error screen if there is no data in address.dat file by using the try-catch structs.

In the function, the locationInArray integer parameter synchronized to zero. Then, locationInArray indexed element in people array, is shown on the application by using the traverseArray function.

EventBtLast:

The eventBtLast is a void typed function. This function works to display the last data in address.dat file. Program will show error screen if there is no data in address.dat file by using the try-catch structs.

In the function, the locationInArray integer parameter synchronized to a minus of peopleInArray integer. Than, locationInArray indexed element in people array, is shown on the application by using the traverseArray function.

EventBtNext:

The eventBtNext is a void typed function. This function works to display the next data from the data that shown on the display. Program will show error screen if no data is shown on the screen, and also error screen will shown if there is no data in address.dat file by using the try-catch structs.

In the function, the locationInArray integer parameter increased by one. Than, the locationInArray indexed element in people array, is shown on the application by using the traverseArray function.

EventBtPrevious:

The eventBtPrevious is a void typed function. This function works to display the previous data from data that shown on the display. Program will show error screen if no data is shown on the screen, and also error screen will shown if there is no data in address.dat file by using the try-catch structs.

In the function, the locationInArray integer parameter decreased by one. Than, the locationInArray indexed element in people array, is shown on the application by using the traverseArray function.

EventBtSearchById:

The eventBtSearchById is a void typed function. This function works to display the data that has same id as entered id in the 'Search/Update Id' text field.

Firstly, the variables are checked for process by using isTfUIdValidForProcess function and wasDataFound boolean parameter. Than, the process continues. By using the for loop, program searchs the data that has same id as entered id in the 'Search/Update Id' text field. The program display the data if there is a match. Otherwise, error screen will shown. Also, by using isTfSUIdValidForProcess function, error screen is shown if user leaves empty the 'Search/Update Id' text field, or the user tries to do not enter integer value in the 'Search/Update Id' text field.

EventBtUpdateById:

The eventBtUpdateById is a void typed function. This function works to update the data that has same id as entered id in the 'Search/Update Id' text field.

Firstly, the variables are checked for process by using isTfUIdValidForProcess, isZipValueInteger, isGenderValueValid, areTextFieldsEmpty functions and wasDataFound boolean parameter. Than, the process continues. By using the for loop, program searchs the data that has same id as entered id in the 'Search/Update Id' text field. Thereafter, the program overwrites the entered data in the text fields with the data that already in the address.dat file. Lastly, the updated data is shown on the application. Error screens are shown if something went wrong by using if-else conditions.

cleantextfields:

The cleantextfields function clears the text fields in the application.

writeAddressToFile:

The writeAddressToFile function writes the spesific data in the address.dat file by using writeFixedLenghtString function in the FileOperations class.

traverseArray:

The traverseArray function displays the spesific data on the text fields of the application.

readFileFillArray:

The readFileFillArray function reads data from address.dat file and the readFileFillArray function adds data in array to process it.

start:

The start function checks that is there any saved data in the address.dat file, and it saves data if is there data. Also, the start function sets geomerty of the stage, and it combines the buttons and their own functions.

main:

The main function launchs the application.

FileOperations Class:

readFixedLengthString:

The readFixedLengthString function is a string typed function. It works to read data in the address.dat file.

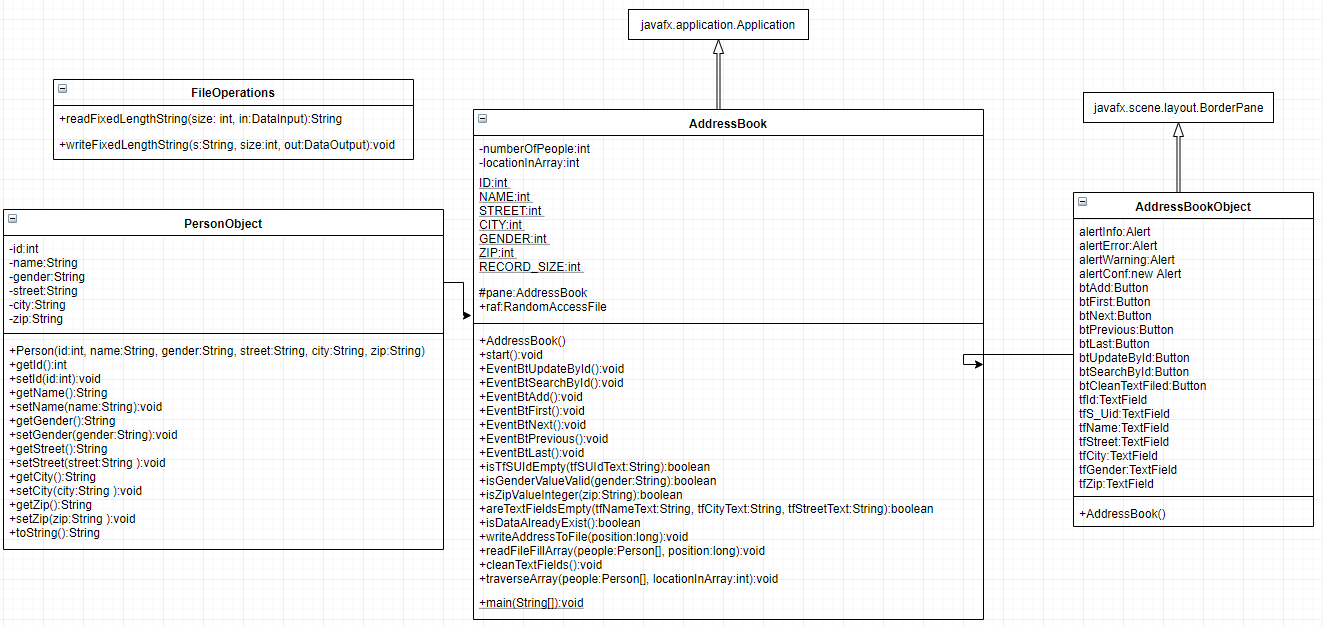
writeFixedLengthString:

The writeFixedLengthString function is a void typed function. It works to write data to in the address.dat file.

PersonObject Class:

A Person class is designed that contains private int data field named id, private String data fields named name, gender, street, city, zip and a constructor that creates a person with the specified id, name, gender, street, city, and zip. Also the super keyword is used. The accessor (getter) and mutator (setter) methods for id, name, gender, street, city, and zip are created. We created toString method which it returns getId, getName, getStreet, getCity, getGender, getZip, and override it.

**UML Diagram:**



We got some problems while we are testing program.

Our first problem was about the gender text field. The user could enter the other characters than "M" or "F". By the way, we created a new function that checks the gender text field.

The other problem was about the zip text field. The user could enter string characters in the zip text field. So, we created new function isZipValueInteger to solve this problem.

Thereafter, we noticed that the user could invalid id in the 'Search/Update Id' text field. We had to show error screen for invalid inputs. So, we created isTfSUIdValidForProcess function. It checks is the input of the 'Search/Update Id' text field valid for the processing.

Also, we noticed the user could add empty data, while we are developing the program. To fix this bug, we created a function that checks are the textfields are leaved empty.

As we close to finish the project, we decide to there must be more alert screen, and we used more if-else statement. As well, we used lots of try-catch bloks to fix the risk that if went something wrong.