**Lab3B**

Lab3B is an assignment that I completed at the beginning of the year while taking Computer Programming II at Jacksonville State University. This assignment was designed to familiarize the class with arrays, array lists, and hash maps. This program reads five hundred lines of input from a file, prints the number of records read, and correctly formats the information into address blocks. Lab3B was written in the programming language Java by using the integrated development environment NetBeans. This program stores all of the data inside array lists and hash maps in order to sort and parse through the input file’s contents.

**Installation**

In order to install this project on your own computer simply right click on the link and save the file to a designated place on your computer.

**Usage**

If you would like to run this program, on your own computer, you have two options. The first, and easiest option, is to open the project folder, open the folder named dist, and double click the executable jar file named Lab3B. Whenever the jar file is opened, five hundred address blocks will be printed in a window. Depending on what type pf computer and operating system you are using this option may or may not work. If not, the other option consists of downloading the integrated development environment NetBeans. In order for NetBeans to work properly, you must first visit oracles website and download Java. The first link is to download Java and the second is to download NetBeans. <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

<https://netbeans.org/downloads/>

Once you have Java and NetBeans downloaded, you will open NetBeans. Next, you will find and open the project folder that you downloaded in Installation. Lastly, you will click the clean and build project icon at the top left of the screen and then the green arrow icon to the right of it to run the program.

**Background**

This section gives a brief explanation on the Java Class Libraries and a few of the methods used in Lab3B. If you already have a decent understanding of java and its class libraries, feel free to skip on to the next section titled Design. In short, the Java Class Libraries are a set of tools predefined by the programming language java that allow for much easier and relaxed coding. These tools include classes and methods that only have to be called upon in order to use. This means the user does not have to manually type out the code for these functions. The only thing the user must do to use these utilities in their program is to explicitly import them at the beginning of their code. For example, if I wanted to use array lists in my program, I would type, “import java.util.ArrayList;” at the beginning of my program. Speaking of which, an array list is a stored collection of elements. These elements can be integers strings, or hash maps, etc. Once initialized, array lists are able to be refined and tampered with. For example, the user can add, remove, and change the location of items in the array list through the use of methods. A hash map is similar to an array list; however, hash maps identify elements by their names instead of location in a list. Both of these classes are of extraordinary importance in Lab3B, as they are constantly used to store and cycle through the contents of the input file.

**Design**

In this section, I will thoroughly examine nearly every line in Lab3B. To get started, Lab3B only consists of one java class. This java class is Lab3B.java which contains the required main method in every java program. In lines 4-8, I import java classes such as ArrayList, HashMap, Scanner, Paths, and Arrays. Next, inside the main method in lines 18 and 19 we initialize two array lists, keys and values. Keys holds key names from header row in the input file and values holds the values read from subsequent rows. Then, the hash map variable rows in line 21 is created which holds the key/value pairs for each data row. Two strings are initialized to act as containers for the file data. Next, we create an array list called data to store all of the hash maps for all the rows. In line 33 we attempt to open the input file by using the Paths object that we imported at the beginning of our program. In line 37-38 we assign the variable “header” equal to the first line of the input file and then assign each header, separated by commas, to the array list keys. In lines 42-68, a while loop is created that will cycle through every row in the input file. Every time a new line is reached a new map is created for the current row. Then, the comma separated values are put into an array list. Lastly, a for loop is created inside the while loop that populates the map by using the key names from the header row and the map is added to the data container. In line 72, the amount of records read is printed and in line 76, the input file is closed. In lines 82-100 a for loop is created which cycles through the data. String variables are initialized and assigned to the information inside of the hash maps by using the getter method. For example, “ String firstname = row.get("FirstName"); ”. At the end of the for loop, all of the information is neatly printed out in the form of address blocks, similar to how a letter would be formatted. This is accomplished by using the System.out.println function and positioning the variables correctly. In our case, we want the person’s full name printed on the top line, the person’s address on the second line, and the persons city, state, and zip code on the bottom line.

**Contributors**

This is simply a notice that all of the code presented was not fully created by me, as my professor Jay Snellen supplied the class with skeleton code for the assignment.