Lab 07: Flumine

CO225: Software Construction

August 23, 2016

1 Introduction

Most programs use data in one form or another, whether it is as input, output, or both. The sources of input and output can vary between a local file, a socket on the network, a database, variables in memory, or another program. Even the type of data can vary between objects, characters, multimedia, and others.

The Java Development Kit (JDK) provides APIs for reading and writing streams of data. These APIs have been part of the core JDK since version 1.0, but are often overshadowed by the more well-known APIs, such as JavaBeans, JFC, RMI, JDBC, and so on. However, input and output streams are the backbone of the JDK APIs, and understanding them is not only crucial, but can also make programming with them a lot of fun.

This hands-on lab takes you through the basics of using Java I/O stream.

2 Problem

In this Lab, you will learn how to handle file and directory related operations. You need to use the given *csv* file and create a phonebook.

3 Practice

Opening a file

Use the $BufferedReader$ class and $FileReader$ class to open a file.

```
import java.io.BufferedReader;
import java.io.FileReader;
...
BufferedReader br = new BufferedReader(new FileReader(pathToCsvFile));
...
```

Reading a file

Once you have access to the file, you can iterate it using the available methods in the *br* object.

```
while ((line = br.readLine()) != null) {

            // use comma as separator
            String[] item = line.split(",");

            System.out.println("Line [a= " + item[0] + " , b=" + item[1] + "]");

        }
```

4 Task

Use the given *contacts.csv* file and create a phonebook application using $HashMap$. Your program should first scan the file and store all the information in a $HashMap$. Then the program should prompt the user to enter a name and display the contact information related to the user (You can assume that our contact list will not have any duplicate names). Program should run until it gets an EOF command as an input. (Check the sample test cases given). Your program should pass the given test cases.

5 Things to consider

You need to consider the following things while designing the PhoneBook

- Proper use of Object-Oriented Programming Concepts
- Proper use of Exception Handling
- Proper use of Data Structures
- Documentation

6 What to turn in?

Submit the Java file(s) *PhoneBook.java* (this file should contain the main method to run the class) to FEeLS before the deadline. Do not submit any form of archives such as .zip etc. If you have more than one Java Class submit them without making an archive.