Java HW7: Final Project

21800025 곽인희

[Overall Code Initiative]

1. Read a total of five folders in the data file: 0001.zip, 0002.zip, 0003.zip, 0004.zip, 0005.zip. To do this, use the zip file reader.

2. Read the two Excel files in each folder.

3. Write a new file under the names of String and Table, respectively, for information saved while reading the file.

[Detailed Code Initiative]

1. Exception

Exception processing is performed when the number of cells is 0.

To do this, when reading the Excel file in the ExcelReader class, allow extension to flow when lastColumn is 0.

|  |
| --- |
| MyException |
|  |
| /\*\*This method is to deal with files or directory which is not following excel format  \*/  +void public MyException()  +void public MyException(String message) |

2. CLI

I get input and output path through the cli.

To do this, create a method to parse an option and a method to make an option. Then, create a print help method to show users the information about help.

(\*\*\*The output file name is divided into String.csv and Table.csv, and you can enter the name of the folder when you receive an output path as a cli for storing different files. You don't have to write down the name of the file. Information is automatically divided into String.csv and Table.csv names in the folder containing the Zip data. Example: -i "C:\Users\Inhee Kwak\git\JavaFinalProject\data" -o "C:\Users\Inhee Kwak\git\JavaFinalProject\data"

|  |
| --- |
| CLI |
| +Static Input\_path  +Static Output\_path  +Static Help |
| + void run(Strin[] args)  + boolean parseOptions(Options options, String[] args)  + Options createOptions  + void printHelp(Options options) |

3. Merge

The data read from ExcelReader and zipfileReader are separately stored as summary and table files, and each data is classified. Merge stores all data in one arraylist and imports it. Therefore, Merge is ordered and classified according to the class number of data types (summary and illustrations and tables), according to the characteristics entered during data storage.

|  |
| --- |
| Merge (in start.java) |
| ArrayList<Object> toGet  ArrayList<Object> allFileContents  ArrayList<Integer> allHeaderNum  ArrayList<Object> allHeader  ArrayList<Object> allTableHeader  ArrayList<Object> allTable  ArrayList<Object> allStr  ArrayList<Integer> table5Header  ArrayList<Integer> table7Header  ArrayList<Integer> table55Header  ArrayList<Integer> tableNum  ArrayList<Integer> strNum |
|  |

4. File Read

First, read each Excel file through ExcelReader. At this time, I save the information to the queue I made. Then, return the queue where the data is stored. It also extends the thread to the ExcelReader to help you read it efficiently when you call the ExcelReader from the zipReader.

|  |
| --- |
| ExcelReader |
| + Workbook wb  + Sheet sheet  + queue q |
| + queue getData(InputStream is) |

Use the Linked list to create a queue. Queue Create each method using basic features and use them to store data.

|  |
| --- |
| queue |
| - LinkedList<String> list |
| + void enqueue(String item)  + String dequeue()  + boolean hasItems()  + int size()  + void addItems(queue q) |

Read the file through the zipReader. Receive read information as q from ExcelReader. Then, enter additional information for classification later and return the queue where the modified data is stored.

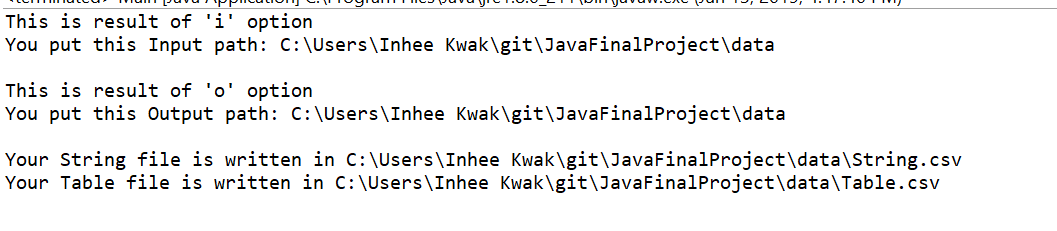
|  |
| --- |
| ZipReader |
| + ZipFile zipFile;  + queue q (to get return value from excelReader)  + Queue<String> queue (for return value from zipReader)  + InputStream stream  + ExcelReader myReader |
| + Queue<String> readFileInZip(String path) |

5. File Write

Write the file in CSV file format. Start classifies, modifies, and receives arraylists and file names containing stored data. Use these data to write the header and data.

|  |
| --- |
| ZipReader |
| + static void writeAFile(ArrayList<String> finishStr, String targetFileName, ArrayList<String> allHeader) throws IOException |
| + FileOutputStream outputStream  + BufferedWriter writer  + CSVPrinter csvPrinter |

[Results screen]

1. Execution Window 

2. String.csv



3. Table.csv

