

TargetReady

day 3 assignment

Assignment #1

CRUD operations on CSV file

Create a CLI-based menu-driven **Java** application that provides the following menu options:

1. Add new customer
2. View all customers
3. Search customers by city
4. Delete a customer (by ID)
5. Search customers by ID and edit/update details
6. Exit

The data for the application is maintained in a CSV file.

Sample CSV data with headers:

```
id,name,city,email,phone
1,Dell Goldfinch,Shijiazhai,dgoldfinch0@netlog.com,8609657940
2,Robb Groven,Tabu,rgroven1@bandcamp.com,5208396145
3,Frannie Mardoll,Pacajus,fmardoll2@tuttocitta.it,3377870047
4,Donaugh Bierling,Gvardeysk,dbierling3@cmu.edu,1214078739
5,Clarie Swiggs,Bílovce,cswiggs4@rediff.com,6662222043
6,Cahra Michiel,Cennan,cmichiel5@usnews.com,3641707554
7,Hinda Moye,Sanjiang,hmoye6@yellowbook.com,1037138359
8,Malia Larmouth,Ābdānān,mlarmouth7@histats.com,8858893642
9,Florentia Blundon,Zaoxi,fblundon8@quantcast.com,7373066451
10,Erwin Kohneke,Zhongxiao,ekohneke9@alibaba.com,9725490770
```

Make the application modular by splitting the code into reusable functions/classes.
Handle exceptions properly so that the application exits only after selecting choice #6.

Assignment #2

Sure, I'll adjust the Java assignment to use arrays instead of lists. Here's the revised version:

Text File Analyzer

Objective: Write a Java program that reads a text file, analyzes its contents, and performs various operations on arrays, implements error handling, and defines methods to accomplish the tasks.

Requirements:

1. Array Operations:

- Read the contents of the text file into an array of strings. Each line of the file should be an element in the array.
- Implement a method to find the longest and shortest lines in the file.
- Implement a method to count the number of words in each line and store them in an array of integers.
- Implement a method to sort the array of word count in descending order.

2. Error Handling:

- Handle errors that may occur during file reading and other operations.
- Display appropriate error messages if any operation fails.

3. Text File Handling:

- Open and read the contents of a text file specified by the user.
- Close the file after reading.

4. Methods:

- Define methods for each of the tasks mentioned above (e.g., `readFile`, `findLongestLine`, `findShortestLine`, `countWords`, `sortWordCount`).

Additional Instructions:

- The program should take the filename as input from the user.
- Display the contents of the longest and shortest lines along with their line numbers.
- Display the word count for each line.
- Display the sorted word count.
- Ensure the program is well-documented with comments explaining the purpose of each method and major blocks of code.
- Test your program with different text files of varying lengths and contents to ensure it works correctly under various scenarios.

Sample Output:

Enter the filename: `example.txt`

Contents of the file:

`This is a sample text file.`

`It contains multiple lines with different lengths.`

`Each line will be analyzed by your app.`

Longest line:

`Line 2: It contains multiple lines with different lengths.`

Shortest line:

`Line 1: This is a sample text file.`

Word count for each line:

Line 1: 6 words

Line 2: 7 words

Line 3: 8 words

Sorted word count:

8 words

7 words

6 words

Function signature reference:

```
import java.io.FileNotFoundException;
```

```
public class TextFileAnalyzer {
```

```
    // Method to find the longest line in the array of strings
```

```
    public static String findLongestLine(String[] lines) {
```

```
        // Implementation to find the longest line
```

```
    }
```

```
    // Method to find the shortest line in the array of strings
```

```
    public static String findShortestLine(String[] lines) {
```

```
        // Implementation to find the shortest line
```

```
    }
```

```
    // Method to count the number of words in each line and store them in an array
```

```
    public static int[] countWords(String[] lines) {
```

```
        // Implementation to count words in each line
```

```
    }
```

```
    // Method to sort the array of word counts in descending order
```

```
    public static void sortWordCount(int[] wordCounts) {
```

```
        // Implementation to sort word counts
```

```
    }
```

```
    // Main method to execute the program
```

```
    public static void main(String[] args) {
```

```
        // Implementation of main program logic
```

```
    }
```

```
}
```

You will need to implement the main program logic in the `main` method and the methods `findLongestLine`, `findShortestLine`, `countWords`, and `sortWordCount` as per the provided function signatures. Additionally, handle file reading, error handling, and other required functionalities accordingly.