Phase1 Project Specification

Explorer

By: Nilakshi Nitinkumar Patil
Date: 11/07/2021

Explorer

- **Project Objective:**

  Explorer is the prototyped application for file explorer, which performs basic file handling operations such as creating a new file, listing files, deleting a file and searching a file

- **Project Details:**

  Project Name: Explorer
  Developed by: Nilakshi Patil

- **Sprint Details:**
  - **Sprint1:**
    1. Setting up Git and GitHub account to store and track the enhancements of the prototype
    2. Design Welcome screen and basic menu driven logic.
    3. Initialise the Scanner class object to accept the user input.
    4. Handle the cases where exceptions are likely to be caused and in case of invalid inputs.
    5. Add the two features: Listing files in ascending order and creating a new file (case insensitive).
    6. Test the implemented features and fix the bugs found while testing.
    7. Stage and commit all changes to Git and GitHub.

  - **Sprint2:**
    1. Add remaining features, i.e. Deleting file if exists and searching file and show appropriate result upon performing these operations.
    2. Read input folder path from ProjectConfig.properties file to make the application dynamic.
    3. Add binary search logic as a searching technique for performance optimization.
    4. Test the implemented features and fix the bugs found while testing.
    5. Stage and commit all changes to Git and GitHub and release the prototype product.

- **Algorithm:**
    1. Read ProjectConfig.properties and read property value i.e. **RootDirectory**. If this property is absent or file path is absent, then take the current user directory path.
    2. Display the main menu containing the following three options and take user input as number to perform respective action.
        2.1. **List files**: Display all files present in RootDirectory.
        2.2. **File Menu**: Go to Step 3
        2.3. **Close the Application**: Close the application.
    3. Display file menu options:
        3.1. **Add file**: Create new file
        3.2. **Delete file**: Delete file if exists
        3.3. **Search file**: Search file and display result
        3.4. **Main Menu**: Navigate back to the main menu
    4. Repeat Step 3, until the **Main Menu** is not selected.
    5. Repeat Step 2, until the **Close the Application** is not selected.

- **Core Concepts used:**
    - **Classes and Objects:** The project logic is divided into classes, which works on a single responsibility principle.
      Example:
      StartExploring.java: This is the main class where the welcome message is displayed and the main menu and file menu displaying logic is written. It calls up the other class object to perform user specified operations.

    - **File IO and Scanner class:** File IO class is used to implement file operations and Scanner class is used to accept input from the user.

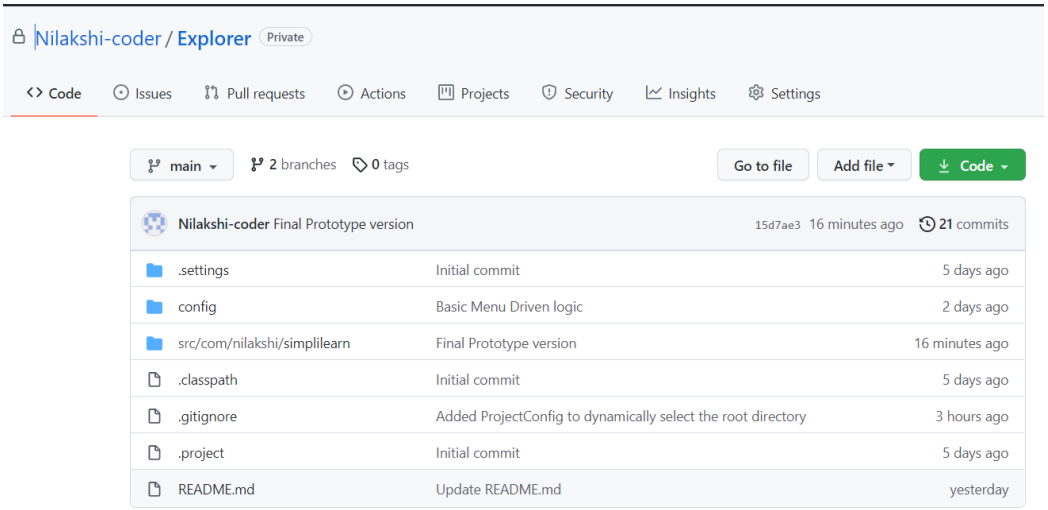
    - **Interface and Inheritance:** To achieve abstraction, interface is being used, i.e. **FileSystem.java**. The FileSystem interface has the following abstract methods- addFile(), deleteFile(), searchFile() and list(). **FileOperations.java** class implements this interface and has implemented all it's abstract methods. To hide the internal implementation details, FileManager.java class has a factory method which uses dynamic binding for object creation of **FileOperations** class.

    - **Exception Handling:** All exceptions are handled using try catch blocks to avoid disruption of application execution flow. For example, Integer input is expected, but you entered String. So in such a case, the user will be notified with an appropriate message and asked to select the correct option.

- ❖ **Data Structures used:** List Collection is used. Binary Search is used for optimization.

- ❖ **Java8 Features:** Lambda expressions and streams are used for better performance and concise code.

- **GitHub Link:**
  https://github.com/Nilakshi-coder/Explorer



- **Conclusion:**

  The current Explorer version is a very basic version. It can be enhanced by adding following features to it:

  1. Current scope for this project is limited to the files. This can be extended to the folders as well.
  2. Rename, Copy feature can be added for ease.
  3. Currently files are shown only in ascending order. This feature can be extended to list the files based on several choices and depending on user choice.
     Example- Listing the file in descending order based on file name,
     Listing the file in ascending/descending order based on file modified time.
  4. File Searching in subsequent directories.

  Above mentioned features can enhance the user experience of the product.