**FILE HANDLING APP**

**Developer Details:**

N Prasad

BTECH (Department of Computer science and Engineering)

BS Abdhur Rahman Crescent Institute of science and technology.

**Contact**: [prasadnithi15@gmail.com](mailto:prasadnithi15@gmail.com)

**Source code:**

For Ubuntu :🡪 <https://github.com/prasad-ops/BNP-Project.git>

For Windows:🡪 <https://github.com/prasad-ops/BNP-Project-Windows.git>

**PROBLEM STATEMENT:**

To develop a prototype of the application and then push the source code to the GitHub repository. The user interaction will be via a command line. The documentation should contains the following lists.

* Product’s capabilities, appearance, and user interactions
* Number and duration of sprints required
* Setting up Git and GitHub account to store and track your enhancements of the prototype
* Java concepts being used in the project
* Data Structures where sorting and searching techniques are used.
* Generic features and three operations:
  + Retrieving the file names in an ascending order
  + Business-level operations:
    - Option to add a user specified file to the application
    - Option to delete a user specified file from the application
    - Option to search a user specified file from the application
    - Navigation option to close the current execution context and return to the main context
  + Option to close the application

**PRODUCT CAPABILITIES AND USER INTERACTION:**

Some of the common file handling operations are;

* Create file
* Delete file
* Read file
* Write file
* Change file permissions

The File class from the **java.io package**, allows us to work with different formats of files. In order to use the File class, you need to create an object of the [class](https://www.edureka.co/blog/java-tutorial/#obj) and specify the filename or directory name.

The user interaction will be via a command line.

**AGILE SCRUM FRAMEWORK:**

Scrum is an agile project management methodology or framework used primarily for software development projects with the goal of delivering new software capability every 2-4 weeks. It is one of the approaches that influenced the [Agile Manifesto](http://agilemanifesto.org/), which articulates a set of values and principles to guide decisions on how to develop higher-quality software faster.

**Scrum Events (Ceremonies)**

*The Sprint*

A sprint is a time-boxed period during which specific work is completed and made ready for review. Sprints are usually 2-4 weeks long but can be as short as one week.

*Sprint Planning* Sprint

Planning team meetings are time-boxed events that determine which product backlog items will be delivered and how the work will be achieved.

*The Daily Stand-up*

The Daily Stand-up is a short communication meeting (no more than 15 minutes) in which each team member quickly and transparently covers progress since the last stand-up, planned work before the next meeting, and any impediments that may be blocking his or her progress.

*The Sprint Review*

The Sprint Review is the "show-and-tell" or demonstration event for the team to present the work completed during the sprint. The Product Owner checks the work against pre-defined acceptance criteria and either accepts or rejects the work. The stakeholders or clients give feedback to ensure that the delivered increment met the business need.

*The Retrospective*

The Retrospective, or Retro, is the final team meeting in the Sprint to determine what went well, what didn't go well, and how the team can improve in the next Sprint. Attended by the team and the ScrumMaster, the Retrospective is an important opportunity for the team to focus on its overall performance and identify strategies for continuous improvement on its processes.

**REQUIREMENTS:**

* Eclipse/IntelliJ: An IDE to code for the application
* Java 8: A programming language to develop the prototype
* Git: To connect and push files from the local system to GitHub
* GitHub: To store the application code and track its versions
* Scrum: An efficient agile framework to deliver the product incrementally
* Specification document: Any open-source document or Google Docs

**Git Repository links:**

For Ubuntu :🡪 <https://github.com/prasad-ops/BNP-Project.git>

For Windows:🡪 <https://github.com/prasad-ops/BNP-Project-Windows.git>

**CONCEPTS:**

**MAVEN:**

Maven’s primary goal is to allow a developer to comprehend the complete state of a development effort in the shortest period of time. In order to attain this goal, Maven deals with several areas of concern:

* Making the build process easy
* Providing a uniform build system
* Providing quality project information
* Encouraging better development practices

**Exception handling:**

It is one of the most important feature of java programming that allows us to handle the runtime errors caused by exceptions. In this guide, we will learn what is an exception, types of it, exception classes and how to handle exceptions in java with examples.

### **Checked exceptions**

All exceptions other than Runtime Exceptions are known as Checked exceptions as the compiler checks them during compilation to see whether the programmer has handled them or not. If these exceptions are not handled/declared in the program, you will get compilation error. For example, SQLException, IOException, ClassNotFoundException etc.

### **Unchecked Exceptions**

Runtime Exceptions are also known as Unchecked Exceptions. These exceptions are not checked at compile-time so compiler does not check whether the programmer has handled them or not but it’s the responsibility of the programmer to handle these exceptions and provide a safe exit. For example, ArithmeticException, NullPointerException, ArrayIndexOutOfBoundsException etc

**COLLECTIONS:**

It is a framework that provides an architecture to store and manipulate the group of objects.

Java Collections can achieve all the operations that you perform on a data such as searching, sorting, insertion, manipulation, and deletion.

Java Collection means a single unit of objects. Java Collection framework provides many interfaces (Set, List, Queue, Deque) and classes ([ArrayList](https://www.javatpoint.com/java-arraylist), Vector, [LinkedList](https://www.javatpoint.com/java-linkedlist), [PriorityQueue](https://www.javatpoint.com/java-priorityqueue), HashSet, LinkedHashSet, TreeSet).

**INTERFACE:**

An **interface in Java** is a blueprint of a class. It has static constants and abstract methods.

The interface in Java is a mechanism to achieve [*abstraction*](https://www.javatpoint.com/abstract-class-in-java). There can be only abstract methods in the Java interface, not method body. It is used to achieve abstraction and multiple [inheritance in Java](https://www.javatpoint.com/inheritance-in-java).

In other words, you can say that interfaces can have abstract methods and variables. It cannot have a method body.

**ALGORITHM:**

* Start the program.
* Print the options displaying the user interaction information.
* accept the user input to select one of the options listed
* The first option should return the current file names in ascending order.
* Create DisplayClass which implements Display. Get the file name as Lists. Perform Collections.sort() for printing the list in ascentin order.
* The second option should add a file to the existing directory list.
* Create AddClass which implements AddInterface. Get the directory name to which the file has to get create.
* Check whether the file is already exist or not. If the file is not available, create the file in the mentioned directory. Else print that the file is already exist.
* The third option should delete a file from the existing directory.
* Create DeleteClass. Get the directory name to which the file for deletion.
* Check whether the file is already exist or not. If the file is available, delete the file in the mentioned directory. Else print that the file is not available.
* The fourth option should search a file from the existing directory.
* Create SearchClass. Get the directory name to which the file for searching.
* add the case sensitivity on the file name to retrieve the correct file
* Display the result upon successful operation
* Display the result upon unsuccessful operation
* Last option to close the application.
* Stop the program

**HOW TO EXECUTE THE APP:**

The app is developed for both windows and Ubuntu system.

The github link is provided through which the app can be downloaded based on the operating system of your device.

STEP 1:

Run the APP directly in eclipse. The user interaction should be in command line.

The welcome page of the app provide a menu through which user can handle files.

Both windows and Ubuntu app will provide the same menu option.

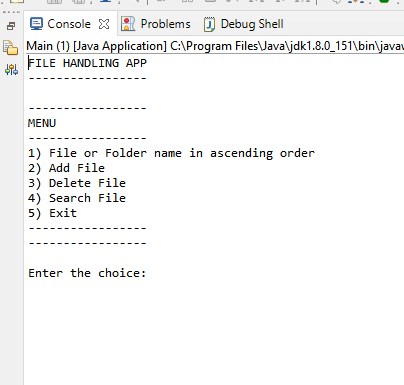


Fig1. Welcome page.

STEP 2:

When the user want to view the file name in the directory .

Enter the choice as 1.It will ask the user to choose the directory in which the file name has to view.After choosing the directory name. The list of files in that directory is displayed in ascending order.

In Ubuntu system the user cant able to choose the directory. It will fix the default directory and display the files in it.

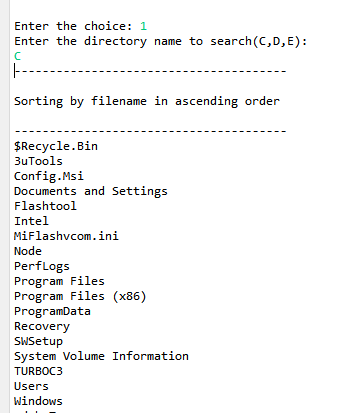


Fig 2. File or folder name in ascending order.

STEP 3:

After performing each task. The app navigate back to the menu page.

The user can add or create the file in the directory by choosing the option 2.

It will ask the user to select the directory to which the file should create.

User should enter the name of the file along with extension. It will create the file in the selected directory.

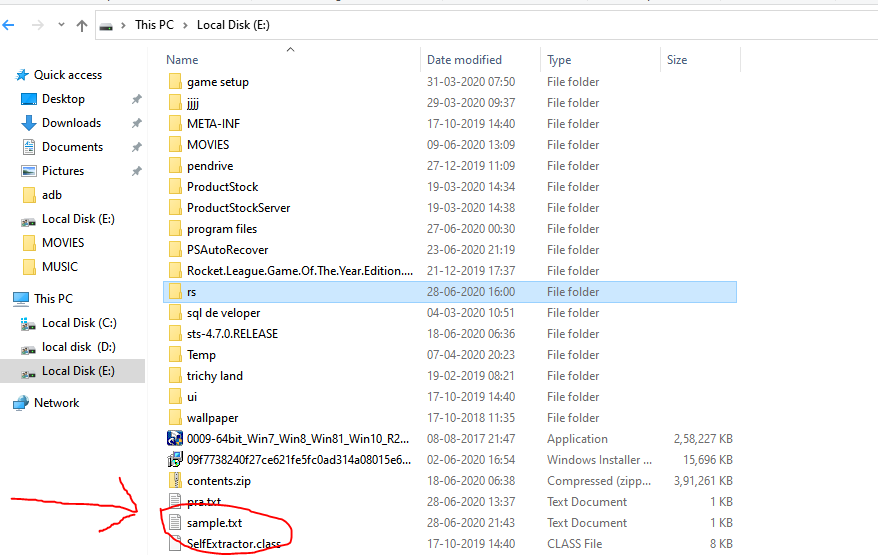
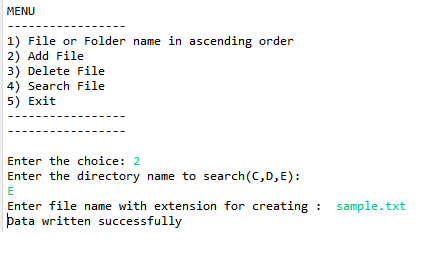


Fig 3. Creating File.

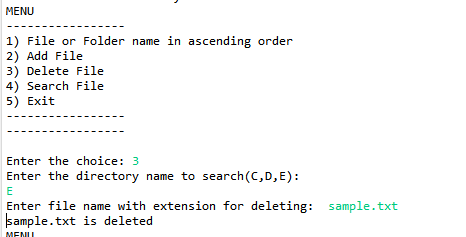
STEP 4:

The user can delete the file in the directory by choosing the option 3.

It will ask the user to select the directory to delete the file.

User should enter the name of the file along with extension. It will delete the file from the selected directory.

In Ubuntu system the default path is specified. if necessary to change the path, user should modify the path address in the source code.



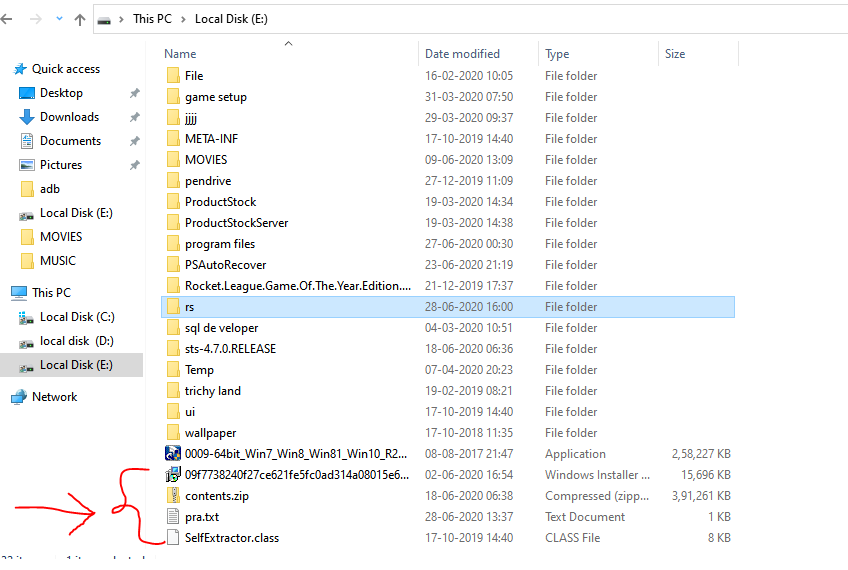


Fig 4. File deletion.

STEP 5:

The user can search the file in the directory by choosing the option 4.

It will ask the user to select the directory for searching the file.

User should enter the name of the file along with extension. It will search the file from the selected directory. It will display the result on successful and unsuccessful operation.

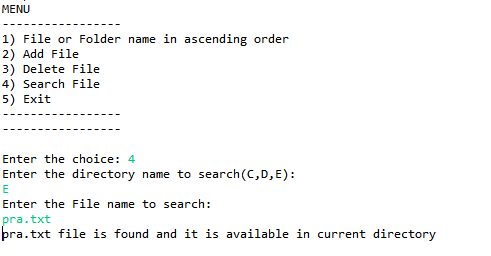


Fig 5. Search operation.

STEP 6:

After all the process user can close the app by selecting option 5.

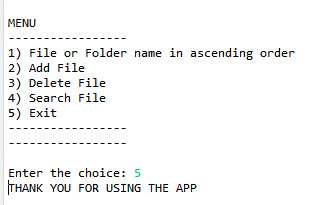


Fig 6. Terminating the app.

CONCLUSION:

Thus, the mentioned task in the problem statement are executed. This was all about the various file operations in Java. The source code is pushed into the git repository. The git link for downloading the source code is given in respect to operating system(windows and Ubuntu). Different between windows and Ubuntu file handling system, the Ubuntu version comes with default directory path which is written in the source program. You need to modify the path internally in the program in-order to change the default directory. Whereas, windows version allows the user to select the directory in which file operation can occur.