Full-stack development program by SimpliLearn & Vodafone

Project 1: LockedMe.com prototype

Documentation

Developer: Michaela Tomanova
Date: 29th January, 2023

Sprint planning

There were 3 five-day sprints planned. Tasks achieved in these sprints are listed below:

Sprint 1
- Project instruction understanding
- Functionalities definition
- User interface design

Sprint 2
- File class definition
- Business logic creation

Sprint 3
- Exception handling
- Program testing
- USPs definition and enhancement proposals

Prototype description

The prototype starts with listing the application and developer names and the main menu options. Users can select options by typing the option number. In case of invalid user input a warning message is shown and the menu is listed again.

The first main menu option displays the names of files in the database.
The second option shows the submenu where files can be added, deleted or searched.
The third option closes the program.

```
WELCOME at LockedMe.com
(developed by Michaela Tomanova)


MAIN MENU: Please select one of these options
1   Display all files
2   Modify files
3   Quit the program
ajsfhob oihewgw
Invalid input -> select 1, 2 or 3

MAIN MENU: Please select one of these options
1   Display all files
2   Modify files
3   Quit the program
```

```
MAIN MENU: Please select one of these options
1   Display all files
2   Modify files
3   Quit the program
2

Please choose your next step
    1   Add a new file
    2   Delete an existing file
    3   Search an existing file
    4   Go back to the main menu
4

MAIN MENU: Please select one of these options
1   Display all files
2   Modify files
3   Quit the program
```

In the main menu, the first option displays all file names in ascending order. If there is no file in the database, an “empty database” message is shown.

```
MAIN MENU: Please select one of these options
1   Display all files
2   Modify files
3   Quit the program
1
No files added yet, the database is empty

MAIN MENU: Please select one of these options
1   Display all files
2   Modify files
3   Quit the program
```

```
MAIN MENU: Please select one of these options
1   Display all files
2   Modify files
3   Quit the program
1
AnotherFile
different file
file1
File1
New file
```

Users can add a file selecting the appropriate option and typing the file name. Creating duplicate files in the database is not allowed. The file name can contain spaces - after finishing the name, “enter” button should be pressed to confirm. The file names are also case-sensitive, meaning that for example files named “file”, “File” and “FILE” are three different files in the database.

```
Please choose your next step
    1   Add a new file
    2   Delete an existing file
    3   Search an existing file
    4   Go back to the main menu
1
Enter file name to add:
File1
File "File1" added.
```

```
Please choose your next step
    1   Add a new file
    2   Delete an existing file
    3   Search an existing file
    4   Go back to the main menu
1
Enter file name to add:
different file
File "different file" already exists, file not added
```

Deleting a file can be provided by selecting the “delete” option and typing the file name. Again, the file names are case-sensitive, so deleting the appropriate file is ensured. Users are informed about either deleting the file or not finding the file in the database.

```
Please choose your next step
    1   Add a new file
    2   Delete an existing file
    3   Search an existing file
    4   Go back to the main menu
2
Enter file name to delete:
file1
File "file1" deleted
```

```
Please choose your next step
    1   Add a new file
    2   Delete an existing file
    3   Search an existing file
    4   Go back to the main menu
2
Enter file name to delete:
OneMoreFile
File "OneMoreFile" not found, cannot be deleted
```

A file can be searched in the database by selecting the “search” option and typing the file name. After that a message is shown, either that the appropriate file was found in the database or not.

```
Please choose your next step
    1   Add a new file
    2   Delete an existing file
    3   Search an existing file
    4   Go back to the main menu
3
Enter file name to search:
AnotherFile
File "AnotherFile" found in the database
```

```
Please choose your next step
    1   Add a new file
    2   Delete an existing file
    3   Search an existing file
    4   Go back to the main menu
3
Enter file name to search:
DifferentFile
File "DifferentFile" not found
```

Via the last option of the submenu users can return to the main menu. If wanted, users can then quit the prototype application by selecting the “quit” option in the main menu.

```
Please choose your next step
    1   Add a new file
    2   Delete an existing file
    3   Search an existing file
    4   Go back to the main menu
4

MAIN MENU: Please select one of these options
1   Display all files
2   Modify files
3   Quit the program
```

```
MAIN MENU: Please select one of these options
1   Display all files
2   Modify files
3   Quit the program
3

Program has ended. Good Bye!

Process finished with exit code 0
```

Algorithms and flowcharts

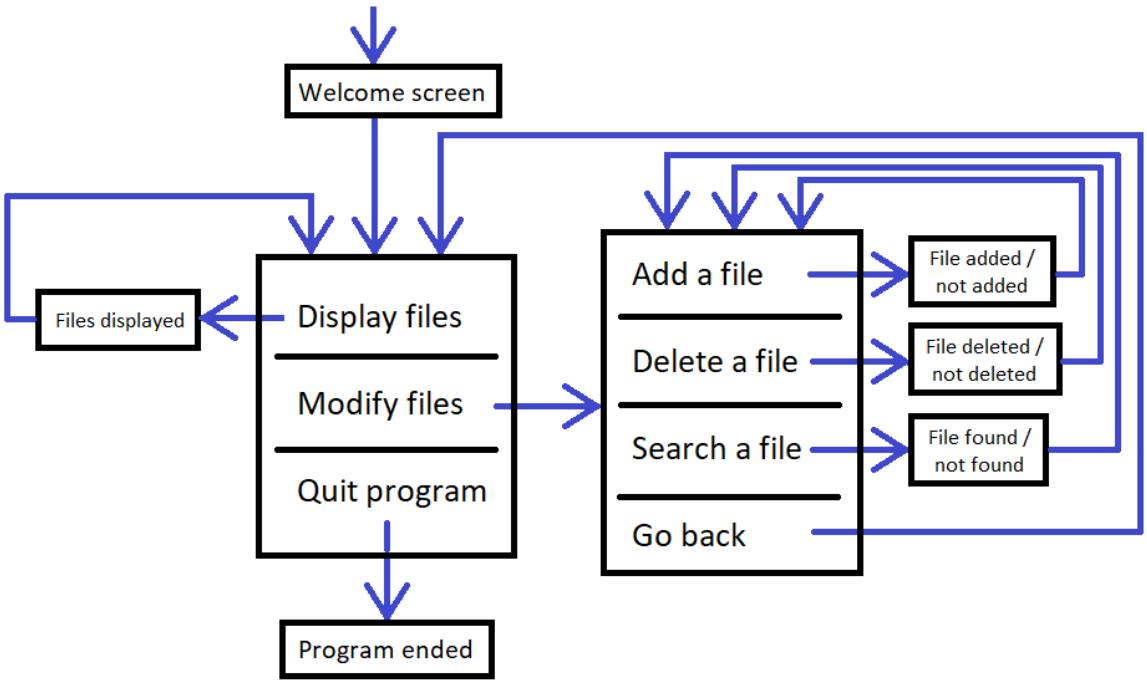
The application prototype is divided into 3 classes - Files, FileDatabase and UserInterface class.

The “Files” class defines the template for file objects. Now, it contains only name-of-the-file variable yet, more attributes can be added in the future. This class also implements the “Comparable” interface so the comparing method could be created.

In the “FileDatabase” class a database using ArrayList is created and key methods such as display, add, delete and search are defined.

As the name indicates, the “UserInterface” class contains the main class and provides users the interaction with the application prototype. It displays options, accepts user’s choices and calls appropriate methods from the “FileDatabase” class.

The application prototype flow is shown in the following picture:



Core concepts used

Object-oriented programming concepts

- In this application prototype, files are represented as objects of the Files class. More file attributes like author, type or description can be added in the future.
- The file database is also represented as an object and eventually more databases can be created from FileDatabase class, so files can be sorted in different directories.

Collections, sorting & searching

- For creating and updating the file database an ArrayList is used ("java.util.List" and "java.util.ArrayList" are imported).
- Files sorting is implemented by using the Comparable interface and imported collection method (java.util.Collections).
- As this application is a prototype, linear search for file searching is implemented.

Exception handling

- The try-catch blocks are used to validate the user input. If the user inputs anything else than options listed in the menus, a warning message is shown repeating the valid input options.

GitHub repository

https://github.com/mianto5/FSDp_Project1.git

Conclusion and enhancing possibilities

LockedMe.com prototype is an application which provides an easy and clear way of working with files. It provides basic file handling functionalities as adding a file into the database, deleting a file from the database and searching a file in the database. The existing files in the database can be displayed in ascending order.

Users interact with the application via a command line. Every user's step starts with listing the file handling options to choose from and corresponding number to input. In case of invalid input users are informed again from which options they can choose. Every time a file handling operation is executed, a message is shown to users, either that the operation was successful or that the operation failed and why.

The application prototype is designed so that more file details can be added easily, for example author name and other file specifications. Based on these details, a filtering option can be designed to display or search existing files meeting certain criteria.

Also multiple databases can be created to form more complex data structures. Files would then be stored in the databases based on the user's choice or based on mutual similarities.