

Virtual Key for your repositories

Introduction:

This is an program with generic features like:

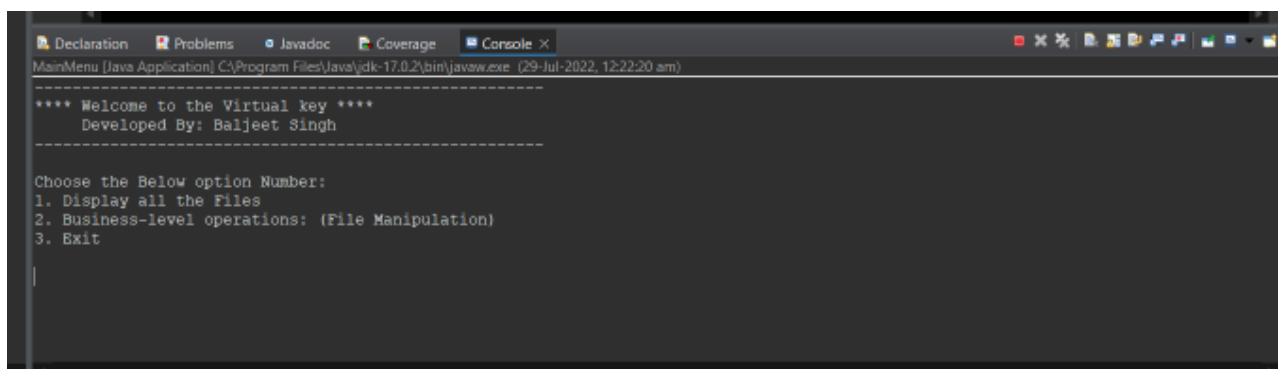
- Retrieving the file names in an ascending order
- Business-level operations:
 - Option to add a user specified file to the application
 - Option to delete a user specified file from the application
 - Option to search a user specified file from the application
 - Navigation option to close the current execution context and return to the main context
- Option to close the application.

Installation Guide:

1. GitHub link: <https://github.com/baljeet-singh97//Virtual-Key-for-Your-Repositories>
2. Download the entire project as Zip in local system.
3. import the project in **Eclipse IDE**

How to use:

1. After running the code you will see the window like this in which you have to select the desired option.

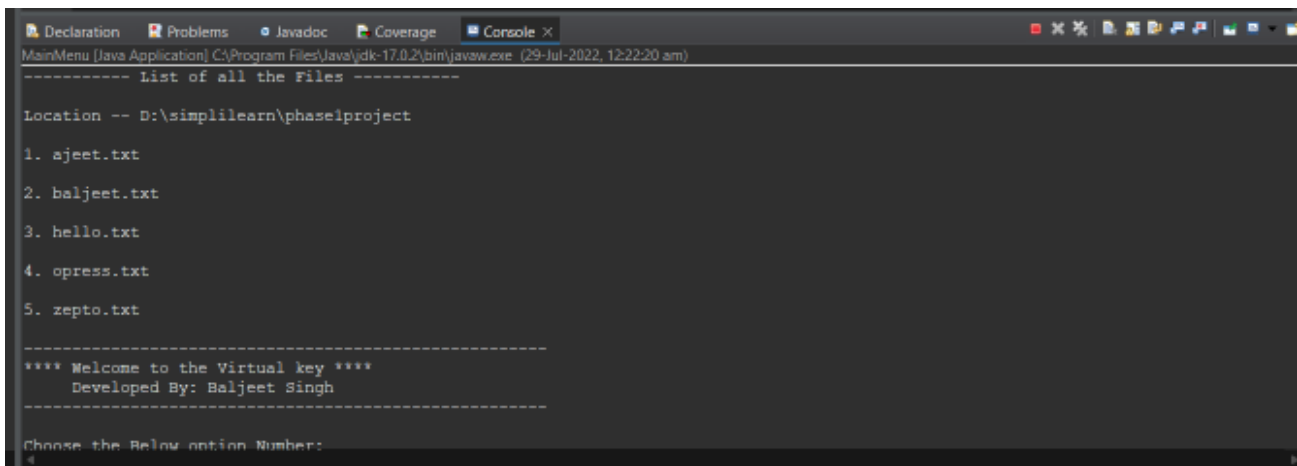


```
Declaration Problems Javadoc Coverage Console X
Main Menu [Java Application] C:\Program Files\Java\jdk-17.0.2\bin\javaw.exe (29-Jul-2022, 12:22:20 am)

**** Welcome to the Virtual key ****
    Developed By: Baljeet Singh
-----

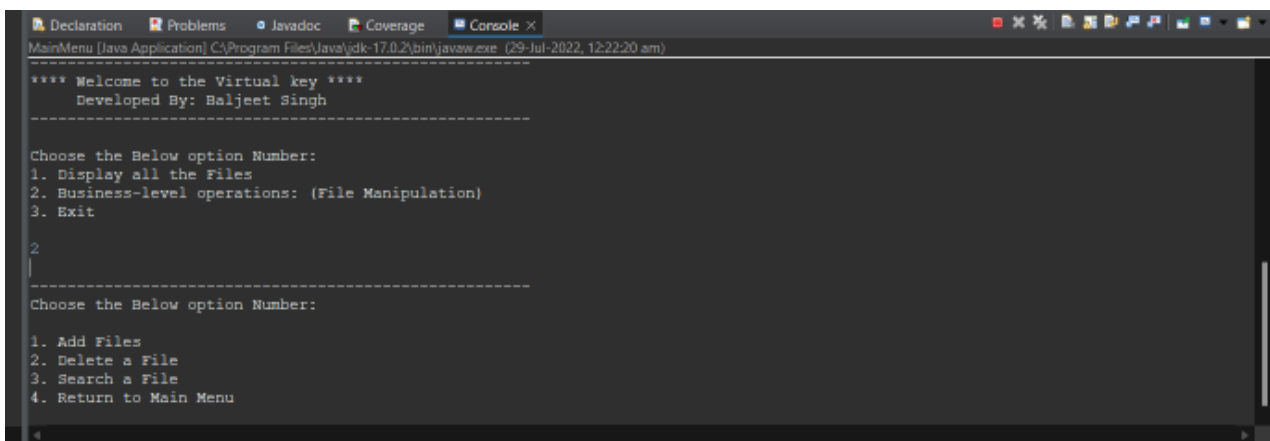
Choose the Below option Number:
1. Display all the Files
2. Business-level operations: (File Manipulation)
3. Exit
|
```

2. If user selects “option 1. Display all the files” then it will print all the files present in the directory, as shown below.

A screenshot of a Java application's console window. The window title is 'MainMenu [Java Application] C:\Program Files\Java\jdk-17.0.2\bin\javaw.exe (29-Jul-2022, 12:22:20 am)'. The console output shows a list of files in a directory: 'Location -- D:\simplilearn\phase1project' followed by a list of five files: '1. ajeet.txt', '2. baljeet.txt', '3. hello.txt', '4. opress.txt', and '5. zepto.txt'. Below this list, there is a separator line, a welcome message '**** Welcome to the Virtual key ****', the developer's name 'Developed By: Baljeet Singh', another separator line, and the prompt 'Choose the Below option Number:'. The cursor is positioned at the end of the prompt.

```
Declaration Problems Javadoc Coverage Console X
MainMenu [Java Application] C:\Program Files\Java\jdk-17.0.2\bin\javaw.exe (29-Jul-2022, 12:22:20 am)
----- List of all the Files -----
Location -- D:\simplilearn\phase1project
1. ajeet.txt
2. baljeet.txt
3. hello.txt
4. opress.txt
5. zepto.txt
-----
**** Welcome to the Virtual key ****
Developed By: Baljeet Singh
-----
Choose the Below option Number:
4
```

3. If user selects “option 2 Buisness-level operations: (File Manipulation), then this window will come up with several options. As shown below.

A screenshot of a Java application's console window. The window title is 'MainMenu [Java Application] C:\Program Files\Java\jdk-17.0.2\bin\javaw.exe (29-Jul-2022, 12:22:20 am)'. The console output shows a welcome message '**** Welcome to the Virtual key ****', the developer's name 'Developed By: Baljeet Singh', a separator line, and the prompt 'Choose the Below option Number:'. Below this, there is a list of three options: '1. Display all the Files', '2. Business-level operations: (File Manipulation)', and '3. Exit'. The number '2' is entered, followed by a vertical bar '|' as input. Below this, there is another separator line, the prompt 'Choose the Below option Number:', and a list of four options: '1. Add Files', '2. Delete a File', '3. Search a File', and '4. Return to Main Menu'. The cursor is positioned at the end of the prompt.

```
Declaration Problems Javadoc Coverage Console X
MainMenu [Java Application] C:\Program Files\Java\jdk-17.0.2\bin\javaw.exe (29-Jul-2022, 12:22:20 am)
**** Welcome to the Virtual key ****
Developed By: Baljeet Singh
-----
Choose the Below option Number:
1. Display all the Files
2. Business-level operations: (File Manipulation)
3. Exit
2
|
-----
Choose the Below option Number:
1. Add Files
2. Delete a File
3. Search a File
4. Return to Main Menu
4
```

4. When user selects “option 1. Add Files”, user will be asked to enter number of files user want to create after that a pop up for file name and after that write something to file, Same will repeat n number of times user given before. And then all files and data will be saved into D: drive.

```
Declaration Problems Javadoc Coverage Console X
MainMenu [Java Application] C:\Program Files\Java\jdk-17.0.2\bin\javaw.exe (29-Jul-2022, 12:22:20 am)

1. Add Files
2. Delete a File
3. Search a File
4. Return to Main Menu
1
Enter number of files you want to add
2
enter file name(abc.txt).
file1.txt
Write something.
inside file1
Data Written Successfully
enter file name(abc.txt).
file2.txt
Write something.
inside file2
Data Written Successfully
-----
**** Welcome to the Virtual key ****
-----
```

5. When user selects “option 2. Delete a File” user will be asked to enter the name of the file. After that a message will be shown successfully deleted or not as per situations.

```
Declaration Problems Javadoc Coverage Console X
MainMenu [Java Application] C:\Program Files\Java\jdk-17.0.2\bin\javaw.exe (29-Jul-2022, 12:22:20 am)

1. Add Files
2. Delete a File
3. Search a File
4. Return to Main Menu
2
Enter the name of the file (abc.txt)
file1.txt
File file1.txt deleted successfully.

-----
**** Welcome to the Virtual key ****
Developed By: Baljeet Singh
-----

Choose the Below option Number:
1. Display all the Files
2. Business-level operations: (File Manipulation)
3. Exit
```

6. If user selects “option 3. Search a File”. User has to enter the name of the file and after entering the name it will print the content of file if file available in the directory.

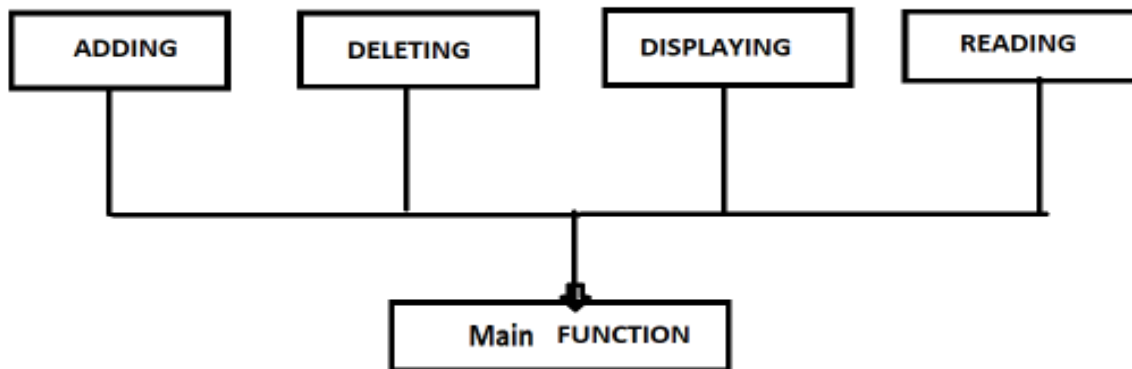
```
MainMenu [Java Application] C:\Program Files\Java\jdk-17.0.2\bin\javaw.exe (29-Jul-2022, 12:22:20 am)

2. Delete a File
3. Search a File
4. Return to Main Menu
3
Enter the name of the file you want to delete (abc.txt)
file2.txt
inside file2
--- End of file ---

-----
**** Welcome to the Virtual key ****
Developed By: Baljeet Singh
-----

Choose the Below option Number:
1. Display all the Files
2. Business-level operations: (File Manipulation)
3. Exit
```

Code Description:



Classes:

1. MainMenu :

This class includes all the 'welcome Screen material' all the switch cases and imports all other classes data.

At first we are displaying all the welcome screen material.

Taking input from the user in variable 'operation' and iterated it in switch case statement.

Case 1:

Closed case 1 in try and catch block, in try block made object of class "DisplayFile" and then called the Display method in that class. this class displays all the files and directory stored in the directory. And used "main(null)" so that program back to its home screen after every operation.

```
DisplayFile obj = new DisplayFile();  
obj.Display();  
main(null);
```

Case 2:

case 2 includes a nested switch case in which file manipulation operations are performed "case 2 shows the various operations for file manipulation" includes "add file", "delete File", "Search File", "Return to main menu" taking input from the user in variable named 'choice' and based on that switched the cases.

Case 2 --> Case 1:

Used try and catch block to handle exceptions. In here calling the add file class which performs operation to creating and adding text file in local storage. Made object 'obj' for class 'AddFile' and called method 'Add()' inside that class.

```
AddFile obj = new AddFile();  
obj.Add();
```

Case 2:

Case 2 includes operation to delete a file from the database created object 'obj' for class 'DeleteFile' and called the method inside that class name 'Delete()'.

```
DeleteFile obj = new DeleteFile();  
obj.Delete();
```

Case 3:

Case 3 includes the read operation if user want to read a specific file then user has to select the case 3. In here we have created a object 'obj' for class ReadFile and called the method 'read()' inside that class.

```
ReadFile obj = new ReadFile();  
obj.read();
```

Case 4:

Case 4 is just used to return back to the main menu or outside of the window Business level operation. Called main(null).

```
Main(null);
```

Case 3:

Case 3 used to exit from the program user presses for case 3 program stops working.

```
System.out.println("Thank you for using this app. Have a NICE DAY.");
```

2. AddFile :

Created a method Add() to perform all the activities related to adding file.

```
public void Add() throws IOException
```

Created a directory to store all the txt files at one place

```
String path = "D:\\simplilearn\\phase1project\\";
```

```
File theDir = new File(path);
```

Added an "If" statement to check if the directory is present in the directory if it is present or already created then program will move to next statement otherwise it will create directory and move to next.

```
If(!theDir.exists())
```

```
{theDir.mkdirs();}
```

Then program asks for number of files user want to create then took input from the user for file number and run a loop n*times called another method "creating()" in the same class to create the file.

In method creating() defined the process of creating and writing content of txt files.

The default path to save txt files

```
String path= "D:\\simplilearn\\phase1project\\";
```

```
Scanner sc = new Scanner(System.in);
```

```
System.out.println("enter file name(abc.txt).");
```

```
String filename = sc.nextLine();
```

Adding path and file name

```
String FinalPath = path+filename;
```

Opening file with specified location

```
FileOutputStream out= new FileOutputStream(FinalPath);
```

Asking user to enter the data in the file.

```
System.out.println("Write something.");
```

Writing data to txt file

```
out.write(array);
```

```
System.out.println("Data Written Successfully");
```

Closing the file after creating it

```
out.close();
```

3 . DeleteFile

Firstly user is being asked to enter the name of the file which user want to delete.

Storing the file name and path in Delfile as a File method.

```
File Delfile = new File("D:\\simplilearn\\phase1project\\"+fileName);
```

```
if(Delfile.delete())
```

Checking if file deletes print “successfully deleted” otherwise “failed to delete”.

4 . DisplayFile

In this class created a method with name Display() in this method all the actions are being performed for displaying all the file in the current directory.

Providing the default directory location

```
File path = new File("D:\\simplilearn\\phase1project\\");
```

Adding all the file names in an array

```
File contentsArr[] = path.listFiles();
```

Sorting the array so all files display in ascending order.

```
Arrays.sort(contentsArr);
```

Adding If statement if directory is empty it will print “directory is empty”.

```
if(contentsArr.length<=0)  
{System.out.println("Directory is empty");}
```

In else statement using for loops displaying the path once and all the file names. Used listFiles it provides the address and name of the file .

Printing path of the directory

```
System.out.println("Location -- "+path+"\n");
```

Printing all the file names using for loop

```
for(int i=0; i<contentsArr.length; i++)  
{ System.out.println(i+1+" "+contentsArr[i].getName());  
System.out.println();}
```

4 . ReadFile

In this class created a method read().

Taking name of the file user want to delete.

```
String fileName = sc.nextLine();
```

Getting location of the file in stream using FileInputStream

```
FileInputStream stream= new  
FileInputStream("D:\\simplilearn\\phase1project\\"+fileName);
```

Reading the whole text file and output the text.

```
while((data=stream.read())!=-1)  
{System.out.print((char)data);}
```