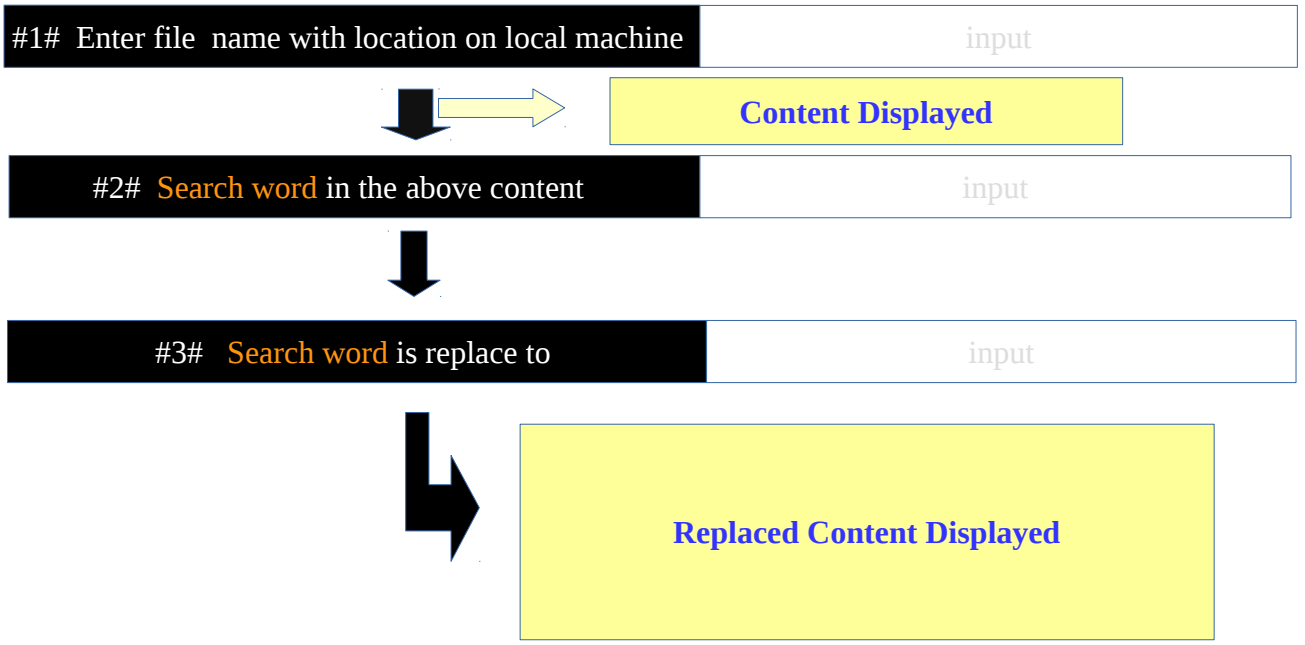
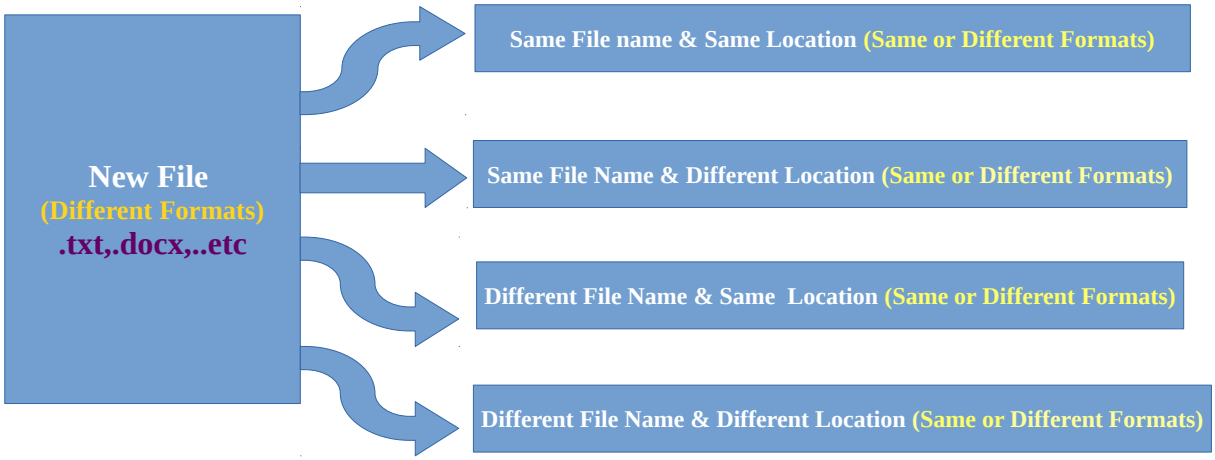
File – String - Search – Replace - Save





Reference Document :

Introduction

Java has a concept of working with streams of data. You can say that a Java program reads sequences of bytes from an input stream (or writes into an output stream): byte after byte, character after character, primitive after primitive. Accordingly, Java defines various types of classes supporting streams, for example, InputStream or OutputStream. There are classes specifically meant for reading character streams such as Reader and Writer.

Before an application can use a data file, it must open the file. A Java application opens a file by creating an object and associating a stream of bytes with that object. Similarly, when you finish using a file, the program should close the file —that is, make it no longer available to your application.

Below is a list of very important java library classes related to Streams.

Class	Description
InputStream	Abstract class containing methods for performing input
OutputStream	Abstract class containing methods for performing output
FileInputStream	Child of InputStream that provides the capability to read from disk files
FileOutputStream	Child of OutputStream that provides the capability to write to disk files
PrintStream	Child of FilterOutputStream, which is a child of OutputStream; PrintStream handles output to a system’s standard (or default) output device, usually the monitor
BufferedInputStream	Child of FilterInputStream, which is a child of InputStream; BufferedInputStream handles input from a system’s standard (or default) input device, usually the keyboard

..

Source Code :

```
package com.penchal.test;


import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;

import java.util.Scanner;

public class FileStringReplaceTest {
	public static void main(String args[]) {
		try {

			String line = "", oldText = ""; int content;

			Scanner sc = new Scanner(System.in);

			System.out.println("Enter File Name and Location :");
			System.out.println("------------------------------");

			//open the file on local machine
			String localMachineFile = sc.nextLine();
			File file = new File(localMachineFile);

			BufferedReader reader = new BufferedReader(new
FileReader(file)); //FileReader connection open

			// File content displaying--Start
			System.out.println("Existing text file content is :");
			System.out.println("------------------------------");
			try (FileInputStream fis = new FileInputStream(file)) {

				while ((content = fis.read()) != -1) {
					// convert to char and display it
					System.out.print((char) content);
				}

			} catch (IOException e) {
				e.printStackTrace();
			}
			// File content displayed--End


			//search the word
			System.out.println("Search word in the above file content :");
			System.out.println("---------------------------------------");
			String existWord = sc.nextLine();

			//replace the word
			System.out.println("'" + existWord + "'" + " is replace to
-->");

	System.out.println("----------------------------------------");
			String replaceWord = sc.nextLine();

			while ((line = reader.readLine()) != null) {
				oldText += line + "\r\n";
```

```
                }
                reader.close();     //FileReader connection closed


                //business logic for replace a word in a file
                String newText = oldText.replaceAll(existWord, replaceWord);

                //uncomment the below line the replaced content will be saved
on same location and same file
                //FileWriter writer = new FileWriter(localMachineFile);

                // Different file name and different file location (that's
your wish)
                System.out.println("Create your own file to save the replaced
content --> :");

      System.out.println("-----------------------------------------------------
");

                String createNewTextFile = sc.nextLine();

                File newFileGenerate = new File(createNewTextFile);
                FileWriter writer = new FileWriter(newFileGenerate);  //
FileWriter connection is open
                writer.write(newText);

                System.out.println("'" + existWord + "'" + " is replaced to "
+ "'" + replaceWord + "'"
                            + " into the file.The updated content of the file
is : ");


System.out.println("-------------------------------------------------------------
-----");

                //replaced content display
                System.out.println(newText);

                writer.close(); // FileWriter connection is closed

            } catch (IOException ioe) {
                ioe.printStackTrace();
            }
      }
}
```