

Reflection log

Class Definition: FindAndReplace

The `FindAndReplace` class is a console-based application that allows users to search for a specific word or phrase in a file, replace it with another word, and save the updated content back to the file.

Main Method:

The main method is responsible for launching the program and ensuring the search and replace process runs in a proper sequence. It prompts the user for input (file name, search word, and replacement word), checks whether the file exists, and calls the necessary methods to perform the find and replace operation.

```
public static void main(String[] args) {  
  
    // Run the program and ensure the FindAndReplace process is  
    handled correctly  
  
    Scanner scanner = new Scanner(System.in);  
  
  
    // Prompt user for file name, search word, and replacement word  
    System.out.print("Enter the file name: ");  
  
    String filename = scanner.nextLine();  
  
  
    // Create an instance of FindAndReplace to handle the operations  
    FindAndReplace findAndReplace = new FindAndReplace();  
  
    findAndReplace.processFile(filename, scanner);  
  
  
    scanner.close();  
}
```

```
}
```

Constructor:

The constructor of the `FindAndReplace` class doesn't need to initialize any GUI components because this is a console application. However, it does initialize important fields, if needed, to store information for further processing.

```
public FindAndReplace() {  
  
    // Constructor can be kept empty or used to initialize any fields,  
    if necessary  
  
}
```

Method: `processFile`

This method is responsible for handling the file operations. It reads the contents of the file, checks if the file exists, performs the search and replace operation, and writes the modified content back to the file.

```
public void processFile(String filename, Scanner scanner) {  
  
    try {  
  
        File file = new File(filename);  
  
  
        // Check if file exists  
  
        if (!file.exists()) {  
  
            System.out.println("File [" + filename + "] was not  
found.");  
  
            return;  
  
        }  
  
  
  
        // Read the content of the file
```

```
String content = new
String(Files.readAllBytes(Paths.get(filename)));

// Ask for the search term and replacement term from the user
System.out.print("Enter the word or phrase to search for: ");
String searchWord = scanner.nextLine();

// Check if the word exists in the file
if (!content.contains(searchWord)) {
    System.out.println("The word [" + searchWord + "] was not
found in the file.");
    return;
}

System.out.print("Enter the replacement word or phrase: ");
String replacementWord = scanner.nextLine();

// Perform the replacement
String updatedContent = content.replace(searchWord,
replacementWord);

// Write the updated content back to the file
Files.write(Paths.get(filename), updatedContent.getBytes());
```

```
        System.out.println("Replacement completed successfully.");

    } catch (IOException e) {

        System.out.println("An error occurred while reading or writing
the file.");

        e.printStackTrace();

    }

}
```

Key Components of the Class:

1. **File Handling:**
 - The file is read using `Files.readAllBytes` and the updated content is written back using `Files.write`.
 - If the file doesn't exist, the program informs the user and exits early.
2. **User Input:**
 - The program prompts the user to input the filename, the search word, and the replacement word. This ensures the user can specify the exact file and operation they wish to perform.
3. **Search and Replace:**
 - The program uses the `String.replace()` method to perform the find-and-replace operation, which replaces all occurrences of the search word with the replacement word in the file's content.
4. **Error Handling:**
 - Proper exception handling is used to catch and report any errors related to file I/O, such as if the file is not found or if there's an issue while reading/writing to the file.