Master of Applied Computing

COMP 8547 Advanced Computing Concepts – Summer 2024

Assignment 4

Task 6: Extracting Email Addresses from Text

Technical Report

Author:
Yesha Umeshkumar Patel
110164042

Submitted to:
Dr. Olena Syrotkina

Introduction

This report details the implementation of a program to extract email addresses from a text file. The program utilizes regular expressions to identify email address patterns within the text and compiles a list of all extracted email addresses.

Implementation

1. Regular Expression to Find Email Addresses

A regular expression (regex) is a sequence of characters that define a search pattern. For this task, the regex will match typical email address formats as shown in Figure 1.

```
String emailRegex_yesha = \\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\\.[A-Z|a-z]{2,}\\b;
```

2. Method to Extract Email Addresses

The method `extractEmailsFromFile` reads the text from a file, applies the regex to find all email addresses, and returns them in a list as shown in Figure 1.

ExtractEmailsFromFile Method:

- Take the file name as an argument.
- Initializes a list (`extractedEmailAddresses_yesha`) to store email addresses.
- Uses `BufferedReader` to read lines from the file.
- Appends each line to a `StringBuilder` (`fileContentBuilder_yesha`) to form the complete file content.
- Uses regex to find and extract email addresses from the file content.
- Adds each found email address to the list.
- Handles `IOException` to ensure the file is read and closed properly.

Main Method (main):

- Defines the file name (`fileName_yesha`) containing the text to be processed.
- Calls `extractEmailsFromFile(fileName_yesha)` to extract email addresses.
- Prints each extracted email address.

```java
package assignment4;

import java.io.BufferedReader;

public class Assignment4 {

    public static void main(String[] args) {
        String fileName_yesha = "hoteltext.txt";
        List<String> extractedEmailAddresses_yesha = extractEmailsFromFile(fileName_yesha);

        // Displaying the extracted email addresses
        System.out.println("Extracted Email Addresses:");
        for (String emailAddress_yesha : extractedEmailAddresses_yesha) {
            System.out.println(emailAddress_yesha);
        }
    }

    public static List<String> extractEmailsFromFile(String fileName_yesha) {
        List<String> extractedEmailAddresses_yesha = new ArrayList<>();
        BufferedReader bufferedReader_yesha = null;

        try {
            bufferedReader_yesha = new BufferedReader(new FileReader(fileName_yesha));
            StringBuilder fileContentBuilder_yesha = new StringBuilder();
            String line_yesha = bufferedReader_yesha.readLine();

            while (line_yesha != null) {
                fileContentBuilder_yesha.append(line_yesha);
                fileContentBuilder_yesha.append(System.lineSeparator());
                line_yesha = bufferedReader_yesha.readLine();
            }

            String fileContent_yesha = fileContentBuilder_yesha.toString();
            // Regular expression to find email addresses
            String emailRegex_yesha = "\\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\\.[A-Z|a-z]{2,}\\b";
            Pattern emailPattern_yesha = Pattern.compile(emailRegex_yesha);
            Matcher emailMatcher_yesha = emailPattern_yesha.matcher(fileContent_yesha);
```

Figure 1: Implementation of regular expressions to extract email from text

3. Output

Figure 2 shows the console output listing the email addresses extracted from the text file. Each email address is printed on a new line under the heading "Extracted Email Addresses"

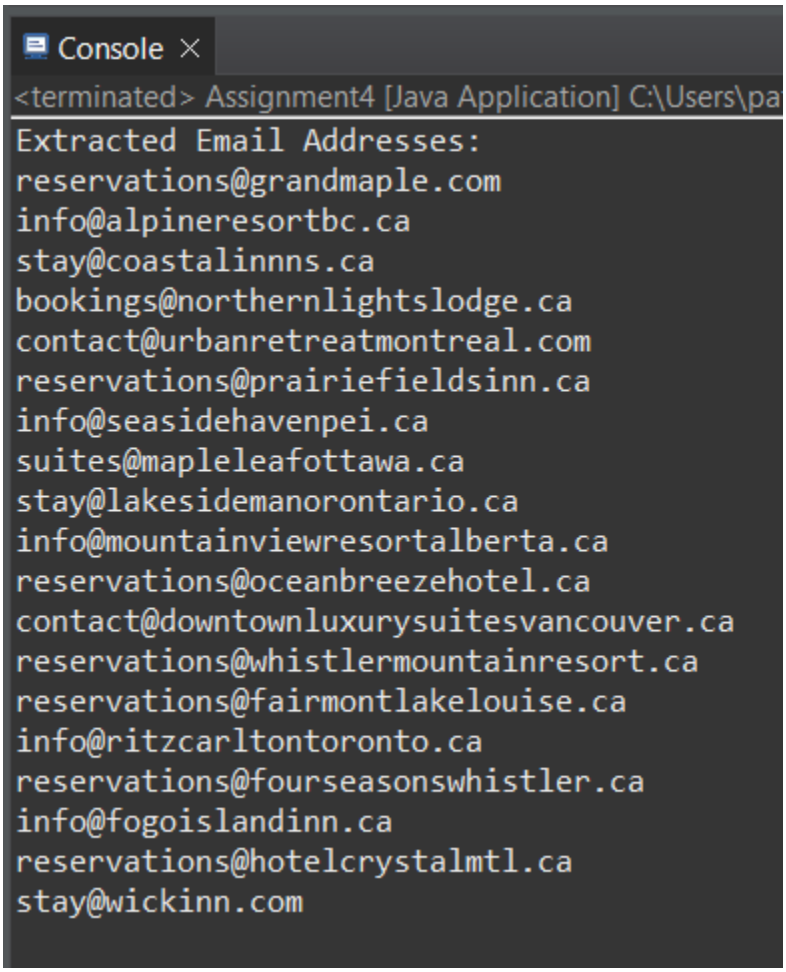

Figure 2: Console output

Conclusion

The assignment successfully demonstrated the implementation of a feature to extract email addresses from a text file using Java. By leveraging regular expressions, the program efficiently identified and extracted valid email addresses, displaying them in the console. The process involved reading the file, applying the regex pattern, and managing potential exceptions, ensuring robustness and reliability.