College of Engineering, Thiruvananthapuram

Object-Oriented Programming Lab

Anirudh A. V.

S2 CSE R2, Roll No. 11

Department of Computer Science
& Engineering

February 12, 2022

1 Reader/Writer

1.1 Aim

To Write a file handling program in Java with reader/writer.

1.2 Algorithm

1. Start
2. Use the try statement to create the FileWriter object.
3. Create the file through the path if it doesn't exist or open it with path if it does exist.
4. Write a sample string "data" into the file using the file object.
5. Accept the user input and write the string into the file using file object.
6. Close the file object/stream.
7. Declare catch statements to catch any IO exeptions if it occurs.
8. Open the file just created by creating a FileReader object in a try statement.
9. Read the characters from the file and display them to the user.
10. Close the file object/stream.
11. Catch statement is specified to catch IO exceptions.
12. Stop

1.3 Code

```java
import java.io.*;
import java.util.Scanner;

public class Filehandling {
    public static void main(String [] args){
        try {
            Writer w = new FileWriter("file.txt");
            Scanner sc = new Scanner(System.in);
            System.out.printf("\nEnter the data : ");
            String content = sc.nextLine();
            w.write(content);
            w.close();
            sc.close();

            Reader r = new FileReader("file.txt");
            if (r.ready()) {
                System.out.println("\nData inside the file : ");
                int character = r.read();
                while (character != -1) {
                    System.out.print((char)character);
                    character = r.read();
                }
                r.close();
            } else {
                System.err.println("\nFile not Found.\n");
            }
        } catch (Exception e) {
            System.err.println(e+" occurred.\n");
        }
    }

}
```

1.4 Sample Output

```
PS E:\Anirudh\Anirudh\CET\SEM 3\OOP_Lab\Java_cycle_3> javac Filehandling.java
PS E:\Anirudh\Anirudh\CET\SEM 3\OOP_Lab\Java_cycle_3> java Filehandling

Enter the data : Java is simple and Powerful

Data inside the file :
Java is simple and Powerful
PS E:\Anirudh\Anirudh\CET\SEM 3\OOP_Lab\Java_cycle_3>
```

2 File Reading

2.1 Aim

To write a Java program that reads a file and displays the file on the screen, with a line number before each line.

2.2 Algorithm

1. Start

2. Use the try statement to create the FileWriter object.

3. Create the file through the path if it doesn’t exist or open it with path if it does exist.

4. Accept the number of lines the user wants to write into the file.

5. Accept the lines from the user and write them to the file using the FileWriter object.

6. Close the file object/stream.

7. Declare catch statements to catch any IO exeptions if it occurs.

8. Open the file just created by creating a FileReader object in a try statement.

9. Keep count of every line using an incrementer.

10. Print the line number followed by the line in the file (read using the FilReader object).

11. Close the file object/stream.

12. Catch statement is specified to catch an IO exceptions.

13. Stop

2.3 Code

```java
import java.io.*;

public class Filenumber {
    public static void main(String[] args) {
        try {
            FileReader f = new FileReader("code.txt");
            String str = "";
            if (f.ready()) {
                int data = f.read();
                while (data != -1) {
                    str = str + (char) data;
                    data = f.read();
                }
            } else {
                System.err.println("\nFile not Found\n");
            }
            int n = 2;
            System.out.print("1 ");
            // int len = str.length();
            for (int i = 0; i < str.length(); i++) {
                if (str.charAt(i) == '\n') {
                    System.out.print(str.charAt(i));
                    System.out.print(n + " ");
                    n++;
                } else {
                    System.out.print(str.charAt(i));
                }
                f.close();
            }
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

2.4Sample Output

```
PS E:\Anirudh\Anirudh\CET\SEM 3\OOP_Lab\Java_cycle_3> javac Filenumber.java
PS E:\Anirudh\Anirudh\CET\SEM 3\OOP_Lab\Java_cycle_3> java Filenumber
1 #include <stdio.h>
2
3 void main(){
4     printf("Hello, World!);
5 }
PS E:\Anirudh\Anirudh\CET\SEM 3\OOP_Lab\Java_cycle_3>
```

3 Number of words, lines & Characters

3.1 Aim

To write a Java program that displays the number of characters, lines, and words in a text file.

3.2 Algorithm

1. Start
2. Create a class Counter.
3. A counter() method is created to read from the file and display all the counts.
    a. Use try statement to create the FileWriter object.
    b. Open the file by creating a FileReader object in a try statement.
    c. As characters are read, increment n_chars variable to keep count.
    d. As spaces are detected increment n_wrds to keep count of all the words.
    e. As the new line character is read, increment n_lines to keep count of all the lines.
    f. Print the number of lines, number of words, and the number of characters to the user.
    g. Close the file object/stream.
    h. Catch statement is specified to catch IO exceptions.
4. Declare the main function and create an object of the class Counter.
5. Call the counter() method of the object.
6. Stop

3.3 Code

```
import java.io.*;

public class Numberof_file {
    public static void main(String [] args){
        try {
            FileReader f = new FileReader("file.txt");
            String str = "";
            if (f.ready()) {
                int data = f.read();
                while (data !=-1) {
                    str = str + (char)data;
                    data = f.read ();
                }
            } else {
                System.err.println("\nFile not Found\n");
            }
            f.close();
            int new_line_count = 0, len = str.length(), no_of_words = 0,
no_of_lines = 0;;
            for (int i = 0; i < len; i++) {
                if (str.charAt(i) == '\n') {
                    new_line_count++;
                    no_of_lines++;
                }
                if ((str.charAt(i) == '\n' || str.charAt(i)==' ') &&
str.charAt(i-1) != ' ' && str.charAt(i-1) != '\n') {
                    no_of_words++;
                }
            }
            System.out.printf("No. of characters : %d\nNo. of words : %d\nNo.
of lines : %d", len-new_line_count, no_of_words+1, no_of_lines+1);
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

3.4 Sample Output

```
PS E:\Anirudh\Anirudh\CET\SEM 3\OOP_Lab\Java_cycle_3> javac Numberof_file.java
PS E:\Anirudh\Anirudh\CET\SEM 3\OOP_Lab\Java_cycle_3> java Numberof_file
No. of characters : 29
No. of words : 7
No. of lines : 1
PS E:\Anirudh\Anirudh\CET\SEM 3\OOP_Lab\Java_cycle_3>
```

4 Reading & Writing a File

4.1 Aim

To write a Java program that reads from a file and writes to a file by handling all file-related exceptions.

4.2 Algorithm

1. Start
2. Use try statement to create the FileWriter object.
3. Create the file through the path if it doesn’t exist or open it with path if it does exist.
4. Wrtite sample lines from into the file using the FileWriter object.
5. Close the file object/stream.
6. Declare catch statements to catch all IO exeptions if it occurs.
7. Open the file just created by creating a FileReader object in a try statement.
8. Loop till the end of the file and Print all the lines from the file
9. Close the file object/stream.
10. Catch statement is specified to catch all IO exceptions if any.
11. Stop

4.3 Code

```java
import java.io.*;
import java.util.Scanner;

public class readnwrite {

    public static void main(String[] args) {
        try {
            Writer w = new FileWriter("file.txt");
            Scanner sc = new Scanner(System.in);
            System.out.printf("\nEnter the data : ");
            String content = sc.nextLine();
            w.write(content);
            w.close();
            sc.close();

            Reader r = new FileReader("file.txt");
            if (r.ready()) {
                System.out.println("\nData inside the file : ");
                int character = r.read();
                while (character != -1) {
                    System.out.print((char)character);
                    character = r.read();
                }
                r.close();
            } else {
                System.err.println("\nFile not Found.\n");
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

4.4 Sample Output

```
Enter the data : I am Baymax, Nice to meet you

Data inside the file :
I am Baymax, Nice to meet you
PS E:\Anirudh\Anirudh\CET\SEM 3\OOP_Lab\Java_cycle_3>
```

5 String Tokenizers

5.1 Aim

To write a Java program that reads a line of integers, and then displays each integer, and the sum of all the integers.

5.2 Algorithm

1. Start
2. Use the try statement to create the FileWriter object.
3. Create the file through the path if it doesn't exist or open it with the path if it does exist.
4. Accept the list of integers separated by spaces.
5. Write the list into the file using the FileWriter object.
6. Close the file object/stream.
7. Declare catch statements to catch an IO exception if it occurs.
8. Open the file just created by creating a FileReader object in a try statement.
9. Check if the file is empty. If yes, exit out of the program else continues:
    A) Create the StringTokenizer object
    B) Loop through all of the tokens
        a) Read the tokens to a variable
        b) Convert the token characters to numbers of type int.
        c) Display the integer to the user.
        d) Add the value of the integers to a sum variable.
    C) Display the value of the sum to the user.
10. Close the file object/stream.
11. Catch statement is specified to catch IO exceptions if any.
12. Stop

5.3 Code

```java
import java.util.Scanner;
import java.util.StringTokenizer;

public class Line_of_Integers {
    public static void main(String[] args) {
        int sum = 0;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the line of numbers : ");
        String str = sc.nextLine();

        StringTokenizer num = new StringTokenizer(str, " ");
        while (num.hasMoreTokens()) {
            String s = num.nextToken();
            System.out.println(s);
            sum = sum + Integer.parseInt(s);
        }
        System.out.println("The sum of the list of numbers is "+sum);
        sc.close();
    }
}
```

5.4 Sample Output

```
Enter the line of numbers :
1 2 3 4 5 6 7 8 9 10
1
2
3
4
5
6
7
8
9
10
The sum of the list of numbers is 55
PS E:\Anirudh\Anirudh\CET\SEM 3\OOP_Lab\Java_cycle_3>
```