Experiment – 10

File Handling

Date of Submission: 23-10-2020

Aim: Write a Java program that read from a file and write to file by handling all file related exceptions.

Concepts Used: File Input and output, exception handling

Algorithm:

1. Start
2. import java.io package
3. fileName = “file.txt”
4. File f = new File(fileName)
5. if(!.f.exists()) then
6. f.createNewFile()
7. endif
8.
9. try
10. FileReader fr = new FileReader(file)
11. File copy = File(“copy.txt”)
12. if(!copy.exists()) then
13. copy.createNewFile()
14. endif
15. FileWriter fw = new FileWriter(copy)
16. while i=fr.read() and i!=-1 do
17. fw.write(i)
18. endwhile
19. endtry
20. catch FileNotFoundException e
21. Print “File is not found”
22. endcatch
23. Stop
24.

Result: The program was compiled successfully and the required output was obtained

Program Code:

```
/* Java Program to read and write to a file
 * by: Rohit Karunakaran
 *
 */

import java.io.*;


public class ReadWriteToFile
{
    public static void main(String args[]) throws IOException,
FileNotFoundException
    {

        //BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        int i;
        String fileName = "file.txt";
        File originalFile = new File(fileName);

        if(!originalFile.exists()) originalFile.createNewFile();

        try
        {
            FileReader fileReader1 = new FileReader(fileName);
            File copy = new File("copy.txt");

            if(!copy.exists())
            {
                copy.createNewFile();
            }

            try
            {
                FileWriter fileOutput = new FileWriter(copy);

                System.out.println("The String in the old file is : ");
                while ((i=fileReader1.read())!=-1)
                {
                    fileOutput.write((char)i);
                    System.out.print((char)i);
                }
                System.out.println(" ");
                fileOutput.flush();
                fileOutput.close();
            }
            catch (FileNotFoundException e)
            {
                System.out.println("The File is not writable or the file doesnt
exist");
                e.printStackTrace();
            }
            finally
            {
                fileReader1.close();
```

```
            }

            FileReader fileReader2 = new FileReader(copy);

            System.out.println("\nThe contents of the new file is :");
            while((i=fileReader2.read())!=-1)System.out.print((char)i);
            System.out.print("\n");

        }
        catch (FileNotFoundException e)
        {
            System.out.println("File is not found or the file is not readable "+e);
        }
    }
}
```

Sample Output:

```
The String in the old file is :
I am a file and I think there is a copy of me somewhere here


The contents of the new file is :
I am a file and I think there is a copy of me somewhere here
```

Experiment – 11

Console Input and Output

Date of Submission: 23-10-2020

Aim: Write a Java program that reads a line of integers, and then displays each integer, and the sum of all the integers (Use String Tokenizer class of java.util)

Concepts Used: String Tokenizer, Console input

Algorithm:
Input: A string containing of digits
Output: Sum of the digits

Steps

1. Start
2. import java.io package and StringTokenizer Class from java.util package
3. sum = 0
4. BufferedReader br = new BufferedReader(new InputStreamReader(System.in))
5. s = br.readLine()
6. StringTokenizer st = new StringTokenizer(s,”0123456789”,true)
7. while(st.hasMoreTokens())do
8. try
9. a = Integer.parseInt(st.nextToken())
10. sum+=a
11. endtry
12. catch NumberFormatException
13. print ”Number expected”
14. endcatch
15. endwhile
16. Stop

Reuslt: The program was successfully compiled and the required output was obtained

Program Code:

```
/* Read interger as a stirng and print it's sum
 * File Name: StringTokenizerExample.java
 *
 * Done By: Rohit Karunakaran
 * */

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

```
public class StringTokenizerExample
{
    public static void main(String args[]) throws IOException
    {
        int sum = 0,a=0;

        System.out.println("Enter an Integer Value");

        String s = new String(new BufferedReader(new
InputStreamReader(System.in)).readLine()); //reads a line from the input

        StringTokenizer st = new StringTokenizer(s,"0123456789",true); //if the
boolean argument is true each of the delimeter is considered as tokens themselves

        System.out.print("The Number entered = ");
        while(st.hasMoreTokens())
        {
            try
            {
                a = Integer.parseInt(st.nextToken());
                System.out.print(a);
            }
            catch(NumberFormatException e)
            {
                System.out.println("Expected an integer value");
                System.out.println("\nHere is the stack trace");

                e.printStackTrace();
            }

            sum+=a;
        }

        System.out.println("\nSum = "+sum);
    }
}
```

Sample Input 1:

```
10000000000000000000000000000000000000000000000000000000009
```

Sample output 1:

```
Enter an Integer Value : 10000000000000000000000000000000000000000000000000000000009

The Number entered = 10000000000000000000000000000000000000000000000000000000009
Sum = 10
```

Sample Input 2:

```
1329
```

Sample output 2:
Enter an Integer Value : 1329

The Number entered = 1329
Sum = 15

Sample input 3:
247298379237

Sample output 3
Enter an Integer Value : 247298379237

The Number entered = 247298379237
Sum = 63