

Java Assignment – 7

Vishwas Cp

Tecnotree Mysore

1. Create a program that reads in a text file and counts the number of words in the file. The program should display the total number of words at the end.

<https://codeshare.io/lonxmj>

The screenshot shows an IDE interface with a code editor and a terminal window. The code editor contains a Java file named 'countthenumber.java' with the following content:

```
1 package assignment7;
2 import java.io.BufferedReader;
3 import java.io.FileReader;
4 import java.io.IOException;
5
6 public class countthenumber {
7
8     public static void main(String[] args) {
9         // TODO Auto-generated method stub
10
11
12         try {
13             // open the file for reading
14             FileReader fileReader = new FileReader("C:\\Users\\chandvs\\OneDrive - Tecnotree\\Documents\\Assignment\\filename.txt");
15             BufferedReader bufferedReader = new BufferedReader(fileReader);
16             //the text in the filename.txt is Sanjana is a kind hearted Girl
17             // initialize the word count to zero
18             int wordCount = 0;
19
20             // read each line from the file and count the words
21             String line;
```

The terminal window below shows the output of the program execution:

```
<terminated> countthenumber [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (28-Feb-2023, 10:22:25 pm - 10:22:25 pm) [pid: 16368]
Total number of words: 6
```

2. Create a program that reads in two text files and compares them to see if they are identical. The program should display a message indicating whether the files are identical or not.

<https://codeshare.io/8plkr4>

The screenshot shows an IDE interface with a code editor and a terminal window. The code editor contains two Java files: 'countthenumber.java' and 'readtwofiles.java'. The 'readtwofiles.java' file has the following content:

```
1 package assignment7;
2 import java.io.BufferedReader;
3 import java.io.FileReader;
4 import java.io.IOException;
5
6 public class readtwofiles {
7
8     public static void main(String[] args) {
9         // TODO Auto-generated method stub
10
11         //In file1 and file2 the Content is vishwas
12         String file1 = "C:\\Users\\chandvs\\OneDrive - Tecnotree\\Documents\\Assignment\\file1.txt";
13         String file2 = "C:\\Users\\chandvs\\OneDrive - Tecnotree\\Documents\\Assignment\\file2.txt";
14         boolean identical = true;
15
16         try (BufferedReader reader1 = new BufferedReader(new FileReader(file1));
17              BufferedReader reader2 = new BufferedReader(new FileReader(file2))) {
18             String line1 = reader1.readLine();
19             String line2 = reader2.readLine();
20
21             while (line1 != null && line2 != null) {
```

The terminal window below shows the output of the program execution:

```
<terminated> readtwofiles [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (28-Feb-2023, 10:38:36 pm - 10:38:36 pm) [pid: 14668]
The files are identical.
```

- 3. Create a program that reads in a text file and creates a new file that contains the same text, but with all the vowels removed.**

<https://codeshare.io/mpbXeo>

The screenshot shows a Java IDE interface with two tabs: "readandcreate.java" and "Console".

Code (readandcreate.java):

```
1 package assignment;
2 import java.io.*;
3
4 public class readandcreate {
5
6     public static void main(String[] args) {
7         // In the input file content is sanjana is my close friend and in tecnotree i got new friends rachana hema ar
8         String inputFile = "C:\\Users\\chandvs\\OneDrive - Tecnotree\\Documents\\Assignment\\Input.txt";
9         String outputFile = "C:\\Users\\chandvs\\OneDrive - Tecnotree\\Documents\\Assignment\\Output.txt";
10    //After removing vowels in output file content is snjn s my cls frnd nd n tcntr gt nw frnds rchn hm nd rhn
11
12    try (BufferedReader reader = new BufferedReader(new FileReader(inputFile));
13        BufferedWriter writer = new BufferedWriter(new FileWriter(outputFile))) {
14        String line = reader.readLine();
15
16        while (line != null) {
17            String newLine = line.replaceAll("[aeiouAEIOU]", "");
18            writer.write(newLine);
19            writer.newLine();
20            line = reader.readLine();
21        }
22    }
23 }
```

Console Output:

```
<terminated> readandcreate [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (28-Feb-2023, 10:46:34 pm - 10:46:35 pm) [pid: 15496]
Vowels removed from C:\Users\chandvs\OneDrive - Tecnotree\Documents\Assignment\Input.txt and saved to C:\Users\chandvs\OneDrive - Tecnotree\Documents\Assignment\Output.txt
```

- 4. Create a program that reads in a CSV file containing student grades, and calculates the average grade for each student. The program should then write the results to a new CSV file.**

<https://codeshare.io/N3pQbd>

The screenshot shows a Java IDE interface with multiple tabs: Employee.java, Address.java, doublevariab..., doubleoint..., logicalAND.java, typestring.java, toupper.java, and readcsvfile....

Code (readcsvfile....):

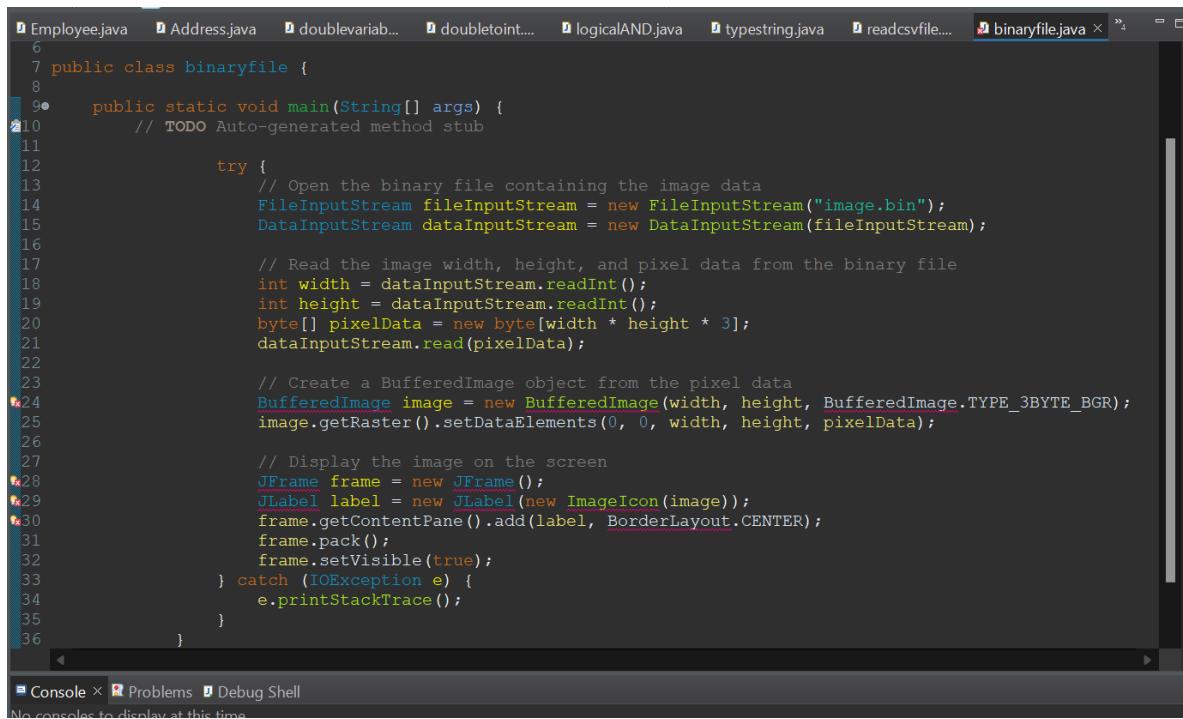
```
1 package assignment7;
2 import java.io.*;
3
4 public class readcsvfile {
5
6     public static void main(String[] args) throws IOException {
7
8         // Prompt the user to enter the number of students and the number of subjects
9         Scanner scanner = new Scanner(System.in);
10        System.out.print("Enter the number of students: ");
11        int numStudents = scanner.nextInt();
12        System.out.print("Enter the number of subjects: ");
13        int numSubjects = scanner.nextInt();
14
15        // Create a list to store the student grades
16        List<String[]> studentGrades = new ArrayList<>();
17
18        // Read the student grades from the user and calculate the average grade for each student
19        for (int i = 1; i <= numStudents; i++) {
20            System.out.print("Enter the grades for student " + i + ": ");
21            scanner.nextLine();
22            String name = "Student " + i;
23            String[] grades = scanner.nextLine().split(",");
24            double total = 0;
25            for (String grade : grades) {
26                total += Double.parseDouble(grade);
27            }
28            double average = total / numSubjects;
29            String[] data = {name, String.format("%.2f", average)};
30            studentGrades.add(data);
31
32        }
33    }
34 }
```

Console Output:

```
No consoles to display at this time.
```

5. Create a program that reads in a binary file containing image data, and displays the image on the screen.

<https://codeshare.io/0gvMVN>



The screenshot shows a Java code editor with the file `binaryfile.java` open. The code reads a binary file named `image.bin` and displays it in a window. The code uses `FileInputStream` and `DataInputStream` to read the width, height, and pixel data. It then creates a `BufferedImage` object and displays it in a `JFrame`. The code editor interface includes tabs for other files like `Employee.java`, `Address.java`, etc., and a bottom bar with `Console`, `Problems`, and `Debug Shell`.

```
1 Employee.java  2 Address.java  3 doublevariab...  4 doubletoint....  5 logicalAND.java  6 typestring.java  7 readcsvfile....  8 binaryfile.java × 9
6
7 public class binaryfile {
8
9     public static void main(String[] args) {
10         // TODO Auto-generated method stub
11
12         try {
13             // Open the binary file containing the image data
14             FileInputStream fileInputStream = new FileInputStream("image.bin");
15             DataInputStream dataInputStream = new DataInputStream(fileInputStream);
16
17             // Read the image width, height, and pixel data from the binary file
18             int width = dataInputStream.readInt();
19             int height = dataInputStream.readInt();
20             byte[] pixelData = new byte[width * height * 3];
21             dataInputStream.read(pixelData);
22
23             // Create a BufferedImage object from the pixel data
24             BufferedImage image = new BufferedImage(width, height, BufferedImage.TYPE_3BYTE_BGR);
25             image.getRaster().setDataElements(0, 0, width, height, pixelData);
26
27             // Display the image on the screen
28             JFrame frame = new JFrame();
29             JLabel label = new JLabel(new ImageIcon(image));
30             frame.getContentPane().add(label, BorderLayout.CENTER);
31             frame.pack();
32             frame.setVisible(true);
33         } catch (IOException e) {
34             e.printStackTrace();
35         }
36     }
}
```