

Name: Tejaswini Anil Kamble

Email : teju000kamble@gmail.com

Day 20 : Asssignments

Task 1: Java IO Basics

Write a program that reads a text file and counts the frequency of each word using FileReader and FileWriter.

Ans : Source Code

```
package com.wipro.day20;

import java.io.*;
import java.util.*;

public class WordFrequencyCounter {

    public static void main(String[] args) {

        String inputFilePath = "input.txt";
        String outputFilePath = "output.txt";
        Map<String, Integer> wordCountMap =
readFileAndCountWords(inputFilePath);
        writeWordFrequenciesToFile(wordCountMap, outputFilePath);
    }

    private static Map<String, Integer> readFileAndCountWords(String filePath) {

        Map<String, Integer> wordCountMap = new HashMap<>();
        try (FileReader fr = new FileReader(filePath); BufferedReader br = new
BufferedReader(fr)) {

            String line;
            while ((line = br.readLine()) != null) {

                String[] words = line.split("\\W+");
                for (String word : words) {

                    if (!word.isEmpty()) {

                        word = word.toLowerCase();
                        wordCountMap.put(word, wordCountMap.getOrDefault(word,
```

```

    0) + 1);
        }
    }
}

} catch (IOException e) {
    System.err.println("Error reading file: " + e.getMessage());
}

return wordCountMap;
}

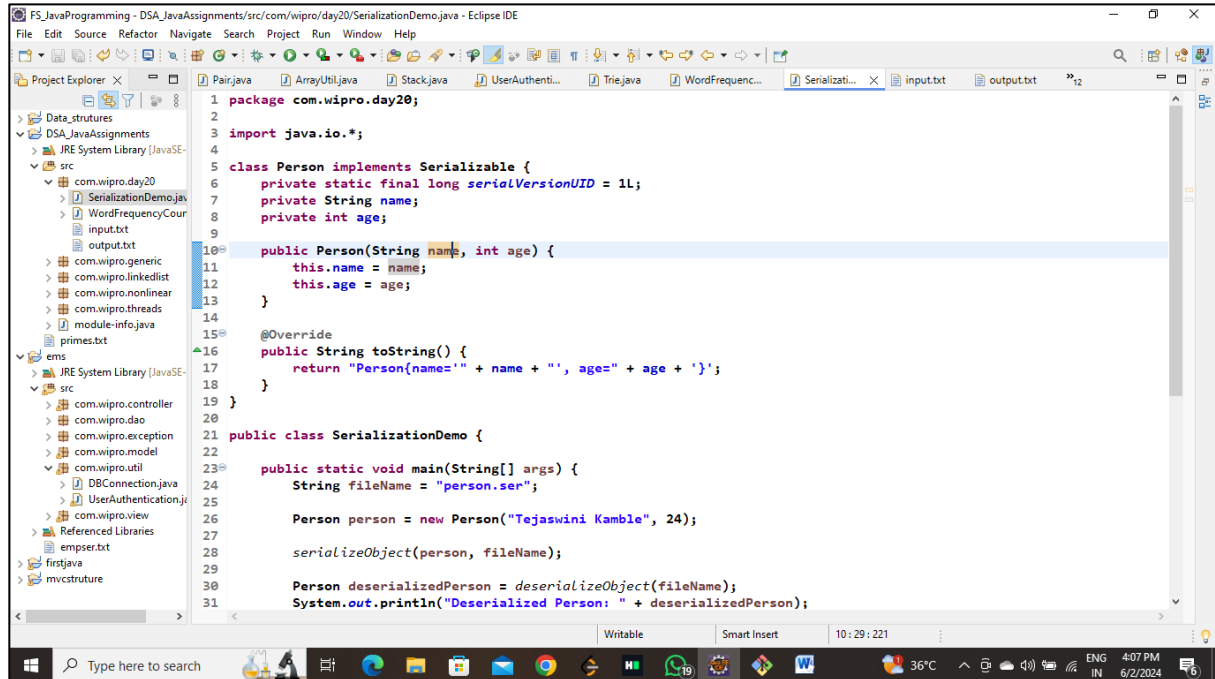
private static void writeWordFrequenciesToFile(Map<String, Integer>
wordCountMap, String filePath) {
    try (FileWriter fw = new FileWriter(filePath); BufferedWriter bw = new
BufferedWriter(fw)) {
        for (Map.Entry<String, Integer> entry : wordCountMap.entrySet()) {
            bw.write(entry.getKey() + ": " + entry.getValue());
            bw.newLine();
        }
    } catch (IOException e) {
        System.err.println("Error writing to file: " + e.getMessage());
    }
}
}

```

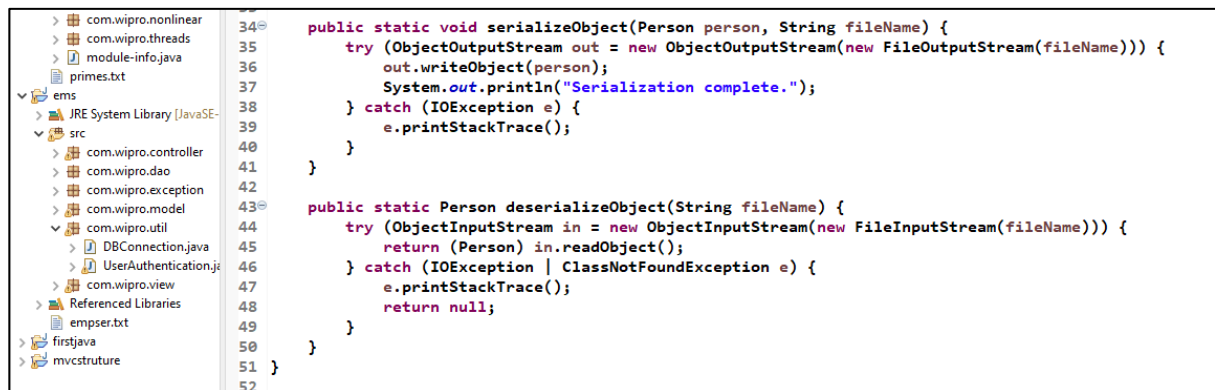
Task 2: Serialization and Deserialization

Serialize a custom object to a file and then deserialize it back to recover the object state.

Ans: Source code

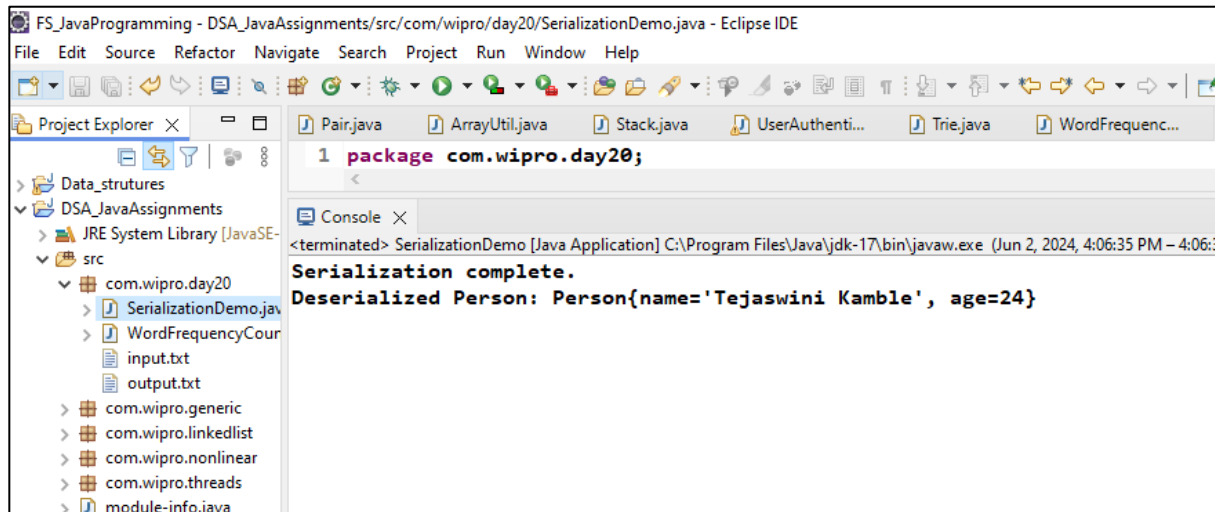


```
1 package com.wipro.day20;
2
3 import java.io.*;
4
5 class Person implements Serializable {
6     private static final long serialVersionUID = 1L;
7     private String name;
8     private int age;
9
10    public Person(String name, int age) {
11        this.name = name;
12        this.age = age;
13    }
14
15    @Override
16    public String toString() {
17        return "Person{name='" + name + "', age=" + age + '}';
18    }
19 }
20
21 public class SerializationDemo {
22
23    public static void main(String[] args) {
24        String fileName = "person.ser";
25
26        Person person = new Person("Tejaswini Kamble", 24);
27
28        serializeObject(person, fileName);
29
30        Person deserializedPerson = deserializeObject(fileName);
31        System.out.println("Deserialized Person: " + deserializedPerson);
32    }
33 }
```



```
34 public static void serializeObject(Person person, String fileName) {
35     try (ObjectOutputStream out = new ObjectOutputStream(new FileOutputStream(fileName))) {
36         out.writeObject(person);
37         System.out.println("Serialization complete.");
38     } catch (IOException e) {
39         e.printStackTrace();
40     }
41 }
42
43 public static Person deserializeObject(String fileName) {
44     try (ObjectInputStream in = new ObjectInputStream(new FileInputStream(fileName))) {
45         return (Person) in.readObject();
46     } catch (IOException | ClassNotFoundException e) {
47         e.printStackTrace();
48         return null;
49     }
50 }
51 }
52 }
```

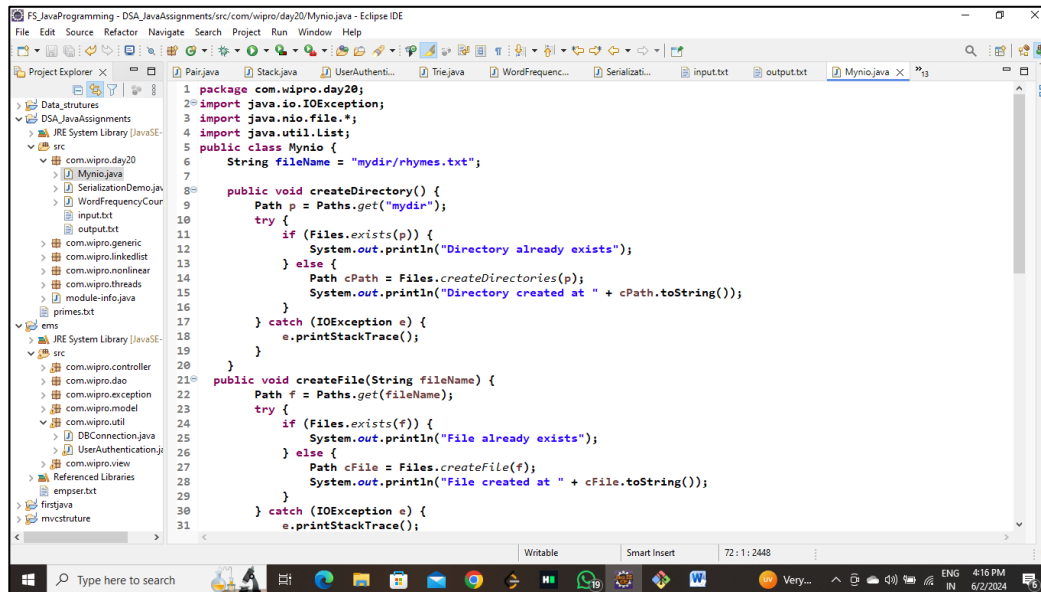
Output:



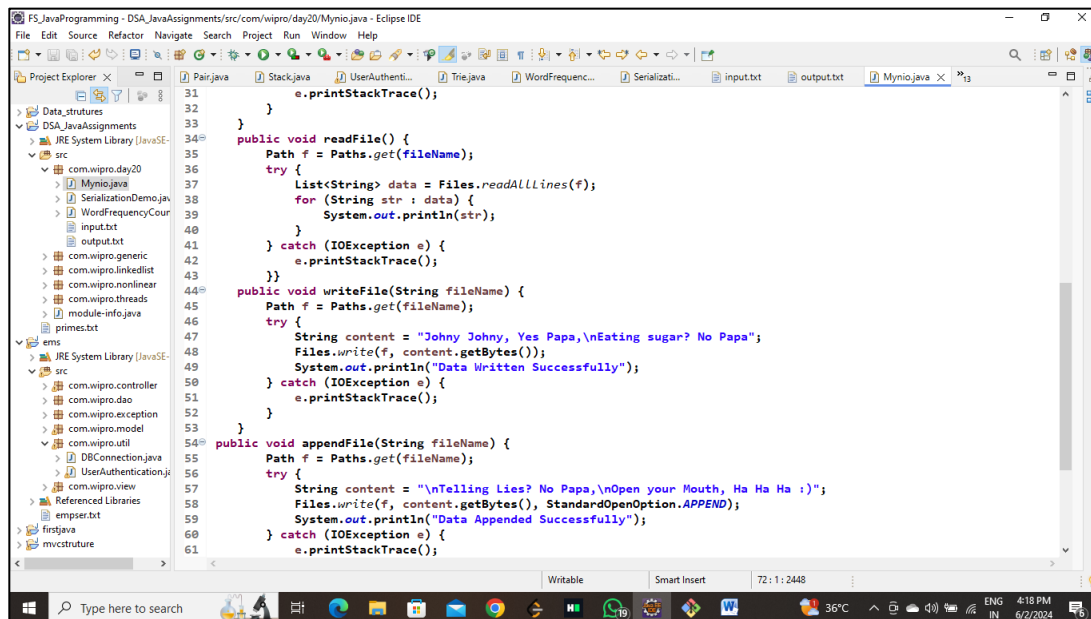
```
<terminated> SerializationDemo [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (Jun 2, 2024, 4:06:35 PM - 4:06:35 PM)
Serialization complete.
Deserialized Person: Person{name='Tejaswini Kamble', age=24}
```

Task 3: New IO (NIO)

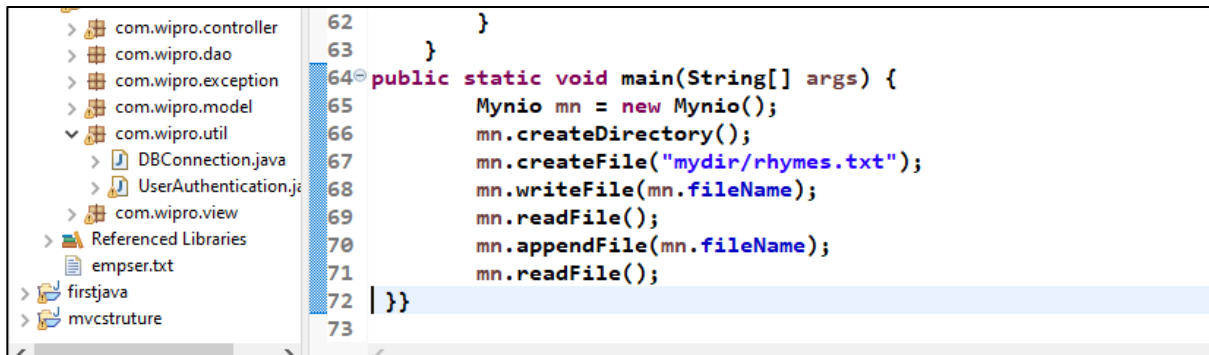
Use NIO Channels and Buffers to read content from a file and write to another file.



```
1 package com.wipro.day20;
2 import java.io.IOException;
3 import java.nio.file.*;
4 import java.util.List;
5 public class Mynio {
6     String fileName = "mydir/rhymes.txt";
7
8     public void createDirectory() {
9         Path p = Paths.get("mydir");
10        try {
11            if (Files.exists(p)) {
12                System.out.println("Directory already exists");
13            } else {
14                Path cPath = Files.createDirectories(p);
15                System.out.println("Directory created at " + cPath.toString());
16            }
17        } catch (IOException e) {
18            e.printStackTrace();
19        }
20    }
21
22    public void createFile(String fileName) {
23        Path f = Paths.get(fileName);
24        try {
25            if (Files.exists(f)) {
26                System.out.println("File already exists");
27            } else {
28                Path cFile = Files.createFile(f);
29                System.out.println("File created at " + cFile.toString());
30            }
31        } catch (IOException e) {
32            e.printStackTrace();
33        }
34    }
35}
```

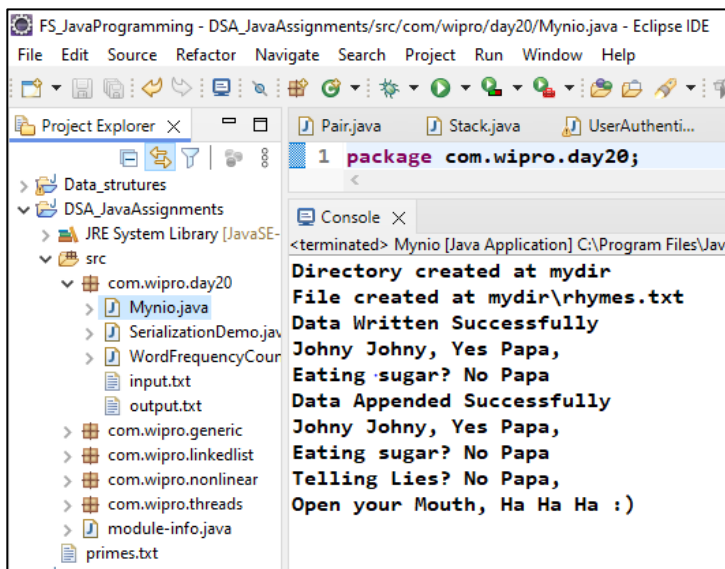


```
31        e.printStackTrace();
32    }
33
34    public void readFile() {
35        Path f = Paths.get(fileName);
36        try {
37            List<String> data = Files.readAllLines(f);
38            for (String str : data) {
39                System.out.println(str);
40            }
41        } catch (IOException e) {
42            e.printStackTrace();
43        }
44    }
45
46    public void writeFile(String fileName) {
47        Path f = Paths.get(fileName);
48        try {
49            String content = "Johnny Johnny, Yes Papa,\nEating sugar? No Papa";
50            Files.write(f, content.getBytes());
51            System.out.println("Data Written Successfully");
52        } catch (IOException e) {
53            e.printStackTrace();
54        }
55    }
56
57    public void appendFile(String fileName) {
58        Path f = Paths.get(fileName);
59        try {
60            String content = "\nTelling Lies? No Papa,\nOpen your Mouth, Ha Ha Ha :);";
61            Files.write(f, content.getBytes(), StandardOpenOption.APPEND);
62            System.out.println("Data Appended Successfully");
63        } catch (IOException e) {
64            e.printStackTrace();
65        }
66    }
67}
```



```
62     }
63 }
64 public static void main(String[] args) {
65     Mynio mn = new Mynio();
66     mn.createDirectory();
67     mn.createFile("mydir/rhymes.txt");
68     mn.writeFile(mn.fileName);
69     mn.readFile();
70     mn.appendFile(mn.fileName);
71     mn.readFile();
72 }
73 }
```

Output:

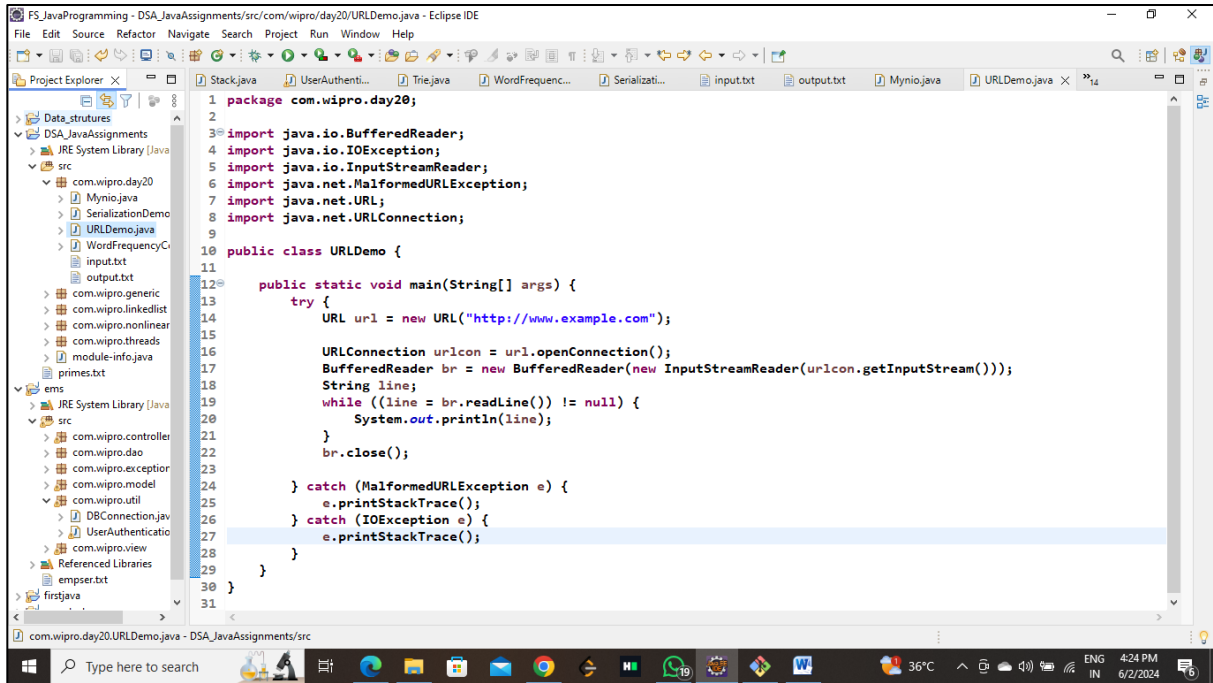


```
FS_JavaProgramming - DSA_JavaAssignments/src/com/wipro/day20/Mynio.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Pair.java Stack.java UserAuthenti...
1 package com.wipro.day20;
Console X
<terminated> Mynio [Java Application] C:\Program Files\Java
Directory created at mydir
File created at mydir\rhymes.txt
Data Written Successfully
Johnny Johnny, Yes Papa,
Eating sugar? No Papa
Data Appended Successfully
Johnny Johnny, Yes Papa,
Eating sugar? No Papa
Telling Lies? No Papa,
Open your Mouth, Ha Ha Ha :)
```

Task 4: Java Networking

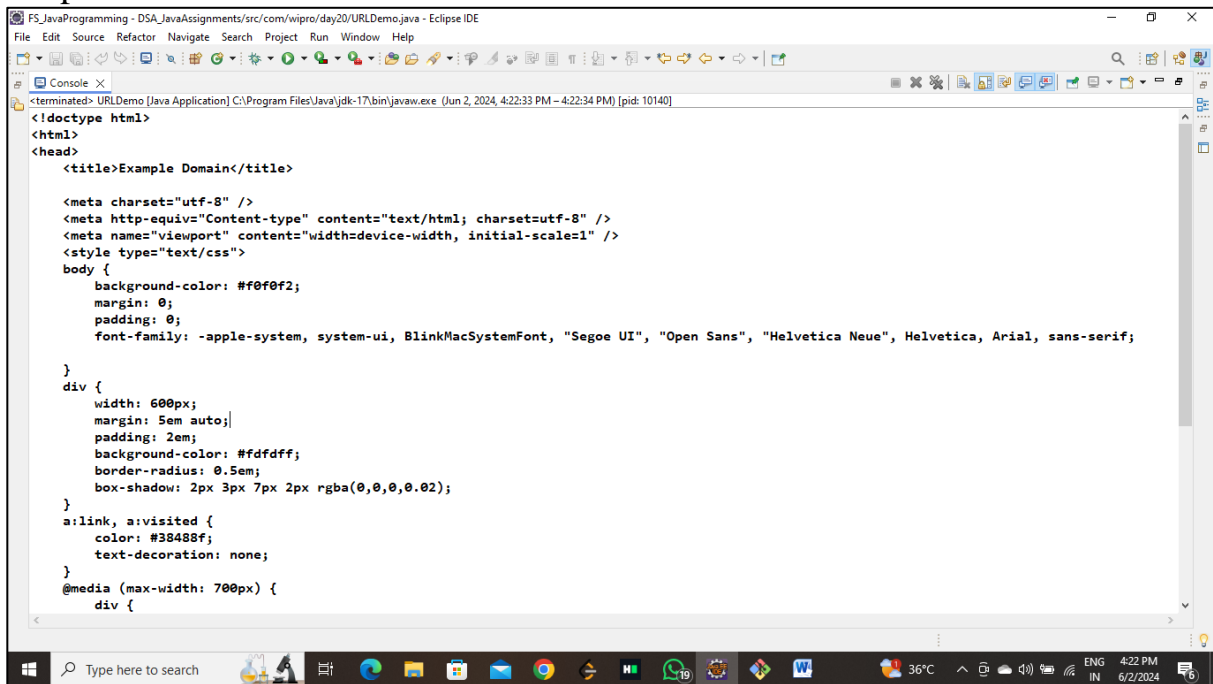
Write a simple HTTP client that connects to a URL, sends a request, and displays the response headers and body.

Ans: Sourec code



```
1 package com.wipro.day20;
2
3 import java.io.BufferedReader;
4 import java.io.IOException;
5 import java.io.InputStreamReader;
6 import java.net.MalformedURLException;
7 import java.net.URL;
8 import java.net.URLConnection;
9
10 public class URLDemo {
11
12     public static void main(String[] args) {
13         try {
14             URL url = new URL("http://www.example.com");
15
16             URLConnection urlcon = url.openConnection();
17             BufferedReader br = new BufferedReader(new InputStreamReader(urlcon.getInputStream()));
18             String line;
19             while ((line = br.readLine()) != null) {
20                 System.out.println(line);
21             }
22             br.close();
23
24         } catch (MalformedURLException e) {
25             e.printStackTrace();
26         } catch (IOException e) {
27             e.printStackTrace();
28         }
29     }
30 }
31
```

Output:



```
<terminated> URLDemo [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (Jun 2, 2024, 4:22:33 PM - 4:22:34 PM) [pid: 10140]
<!doctype html>
<html>
<head>
<title>Example Domain</title>

<meta charset="utf-8" />
<meta http-equiv="Content-type" content="text/html; charset=utf-8" />
<meta name="viewport" content="width=device-width, initial-scale=1" />
<style type="text/css">
body {
background-color: #f0f0f2;
margin: 0;
padding: 0;
font-family: -apple-system, system-ui, BlinkMacSystemFont, "Segoe UI", "Open Sans", "Helvetica Neue", Helvetica, Arial, sans-serif;
}
div {
width: 600px;
margin: 5em auto;
padding: 2em;
background-color: #fdfdff;
border-radius: 0.5em;
box-shadow: 2px 3px 7px rgba(0,0,0,0.02);
}
a:link, a:visited {
color: #38488f;
text-decoration: none;
}
@media (max-width: 700px) {
div {
width: auto;
}
}
</style>
</head>
<body>
<div>
<h1>Example Domain</h1>
<p>This domain is for use in illustrative examples in documents. You may use this
domain in literature without prior coordination or asking for permission.</p>
<p><a href="https://www.iana.org/domains/example">More information...</a></p>
</div>
</body>
</html>
```

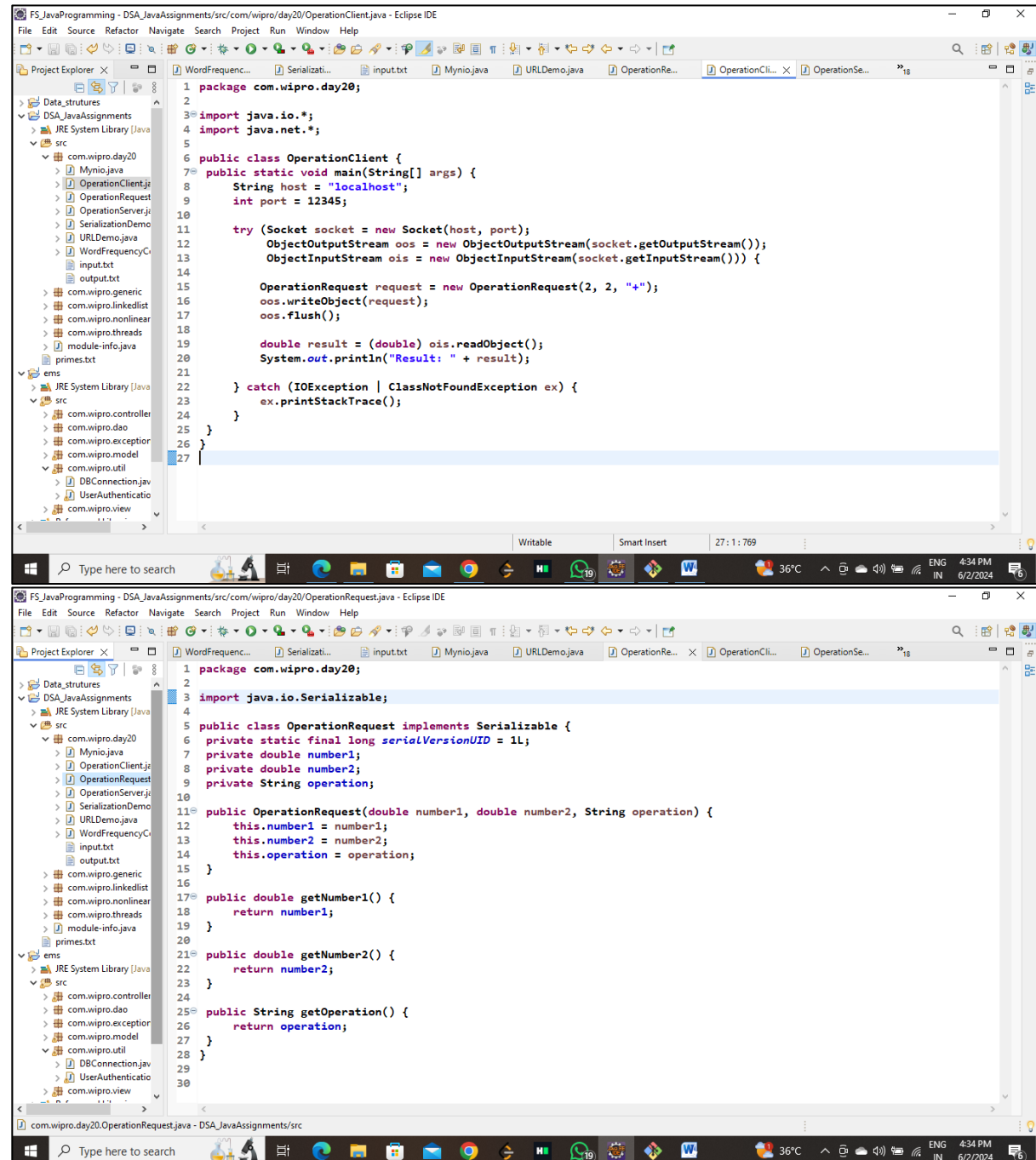
```
@media (max-width: 700px) {
div {
margin: 0 auto;
width: auto;
}
}
</style>
</head>
<body>
<div>
<h1>Example Domain</h1>
<p>This domain is for use in illustrative examples in documents. You may use this
domain in literature without prior coordination or asking for permission.</p>
<p><a href="https://www.iana.org/domains/example">More information...</a></p>
</div>
</body>
</html>
```

Task 5: Java Networking and Serialization

Develop a basic TCP client and server application where the client sends a serialized object with 2 numbers and operation to be performed on them to the server, and the server computes the result and sends it back to the client. for eg, we could send 2, 2, "+" which would mean 2 + 2

Operationrequest :

Ans : Source Code



The image displays two screenshots of the Eclipse IDE, showing the source code for a Java application. The top screenshot shows the `OperationClient.java` file, and the bottom screenshot shows the `OperationRequest.java` file.

OperationClient.java

```
1 package com.wipro.day20;
2
3 import java.io.*;
4 import java.net.*;
5
6 public class OperationClient {
7     public static void main(String[] args) {
8         String host = "localhost";
9         int port = 12345;
10
11         try (Socket socket = new Socket(host, port);
12             ObjectOutputStream oos = new ObjectOutputStream(socket.getOutputStream());
13             ObjectInputStream ois = new ObjectInputStream(socket.getInputStream())) {
14
15             OperationRequest request = new OperationRequest(2, 2, "+");
16             oos.writeObject(request);
17             oos.flush();
18
19             double result = (double) ois.readObject();
20             System.out.println("Result: " + result);
21
22         } catch (IOException | ClassNotFoundException ex) {
23             ex.printStackTrace();
24         }
25     }
26 }
27
```

OperationRequest.java

```
1 package com.wipro.day20;
2
3 import java.io.Serializable;
4
5 public class OperationRequest implements Serializable {
6     private static final long serialVersionUID = 1L;
7     private double number1;
8     private double number2;
9     private String operation;
10
11     public OperationRequest(double number1, double number2, String operation) {
12         this.number1 = number1;
13         this.number2 = number2;
14         this.operation = operation;
15     }
16
17     public double getNumber1() {
18         return number1;
19     }
20
21     public double getNumber2() {
22         return number2;
23     }
24
25     public String getOperation() {
26         return operation;
27     }
28 }
29
30
```



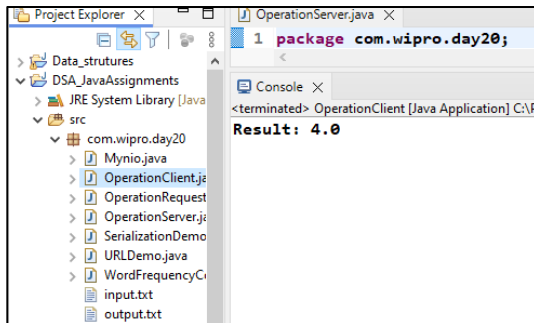
```
1 package com.wipro.day20;
2
3 import java.io.*;
4 import java.net.*;
5 public class OperationServer {
6     public static void main(String[] args) {
7         int port = 12345;
8
9         try (ServerSocket serverSocket = new ServerSocket(port)) {
10             System.out.println("Server is listening on port " + port);
11
12             while (true) {
13                 try (Socket socket = serverSocket.accept();
14                     ObjectInputStream ois = new ObjectInputStream(socket.getInputStream());
15                     ObjectOutputStream oos = new ObjectOutputStream(socket.getOutputStream())) {
16
17                     OperationRequest request = (OperationRequest) ois.readObject();
18                     double result = performOperation(request);
19
20                     oos.writeObject(result);
21                 } catch (IOException | ClassNotFoundException ex) {
22                     ex.printStackTrace();
23                 }
24             } catch (IOException ex) {
25                 ex.printStackTrace();
26             }
27         }
28     }
29     private static double performOperation(OperationRequest request) {
30         double number1 = request.getNumber1();
31         double number2 = request.getNumber2();
```

```
23     }
24 }
25 } catch (IOException ex) {
26     ex.printStackTrace();
27 }
28 }
29 private static double performOperation(OperationRequest request) {
30     double number1 = request.getNumber1();
31     double number2 = request.getNumber2();
32     String operation = request.getOperation();
33
34     switch (operation) {
35         case "+":
36             return number1 + number2;
37         case "-":
38             return number1 - number2;
39         case "*":
40             return number1 * number2;
41         case "/":
42             if (number2 != 0) {
43                 return number1 / number2;
44             } else {
45                 throw new IllegalArgumentException("Division by zero");
46             }
47         default:
48             throw new UnsupportedOperationException("Unsupported operation: " + operation);
49     }
50 }
51 }
52 }
53 }
```

Output:

```
1 package com.wipro.day20;

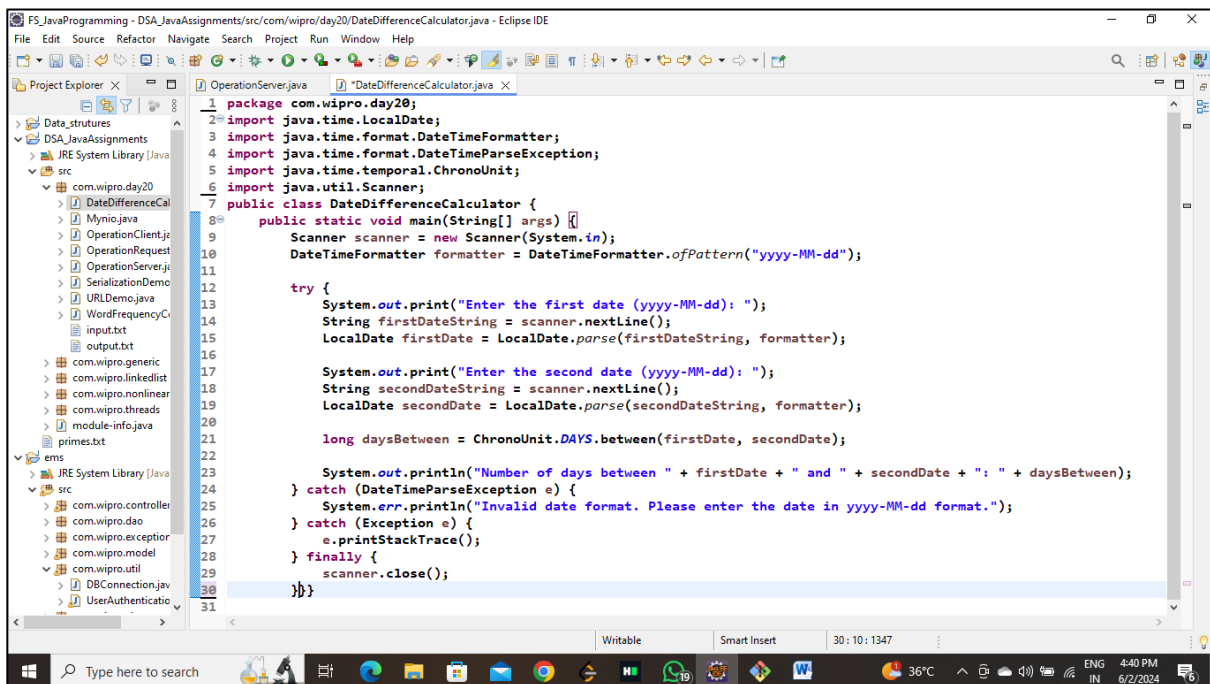
Console
OperationServer [Java Application] C:\Program Files\Java\jdk-11.0.10\bin\java.exe
Server is listening on port 12345
```

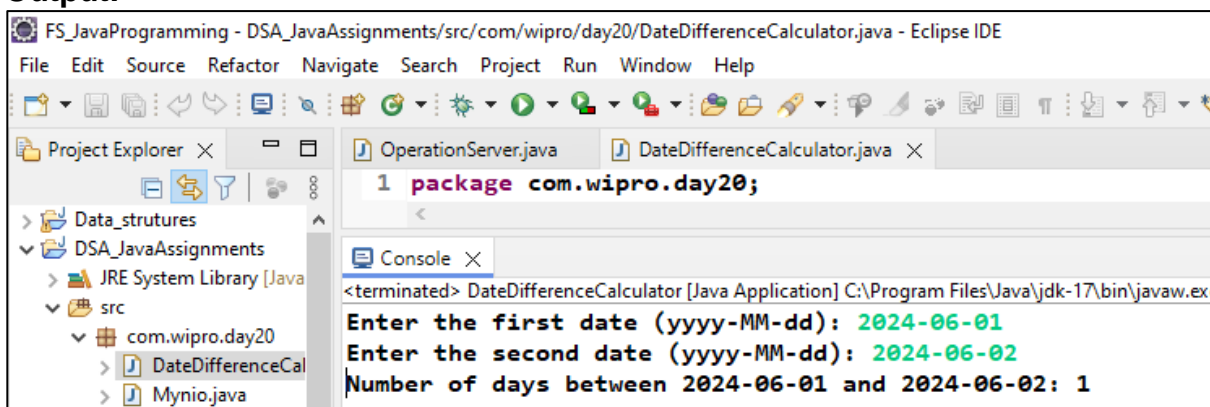
Task 6: Java 8 Date and Time API

Write a program that calculates the number of days between two dates input by the user.

Ans: Source Code



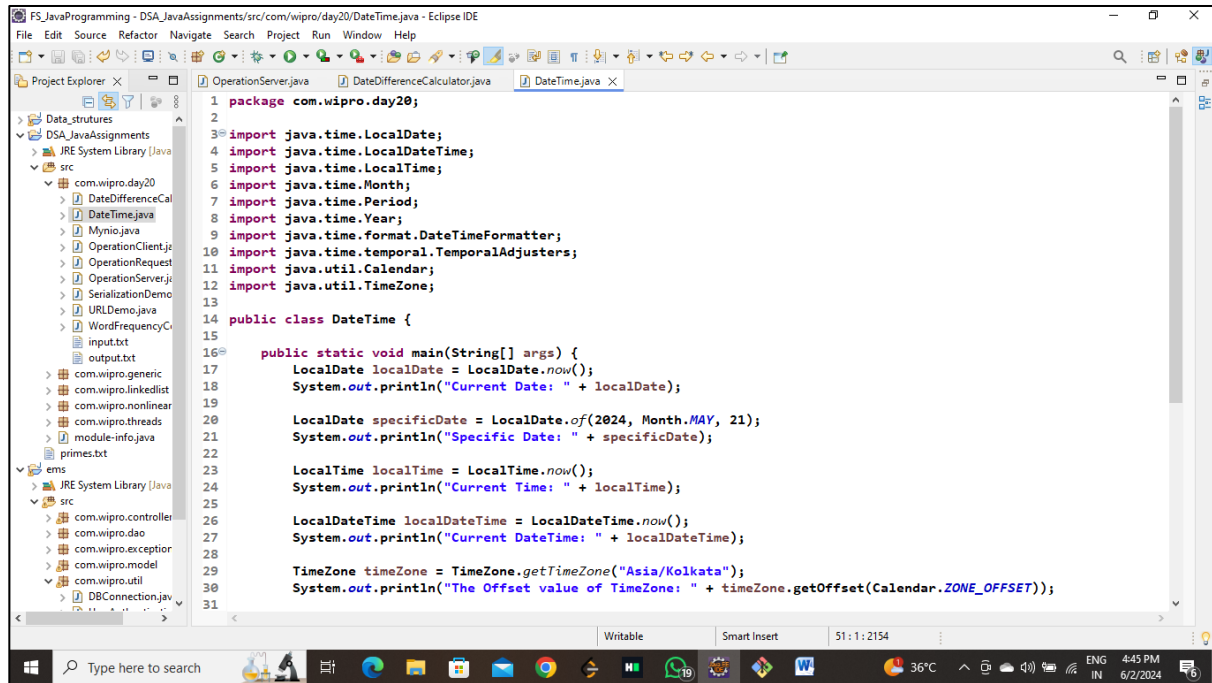
Output:



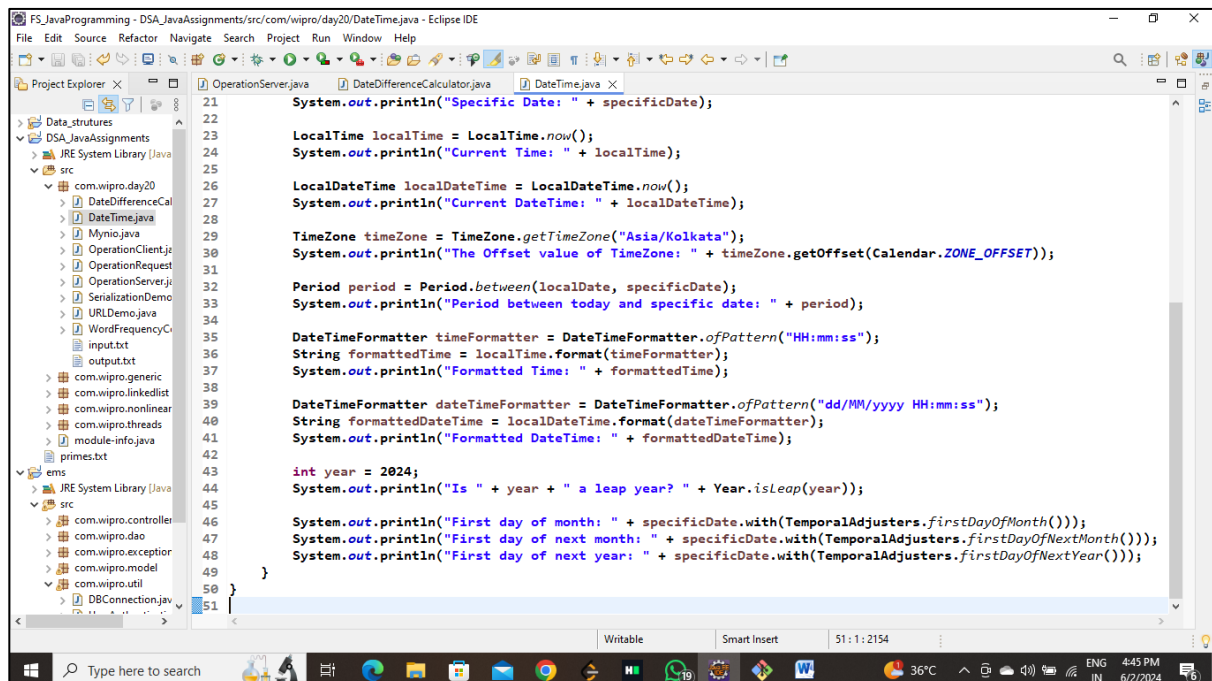
Task 7: Timezone

Create a timezone converter that takes a time in one timezone and converts it to another timezone

Ans: Source Code



```
1 package com.wipro.day20;
2
3 import java.time.LocalDate;
4 import java.time.LocalDateTime;
5 import java.time.LocalTime;
6 import java.time.Month;
7 import java.time.Period;
8 import java.time.Year;
9 import java.time.format.DateTimeFormatter;
10 import java.time.temporal.TemporalAdjusters;
11 import java.util.Calendar;
12 import java.util.TimeZone;
13
14 public class DateTime {
15
16     public static void main(String[] args) {
17         LocalDate localDate = LocalDate.now();
18         System.out.println("Current Date: " + localDate);
19
20         LocalDate specificDate = LocalDate.of(2024, Month.MAY, 21);
21         System.out.println("Specific Date: " + specificDate);
22
23         LocalTime localTime = LocalTime.now();
24         System.out.println("Current Time: " + localTime);
25
26         LocalDateTime localDateTime = LocalDateTime.now();
27         System.out.println("Current DateTime: " + localDateTime);
28
29         TimeZone timeZone = TimeZone.getTimeZone("Asia/Kolkata");
30         System.out.println("The Offset value of Timezone: " + timeZone.getOffset(Calendar.ZONE_OFFSET));
31     }
32 }
```



```
21     System.out.println("Specific Date: " + specificDate);
22
23     LocalTime localTime = LocalTime.now();
24     System.out.println("Current Time: " + localTime);
25
26     LocalDateTime localDateTime = LocalDateTime.now();
27     System.out.println("Current DateTime: " + localDateTime);
28
29     TimeZone timeZone = TimeZone.getTimeZone("Asia/Kolkata");
30     System.out.println("The Offset value of Timezone: " + timeZone.getOffset(Calendar.ZONE_OFFSET));
31
32     Period period = Period.between(localDate, specificDate);
33     System.out.println("Period between today and specific date: " + period);
34
35     DateTimeFormatter timeFormatter = DateTimeFormatter.ofPattern("HH:mm:ss");
36     String formattedTime = localTime.format(timeFormatter);
37     System.out.println("Formatted Time: " + formattedTime);
38
39     DateTimeFormatter dateTimeFormatter = DateTimeFormatter.ofPattern("dd/MM/yyyy HH:mm:ss");
40     String formattedDateTime = localDateTime.format(dateTimeFormatter);
41     System.out.println("Formatted DateTime: " + formattedDateTime);
42
43     int year = 2024;
44     System.out.println("Is " + year + " a leap year? " + Year.isLeap(year));
45
46     System.out.println("First day of month: " + specificDate.with(TemporalAdjusters.firstDayOfMonth()));
47     System.out.println("First day of next month: " + specificDate.with(TemporalAdjusters.firstDayOfNextMonth()));
48     System.out.println("First day of next year: " + specificDate.with(TemporalAdjusters.firstDayOfNextYear()));
49 }
50 }
```

Output:

