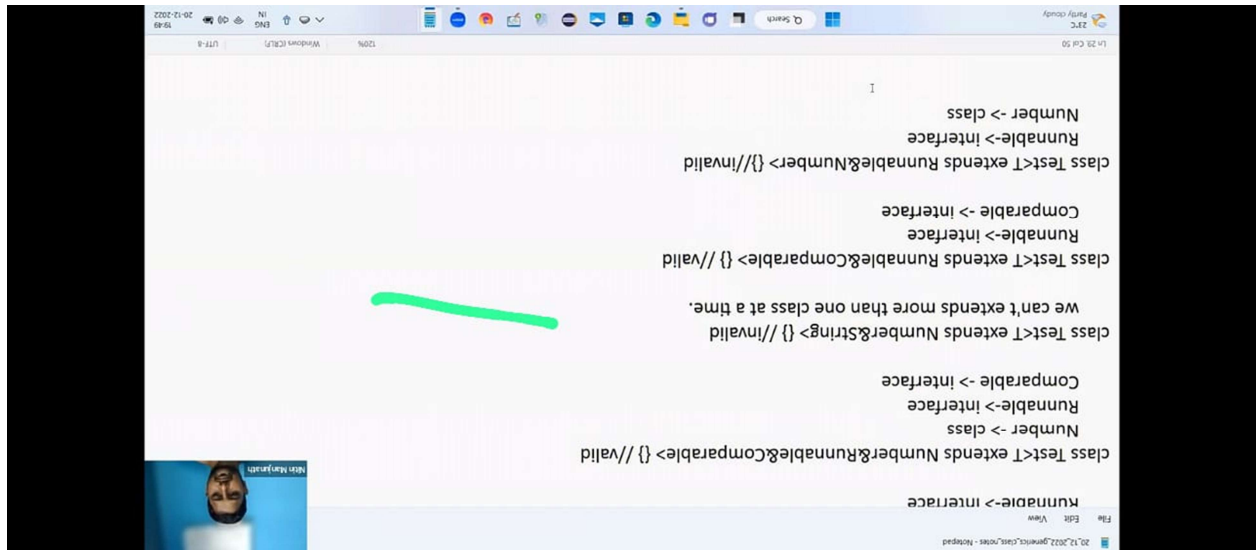


# 44Java\_IO\_Operation\_part1



### Runnable-> interface

class Test<T extends Number&Runnable&Comparable> {} //valid

Number -> class

Runnable-> interface

Comparable -> interface

class Test<T extends Number&String> {} //invalid  
we can't extends more than one class at a time.

class Test<T extends Runnable&Comparable> {} //valid

Runnable-> interface

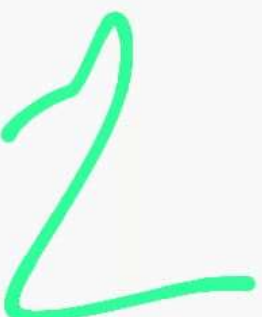
Comparable -> interface

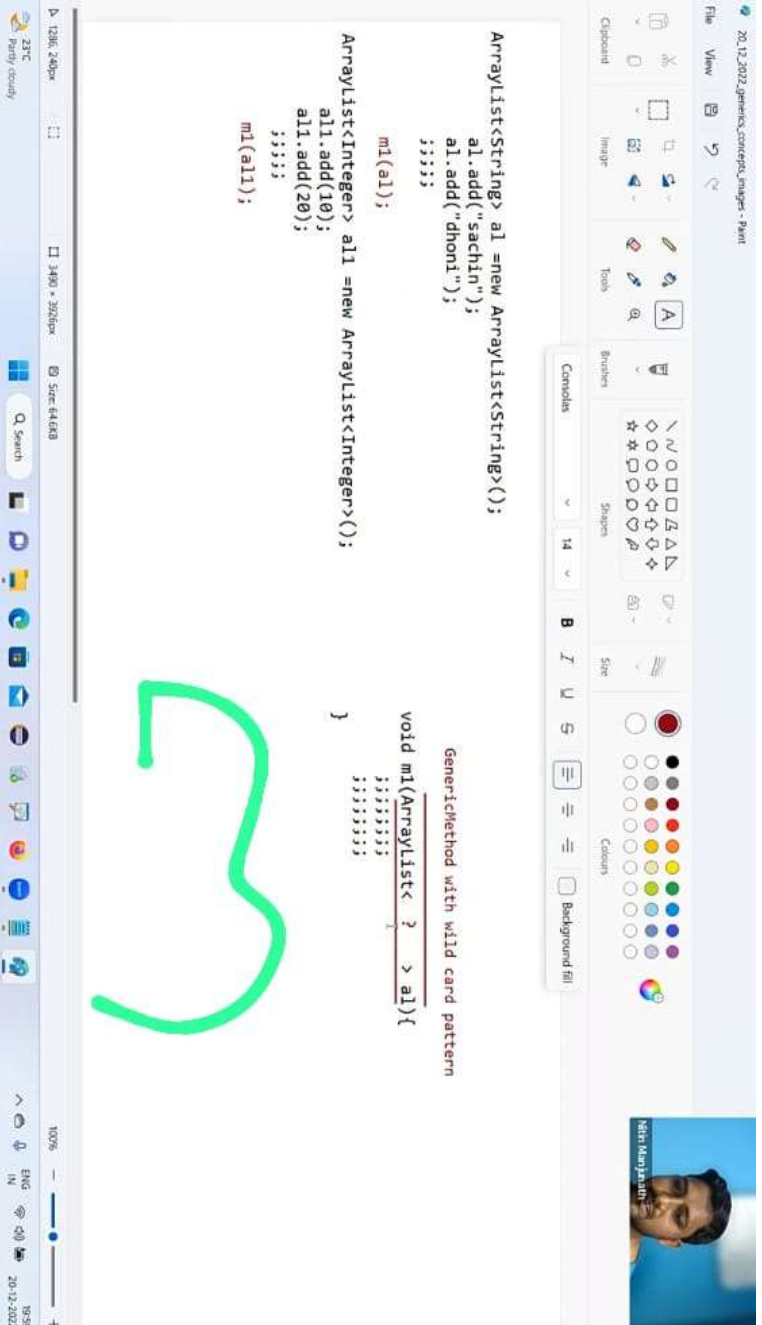
class Test<T extends Runnable&Number> {} //invalid

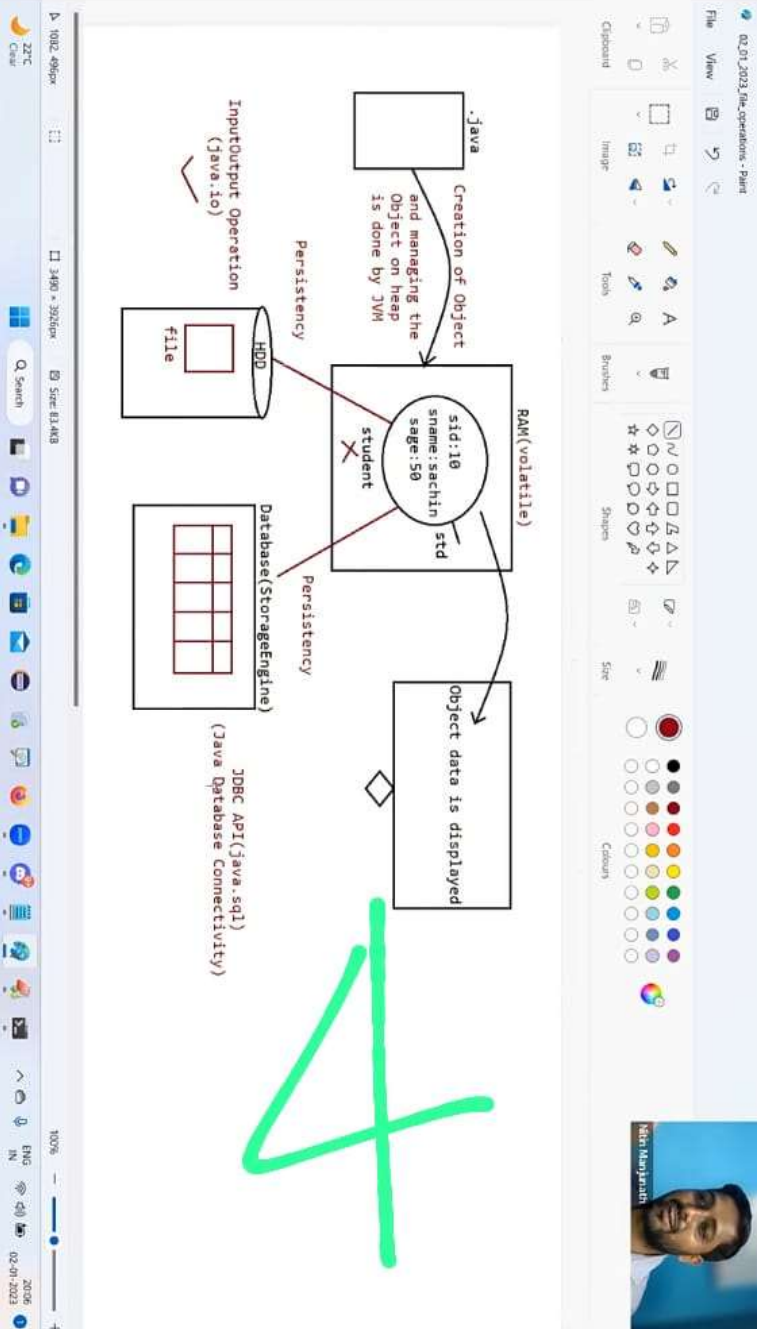
Runnable-> interface

Number -> class

rule: first inherit and the implement







### What is persistency?

It is a mechanism of storing the data permanently on to the file.

In java persistency can be achieved through an API's available inside package called "java.io".

### Input and Output Operations

=====

#### Agenda:

1. File ;
2. FileWriter
3. FileReader
4. BufferedWriter
5. BufferedReader

58



Command Prompt

D:\100operations>javap java.io.File

Compiled from "File.java"

```
public class java.io.File implements java.io.Serializable, java.lang.Comparable<java.io.File> {  
    public static final char separatorChar;  
    public static final java.lang.String separator;  
    public static final char pathSeparatorChar;  
    public static final java.lang.String pathSeparator;  
    static final boolean $assertionsDisabled;  
    final boolean isInvalid();  
    int getPrefixLength();  
    public java.io.File(java.lang.String);  
    public java.io.File(java.lang.String, java.lang.String);  
    public java.io.File(java.io.File, java.lang.String);  
    public java.io.File(java.net.URI);  
    public java.lang.String getName();  
    public java.lang.String getParent();  
    public java.io.File getParentFile();  
    public java.lang.String getPath();  
    public boolean isAbsolute();  
    public java.lang.String getAbsolutePath();  
    public java.io.File getAbsoluteFile();  
    public java.lang.String getCanonicalPath() throws java.io.IOException;  
    public java.io.File getCanonicalFile() throws java.io.IOException;  
    public java.net.URI toURI() throws java.net.MalformedURLException;  
    public java.net.URI toURL();  
    public boolean canRead();  
    public boolean canWrite();  
    public boolean exists();  
    public boolean isDirectory();  
    public boolean isFile();  
    public boolean isHidden();
```

Neha Hanjraani



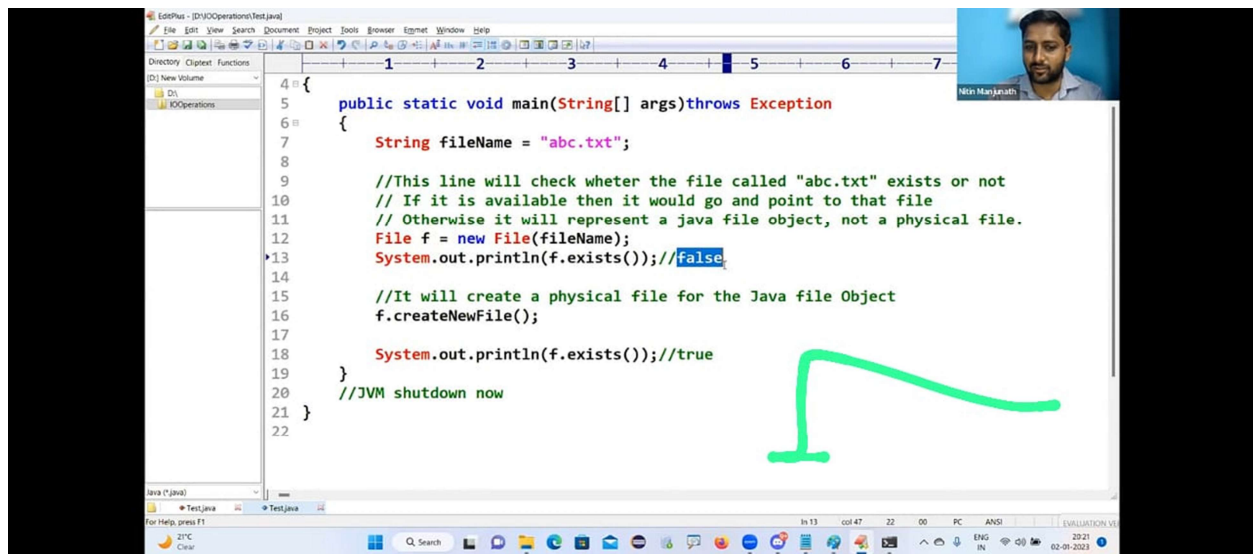
22°C  
Clear

Search



ENG  
IN

2011  
02-07-2023



The screenshot shows an IDE window titled "Eclipse - [D:\IOOperations\Test.java]". The code is as follows:

```
4 {
5     public static void main(String[] args) throws Exception
6     {
7         String fileName = "abc.txt";
8
9         //This line will check wheter the file called "abc.txt" exists or not
10        // If it is available then it would go and point to that file
11        // Otherwise it will represent a java file object, not a physical file.
12        File f = new File(fileName);
13        System.out.println(f.exists()); //False
14
15        //It will create a physical file for the Java file Object
16        f.createNewFile();
17
18        System.out.println(f.exists()); //true
19    }
20    //JVM shutdown now
21 }
22 }
```

A green bracket is drawn on the right side of the code, spanning from line 13 to line 18, indicating the logic flow from the initial check to the file creation and the subsequent check.

The IDE interface includes a "Directory" pane on the left showing "D:\New Volume" and "IOOperations". The bottom status bar shows "In 13 col 47 22 00 PC ANSI" and "EVALUATION SYSTEM". The system tray at the bottom indicates "20:21 02-01-2023".

Command Prompt

D:\IOoperations>javac Test.java

D:\IOoperations>java Test

false

true

D:\IOoperations>java Test

true

D:\IOoperations>



ZTC  
Cafe

Search

File Explorer

Edge

VS Code

PowerShell

Terminal

Git

Python

Java

JavaScript

PHP

MySQL

Android Studio

IntelliJ IDEA

PyCharm

NetBeans

Eclipse

2022

02-07-2023



## 5. BufferedReader

File:

```
File f=new File("abc.txt");
```

This line 1st checks whether abc.txt file is already available (or) not if it is already available then "f" simply refers that file. If it is not already available then it won't create any physical file just creates a java File object represents name of the file.

Example:

```
import java.io.*;
```

```
class FileDemo{
```

```
    public static void main(String[] args)throws IOException{
```

```
        File f=new File("cricket.txt");
```

```
        System.out.println(f.exists());//false
```

```
        f.createNewFile();
```

```
        System.out.println(f.exists());//true
```

```
    }
```

```
}
```

```
Ln 21, Col 1
```

27°C  
Clear

Q Search



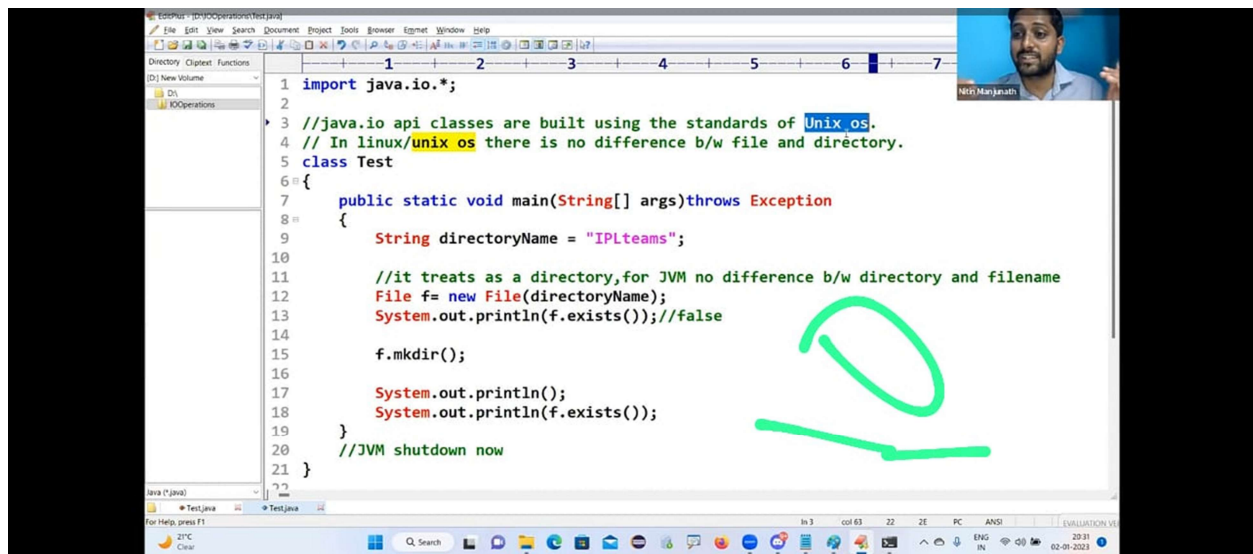
100%

Windows [CTRL]

UTF-8

2023  
02-01-2023





Command Prompt

X + -

D:\I00perations>javac Test.java

D:\I00perations>java Test


false

true

D:\I00perations>

2

Nishu Khanjani



27°C

Clear

Search

Q

Task View

File Explorer

Microsoft Edge

Google Chrome

WhatsApp

Telegram

Zoom

Skype

Discord

Spotify

Netflix

Amazon

Walmart

Target

Kroger

CVS

Walgreens

Costco

Sam's Club

Home Depot

Lowe's

Best Buy

Walmart

Target

Kroger

CVS

Walgreens

Costco

Sam's Club

Home Depot

Lowe's

Best Buy

20:31

02-07-2023

```
File f=new File("IPLTeams");
System.out.println(f.exists());//false
f.mkdir();//Creates a new directory
System.out.println(f.exists());//true
}
}
```

1st run  
=====

false  
true

2nd run  
=====

true  
true

Note: In UNIX everything is a file, java "file IO" is based on UNIX operating system  
hence in java also we can represent both files and directories by File object only.

12

Note: In UNIX everything is a file, java "file IO" is based on UNIX operating system  
hence in java also we can represent both files and directories by File object only.

#### File class constructors

=====

1. File f=new File(String name);

=> Creates a java File object that represents name of the file or directory in current working directory.

eg#1. `File f=new File("abc.txt");`

2. File f=new File(String subdirname,String name);

=> Creates a File object that represents name of the file or directory present in specified sub directory.

eg#1. `File f1=new File("abc");`

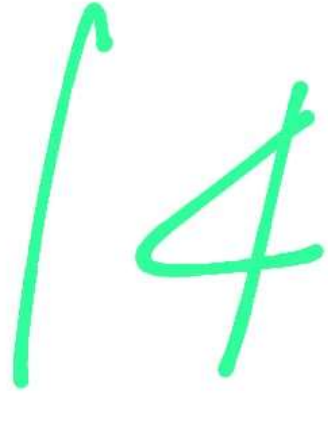
`f1.mkdir();`

`File f2=new File("abc","demo.txt");`



20  
21 //java.io-----> rt.jar(jdk s/w)  
22 class File  
23 {  
24 public boolean isFile()  
25 }  
26  
27 public boolean isDirectory()  
28 }  
29  
30 }  
31  
32 //hibernate-----> hiberante api  
33  
34 //springboot-----> pivotal team api  
35  
36 //microservices-----> pivotal team architecture based api  
37  
38

1 2 3 4 5 6 7



TestRunner - rt.jar(jdk s/w)

File Edit View Search Document Project Tools Browser Editmen Window Help

Directory Content Functions

D:\  
D:\Operations

Java ("java")


TestRunner

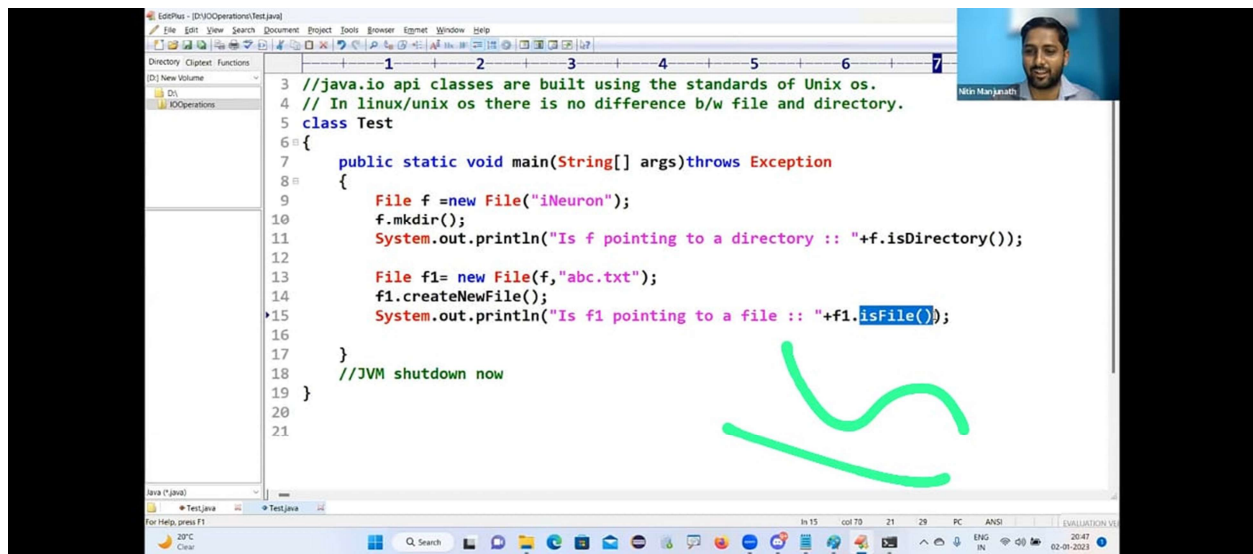
For Help, press F1

20°C Clear

ln 38 'cod 1' 7 00 PC ANSI 2024 02-01-2023 EVALUATION.VB

Narayan Narayan





The screenshot shows an IDE window titled "Eclipse - [D:\IOOperations\Test.java]". The code in the editor is as follows:

```
3 //java.io api classes are built using the standards of Unix os.
4 // In linux/unix os there is no difference b/w file and directory.
5 class Test
6 {
7     public static void main(String[] args) throws Exception
8     {
9         File f = new File("iNeuron");
10        f.mkdir();
11        System.out.println("Is f pointing to a directory :: "+f.isDirectory());
12
13        File f1 = new File(f, "abc.txt");
14        f1.createNewFile();
15        System.out.println("Is f1 pointing to a file :: "+f1.isFile());
16
17    }
18    //JVM shutdown now
19 }
20
21
```

A green squiggle is drawn on the right side of the code area, spanning from line 10 to line 18. In the top right corner, there is a small video feed of a man with a beard, wearing a blue shirt, with the name "Nitin Marjanath" below it. The IDE's bottom status bar shows "In 15 col 70 21 29 PC ANSI" and "EVALUATION VERSION". The system tray at the bottom shows the date "2024 02-01-2023" and the time "20:47".







Editor - (D:\Operations\test.java)  
File Edit View Search Document Project Tools Browser Export Window Help  
Directory Object Functions  
D:\New Volume  
D:\  
D:\Operations

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16

```
3 //no of characters present in file
4 ();//it deletes file or directory based on File Object
5 //it returns the names of all files and directories under specified directory
6
7 //no of characters present in file
8 ();//it deletes file or directory based on File Object
9 //it returns the names of all files and directories under specified directory
10
11 //no of characters present in file
12 ();//it deletes file or directory based on File Object
13 //it returns the names of all files and directories under specified directory
14
15 //no of characters present in file
16 ();//it deletes file or directory based on File Object
17 //it returns the names of all files and directories under specified directory
```

18

Test.java  
For Help, press F1  
20°C  
Cairo

ln 11 col 16 53 PC ANSI 20:58 02-07-2023 EVALUATION VER

Video thumbnail of a man speaking.

File Edit View

File Edit View

It returns the names of all files and subdirectories present in the specified directory.

7. long length();

Returns the no of characters present in the file.

8. boolean delete();

To delete a file or directory

Requirement: Write a program to display the names of all files and directories present in

D:\Java Job Guarantee Batch ?

61

Ln 166, Col 35

20°C  
Clear

Search



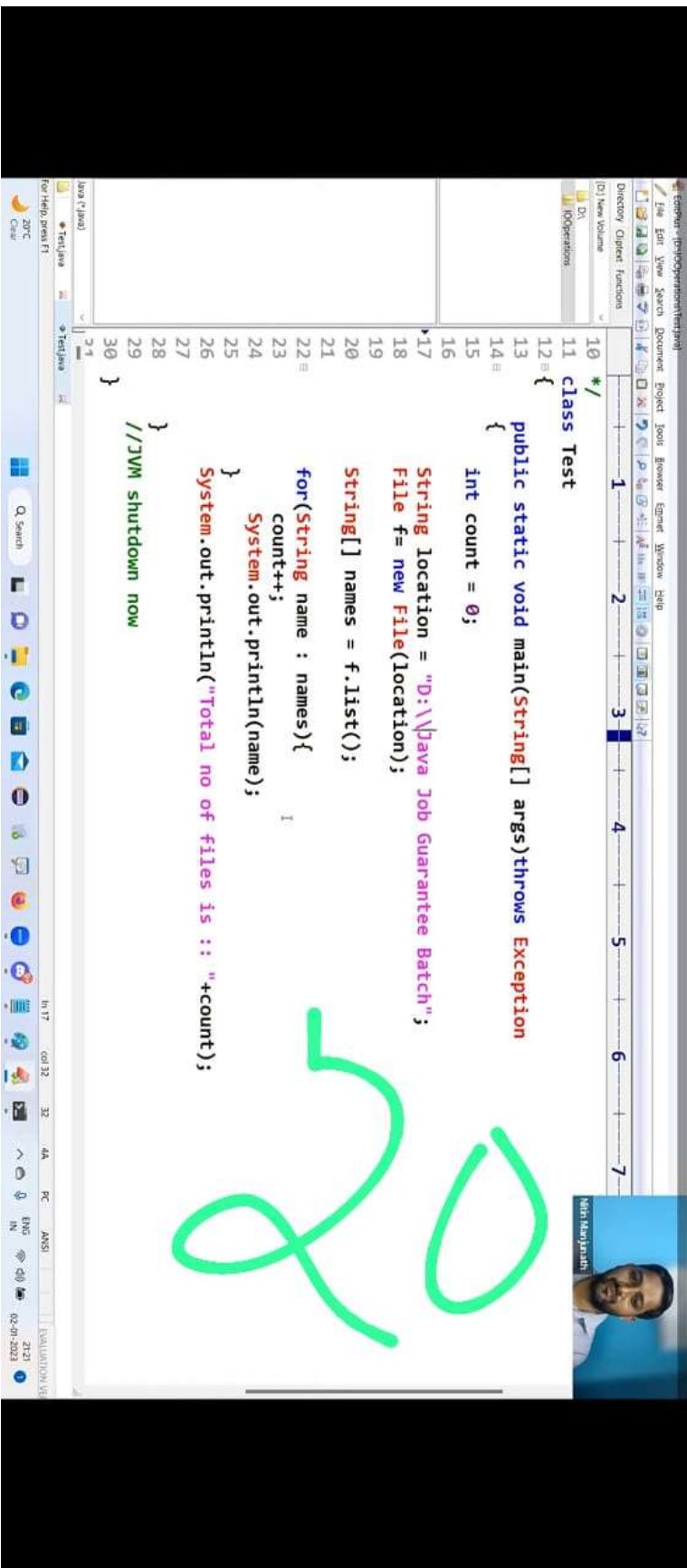
100%

Windows [CTRL]

UTF-8

ENG IN 02-07-2023





```
D:\IOoperations>javac Test.java
D:\IOoperations>java Test
01-PrimitiveValue-Injection
02-One-One-Association-mapping
02_01_2023_file_operations.png
02_01_2023_file_operations_class_notes.txt
03-One-MANY-Association-mapping
04-MANY-ONE-Association-mapping
05-MANY-MANY-Association-mapping
05_12_2022_threads_classnotes.txt
06_10_2022_QA_Discussion_notes.txt
07_10_2022_QA_Discussion_notes_job_gauranteee.txt
07_11_2022_StringBuilder_StringBuffer_image.png
07_12_2022_multithreading_class_notes.txt
07_12_2022_multithreading_images.png
07_12_2022_snippets_session_images.png
08_10_2022_QA_Discussion_notes_job_gauranteee.txt
08_12_2022_deadlock_images.png
08_12_2022_multithreading_notes.txt
09_12_2022_snippets_session_images.png
09_12_2022_snippets_session_notes.txt
10_10_2022_github_notes.txt
10_11_2022_Snippets_images.png
10_11_2022_Snippets_session_classnotes.txt
11_10_2022_github_notes.txt
11_11_2022_Snippets_images.png
11_11_2022_Snippets_session_classnotes.txt
12_10_2022_class_notes.txt
12_10_2022_QA_Discussion.txt
```

21



13 public static void main(String[] args)throws Exception  
14 {  
15 int dirCount = 0;  
16 int fileCount = 0;  
17  
18 String location = "D:\\Java Job Guarantee Batch";  
19 File f= new File(location);  
20  
21 String[] names = f.listFiles();  
22  
23 for(String name : names){  
24 // D:\\Java Job Guarantee Batch  
25 // all files we are iterating  
26 File f1 = new File(f,name);  
27  
28 if (f1.isDirectory())  
29 dirCount++;  
30  
31 if(f1.isFile())  
32 fileCount++;  
33  
34 System.out.println(name);  
35 }  
36 }

22

ANSI PC 22 col 56  
ENG IN  
21:29  
02-01-2023

File Edit View Search Document Project Tools Browser Editort Window Help

Directory Object Functions

D:\New Volume

IOOperations

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

File f= new File(location);

String[] names = f.list();

for(String name : names){

// D:\\Java Job Guarantee Batch

// all files we are iterating

File f1 = new File(f,name);

if (f1.isDirectory())

dirCount++;

if(f1.isFile()) {

fileCount++;

System.out.println(name);

}

System.out.println("Total no of files is :: "+fileCount);

System.out.println("Total no of Directory is :: "+dirCount);

}

// JVM shutdown now

Ln 31 Col 28

43

00

PC

ANSI

ENC

40

IN

02-01-2023

EVALUATION.VI

23

Neer Marjani







7:49 AM

File Edit View Search Database Project Tools Browser Engine Window Help

Directory Object Functions

D:\New Volume

12 {  
13 {  
14 {  
15 {  
16 {  
17 {  
18 {  
19 {  
20 {  
21 {  
22 {  
23 {  
24 {  
25 {  
26 {  
27 {  
28 {  
29 {  
30 {  
31 {  
32 {

```
public static void main(String[] args) throws Exception  
{  
    int dirCount = 0;  
    int jpgFileCount = 0;  
    int txtFileCount = 0;  
    int zipFileCount = 0;  
  
    String location = "D:\\Java Job Guarantee Batch";  
    File f = new File(location);  
  
    String[] names = f.list();  
  
    for(String name : names){  
        // D:\\Java Job Guarantee Batch  
        // all files we are iterating  
        File f1 = new File(f, name);  
  
        if (f1.isDirectory())  
            dirCount++;  
    }  
}
```

25

for help, press F1

20°C  
Clear

Q Search

Taskbar icons: File Explorer, Edge, Teams, OneDrive, etc.

System tray: In 18, 60% 30, 155, 00, PC, ANSI, ENG, IN, 02-07-2023, EVALUATION, VS

Video call window: Niranjan

Editor - (D:\Operations\test.java)

File Edit View Search Document Project Tools Browser Export Window Help

D:\New Volume

Directory Object Functions

D:\

Operations

29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
F9

1 2 3 4 5 6 7

```
if (f1.isDirectory())
    dirCount++;

if(f1.isFile()){
    if (name.endsWith(".jpg"))
        jpgFileCount++;
    if(name.endsWith(".txt"))
        txtFileCount++;
    if(name.endsWith(".zip"))
        zipFileCount++;
}

System.out.println(name);
}

System.out.println("Total no of Jpgfiles is :: "+jpgFileCount);
System.out.println("Total no of txtfiles is :: "+txtFileCount);
System.out.println("Total no of Zipfiles is :: "+zipFileCount);
System.out.println("Total no of Directory is :: "+dirCount);
```

Ln 41 Col 13 55 00 PC ANSI 2032 EVALUATION.VB  
For Help, press F1  
20°C  
Cairo

Search

Taskbar icons

System tray

20

Mr. Manjusha

20

```
6.12.2022_multithreading_class_notes.txt
6.12.2022_multithreading_images.png
6.12.2022_snippets_session_notes.txt
7.11.2022_snippets_session_notes.txt
7.11.2022_StringBuilder_StringBuffer_session_classnotes.txt
7.11.2022_String_snippets_session.png
8.11.2022_snippets_image.png
8.11.2022_snippets_session_classnotes.txt
9.11.2022_QA_discussion_session_classnotes.txt
9.11.2022_Snippets_images.png
9.11.2022_Snippets_session_classnotes.txt
code_snippets.png
DataTypes.png
Git Architecture.png
git integration with eclipse.txt
Github Introduction.png
GitServer Architecture.png
indentification of reserverwords.png
JVM Architecture.png
Local Version Control System.png
Object creation.png
personal access token.txt
Reserverwordchart.png
snippets solving question.png
Source Code.zip
SVN Vs Git.png
Total no of Jpegfiles is :: 0
Total no of txtfiles is :: 86
Total no of Zipfiles is :: 1
Total no of Directory is :: 5

D:\I00perations>
```

27




**File(c)**

On an existing file/directory we just performed the operations like

- a. isFile() b. isDirectory()
- c. length() d. exists()
- e. mkdir() f. createNewFile()
- g. list()

If we want to perform Reader operation on a File, then we need to use "FileReader".  
If we want to perform Write operation on a File, then we need to use "FileWriter".



Command Prompt

X + V

D:\IOOperations>javap java.io.FileWriter

Compiled from "FileWriter.java"

```
public class java.io.FileWriter extends java.io.OutputStreamWriter {  
    public java.io.FileWriter(java.lang.String) throws java.io.IOException;  
    public java.io.FileWriter(java.lang.String, boolean) throws java.io.IOException;  
    public java.io.FileWriter(java.io.File) throws java.io.IOException;  
    public java.io.FileWriter(java.io.File, boolean) throws java.io.IOException;  
}
```

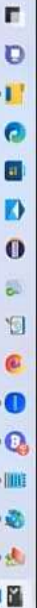
D:\IOOperations>

29



20°C  
Clear

Search



ENGL  
IN

21:44  
02-07-2023

```
D:\I00operations>javap java.io.FileWriter
Compiled from "FileWriter.java"
public class java.io.FileWriter extends java.io.OutputStreamWriter {
    public java.io.FileWriter(java.lang.String) throws java.io.IOException;
    public java.io.FileWriter(java.lang.String, boolean) throws java.io.IOException;
    public java.io.FileWriter(java.io.File) throws java.io.IOException;
    public java.io.FileWriter(java.io.File, boolean) throws java.io.IOException;
    public java.io.FileWriter(java.io.FileDescriptor)
    }
```

```
D:\I00operations>javap java.io.OutputStreamWriter
Compiled from "OutputStreamWriter.java"
public class java.io.OutputStreamWriter extends java.io.Writer {
    public java.io.OutputStreamWriter(java.io.OutputStream, java.lang.String) throws java.io.UnsupportedEncodingException;
    public java.io.OutputStreamWriter(java.io.OutputStream);
    public java.io.OutputStreamWriter(java.io.OutputStream, java.nio.charset.Charset);
    public java.io.OutputStreamWriter(java.io.OutputStream, java.nio.charset.Charset, Encoder);
    public java.lang.String getEncoding();
    void flushBuffer() throws java.io.IOException;
    public void write(int) throws java.io.IOException;
    public void write(char[], int, int) throws java.io.IOException;
    public void write(java.lang.String, int, int) throws java.io.IOException;
    public void flush() throws java.io.IOException;
    public void close() throws java.io.IOException;
    }
```

D:\I00operations>

30



Neha Hajjani





Mithun Manjunath

51


20°C  
Clear

D:\IO0perations>javac Test.java

D:\IO0perations>java Test

```
Compiled from "FileWriter.java"
public class java.io.FileWriter extends java.io.OutputStreamWriter {
    public java.io.FileWriter(java.lang.String) throws java.io.IOException;
    public java.io.FileWriter(java.lang.String, boolean) throws java.io.IOException;
    public java.io.FileWriter(java.io.File) throws java.io.IOException;
    public java.io.FileWriter(java.io.File, boolean) throws java.io.IOException;
}
```

```
D:\IO0perations>javap java.io.OutputStreamWriter
Compiled from "OutputStreamWriter.java"
public class java.io.OutputStreamWriter extends java.io.Writer {
    public java.io.OutputStreamWriter(java.io.OutputStream, java.lang.String) throws java.io.UnsupportedEncodingException;
    public java.io.OutputStreamWriter(java.io.OutputStream);
    public java.io.OutputStreamWriter(java.io.OutputStream, java.nio.charset.Charset);
    public java.io.OutputStreamWriter(java.io.OutputStream, java.nio.charset.Charset, Encoder);
    void flushBuffer() throws java.io.IOException;
    public void write(int) throws java.io.IOException;
    public void write(char[], int, int) throws java.io.IOException;
    public void flush() throws java.io.IOException;
    public void close() throws java.io.IOException;
}
```

D:\IO0perations>

32

20°C  
Borçlı Kar

Q Search

File Explorer

Microsoft Edge

Google Chrome

Firefox

Opera

Vivaldi

Brave

Chromium

WebKit

Gecko

Blink

Trident

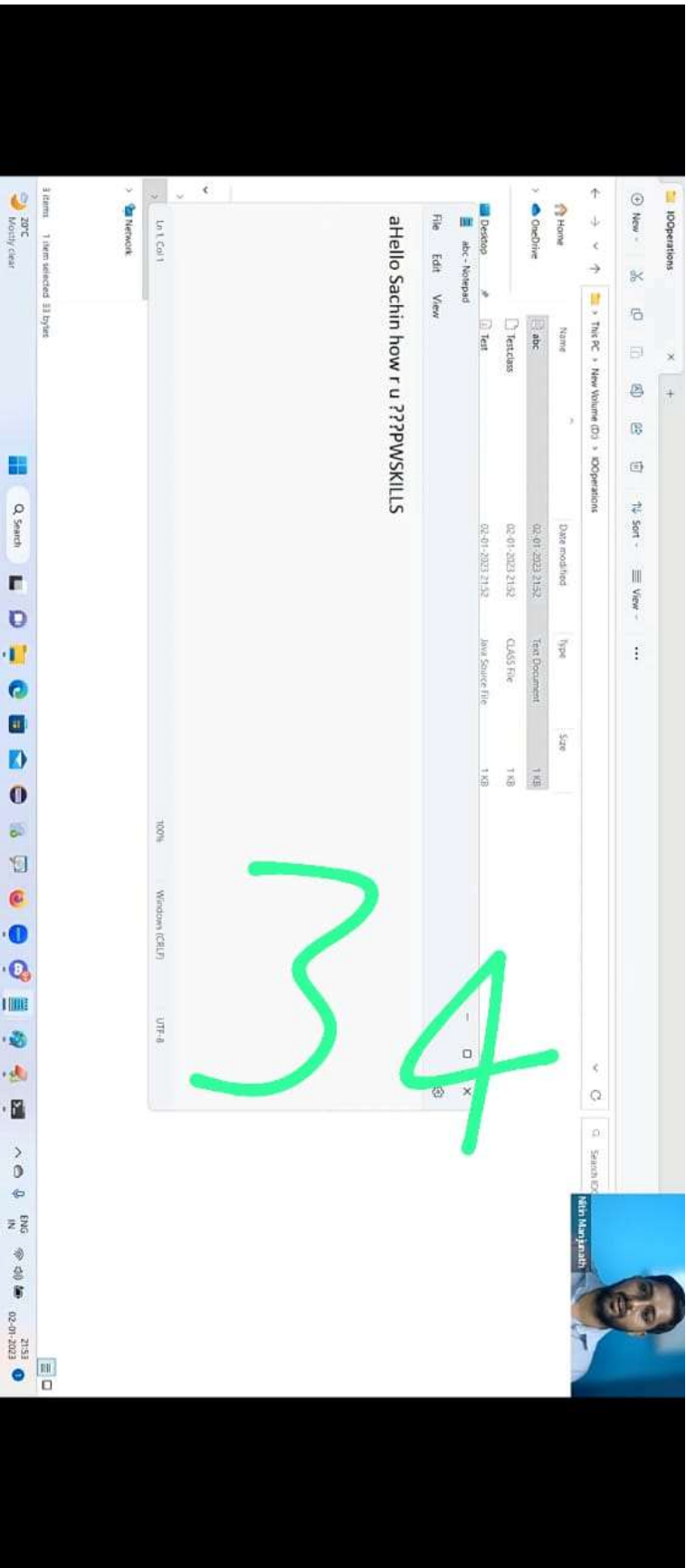
EdgeHTML

WebView

Android







```
4 class Test
5 {
6     public static void main(String[] args) throws Exception
7     {
8         File f1 = new File("abc.txt");
9         FileWriter f2 = new FileWriter(f1,true);
10
11         //performing write operation on a file
12         f2.write(97);//a
13         f2.write("\n");
14         f2.write("Hello Sachin how r u ???");
15         f2.write("\n");
16         char[] ch = {'P', 'W', 'S', 'K', 'I', 'L', 'L', 'S'};
17         f2.write(ch);
18
19         //closing the resource
20         f2.close();
21
22         System.out.println("open abc.txt to see the output");
23     }
24     //JVM shutdown now
}
```

SS

Editor - (D:\Operations\test.java)

File Edit View Search Document Project Tools Browser Export Window Help

Directory Object Functions

D:\New Volume

D:\Operations

7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28

```
//performing write operation on a file
f2.write("\n");
f2.write(97);//a
f2.write("\n");
f2.write("Hello Sachin how r u ???");
f2.write("\n");
char[] ch ={'P','W','S','K','I','L','L','S'};
f2.write(ch);

//making the data to write to the File
f2.flush();
f2.close();

System.out.println("open abc.txt to see the output");
}
//JVM shutdown now
}
```


test.java

Ln 8 Col 19 29 20 PC ANSI

For Help, press F1

20°C

Monday, 22-07-2023



58

Editor - (D:\Operations\test.java)

File Edit View Search Document Project Tools Browser Emmet Window Help

D:\New Volume

Directory Object Functions

D:\

IOOperations

1 import java.io.\*;

2

3 class Test

4 {

5     public static void main(String[] args)throws Exception

6     {

7         File f1 = new File("abc.txt");

8         FileWriter f2 = new FileWriter(f1,true);

9

10         //performing write operation on a file

11         f2.write("\n");

12         f2.write(97);//a

13         f2.write("\n");

14         f2.write("Hello Sachin how r u ???");

15         f2.write("\n");

16         char[] ch ={'P','W','S','K','I','L','L','S'};

17         f2.write(ch);

18

19         //making the data to write to the File

20         f2.flush();

21

22     }

23 }

Test.java

Test.java

Ln 8 Col 47 29 29 PC ANSI

For Help, press F1

20°C

Brillix Clear

Q Search

Ln 8 Col 47 29 29 PC ANSI

For Help, press F1

20°C

Brillix Clear

Q Search

Nitin Hajjajani

37

Command Prompt

D:\IOOperations>javap java.io.FileReader

Compiled from "FileReader.java"

```
public class java.io.FileReader extends java.io.InputStreamReader {  
    public java.io.FileReader(java.lang.String) throws java.io.FileNotFoundException;  
    public java.io.FileReader(java.io.File) throws java.io.FileNotFoundException;  
    public java.io.FileReader(java.io.FileDescriptor);  
}
```

D:\IOOperations>javap java.io.InputStreamReader

Compiled from "InputStreamReader.java"

```
public class java.io.InputStreamReader extends java.io.Reader {  
    public java.io.InputStreamReader(java.io.InputStream, java.lang.String) throws java.io.UnsupportedEncodingException;  
    public java.io.InputStreamReader(java.io.InputStream, java.nio.charset.Charset);  
    public java.io.InputStreamReader(java.io.InputStream, java.nio.charset.CharsetDecoder);  
    public java.lang.String getEncoding();  
    public int read() throws java.io.IOException;  
    public int read(char[], int, int) throws java.io.IOException;  
    public boolean ready() throws java.io.IOException;  
    public void close() throws java.io.IOException;  
}
```

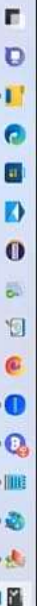
D:\IOOperations>



78

20°C  
Breezy clear

Search



ENG  
IN

22:17  
02-07-2023



Command Prompt

Compiled from "InputStreamReader.java"

```
public class java.io.InputStreamReader extends java.io.Reader {  
    public java.io.InputStreamReader(java.io.InputStream, java.lang.String) throws java.io.UnsupportedEncodingException;  
    public java.io.InputStreamReader(java.io.InputStream, java.nio.charset.Charset);  
    public java.io.InputStreamReader(java.io.InputStream, java.nio.charset.CharsetDecoder);  
    public java.lang.String getEncoding();  
    public int read() throws java.io.IOException;  
    public int read(char[], int, int) throws java.io.IOException;  
    public boolean ready() throws java.io.IOException;  
    public void close() throws java.io.IOException;  
}
```

D:\I00perations>javap java.io.Reader

Compiled from "Reader.java"

```
public abstract class java.io.Reader implements java.lang.Readable, java.io.Closeable {  
    protected java.lang.Object lock;  
    protected java.io.Reader();  
    protected java.io.Reader(java.lang.Object);  
    public int read(java.nio.CharBuffer) throws java.io.IOException;  
    public int read() throws java.io.IOException;  
    public int read(char[]) throws java.io.IOException;  
    public abstract int read(char[], int, int) throws java.io.IOException;  
    public long skip(long) throws java.io.IOException;  
    public boolean ready() throws java.io.IOException;  
    public boolean markSupported();  
    public void mark(int) throws java.io.IOException;  
    public void reset() throws java.io.IOException;  
    public abstract void close() throws java.io.IOException;  
}
```

D:\I00perations>

20°C  
Breezy Clear

Search

File Explorer

Edge

Task View

PowerShell

Visual Studio Code

File Explorer

Edge

Task View

PowerShell

Visual Studio Code

File Explorer

Edge

Task View

PowerShell

Visual Studio Code

File Explorer

Edge

Task View

PowerShell

Visual Studio Code

File Explorer

Edge

Task View

PowerShell

Visual Studio Code

File Explorer

Edge

Task View

PowerShell

Visual Studio Code



Shah Mansoor

65

File Edit View Search Document Project Tools Browser Bookmark Window Help

Directory Object Functions

D:\New Volume

D:\

IOOperations

1 import java.io.\*;

2

3 class Test

4 {

5     public static void main(String[] args) throws Exception

6     {

7         FileReader f1 = new FileReader("sachin.txt");

8         int i = f1.read();

9

10         while (i != -1)

11         {

12             System.out.print((char)i + " ==> ");

13             i = f1.read();

14             System.out.println(i);

15         }

16

17     }

18     //JVM shutdown now

19     }

20 }

21 }

22 }

23 }

24 }

25 }

26 }

27 }

28 }

29 }

30 }

31 }

32 }

33 }

34 }

35 }

36 }

37 }

38 }

39 }

40 }

41 }

42 }

43 }

44 }

45 }

46 }

47 }

48 }

49 }

50 }

51 }

52 }

53 }

54 }

55 }

56 }

57 }

58 }

59 }

60 }

61 }

62 }

63 }

64 }

65 }

66 }

67 }

68 }

69 }

70 }

71 }

72 }

73 }

74 }

75 }

76 }

77 }

78 }

79 }

80 }

81 }

82 }

83 }

84 }

85 }

86 }

87 }

88 }

89 }

90 }

91 }

92 }

93 }

94 }

95 }

96 }

97 }

98 }

99 }

100 }

101 }

102 }

103 }

104 }

105 }

106 }

107 }

108 }

109 }

110 }

111 }

112 }

113 }

114 }

115 }

116 }

117 }

118 }

119 }

120 }

121 }

122 }

123 }

124 }

125 }

126 }

127 }

128 }

129 }

130 }

131 }

132 }

133 }

134 }

135 }

136 }

137 }

138 }

139 }

140 }

141 }

142 }

143 }

144 }

145 }

146 }

147 }

148 }

149 }

150 }

151 }

152 }

153 }

154 }

155 }

156 }

157 }

158 }

159 }

160 }

161 }

162 }

163 }

164 }

165 }

166 }

167 }

168 }

169 }

170 }

171 }

172 }

173 }

174 }

175 }

176 }

177 }

178 }

179 }

180 }

181 }

182 }

183 }

184 }

185 }

186 }

187 }

188 }

189 }

190 }

191 }

192 }

193 }

194 }

195 }

196 }

197 }

198 }

199 }

200 }

201 }

202 }

203 }

204 }

205 }

206 }

207 }

208 }

209 }

210 }

211 }

212 }

213 }

214 }

215 }

216 }

217 }

218 }

219 }

220 }

221 }

222 }

223 }

224 }

225 }

226 }

227 }

228 }

229 }

230 }

231 }

232 }

233 }

234 }

235 }

236 }

237 }

238 }

239 }

240 }

241 }

242 }

243 }

244 }

245 }

246 }

247 }

248 }

249 }

250 }

251 }

252 }

253 }

254 }

255 }

256 }

257 }

258 }

259 }

260 }

261 }

262 }

263 }

264 }

265 }

266 }

267 }

268 }

269 }

270 }

271 }

272 }

273 }

274 }

275 }

276 }

277 }

278 }

279 }

280 }

281 }

282 }

283 }

284 }

285 }

286 }

287 }

288 }

289 }

290 }

291 }

292 }

293 }

294 }

295 }

296 }

297 }

298 }

299 }

300 }

301 }

302 }

303 }

304 }

305 }

306 }

307 }

308 }

309 }

310 }

311 }

312 }

313 }

314 }

315 }

316 }

317 }

318 }

319 }

320 }

321 }

322 }

323 }

324 }

325 }

326 }

327 }

328 }

329 }

330 }

331 }

332 }

333 }

334 }

335 }

336 }

337 }

338 }

339 }

340 }

341 }

342 }

343 }

344 }

345 }

346 }

347 }

348 }

349 }

350 }

351 }

352 }

353 }

354 }

355 }

356 }

357 }

358 }

359 }

360 }

361 }

362 }

363 }

364 }

365 }

366 }

367 }

368 }

369 }

370 }

371 }

372 }

373 }

374 }

375 }

376 }

377 }

378 }

379 }

380 }

381 }

382 }

383 }

384 }

385 }

386 }

387 }

388 }

389 }

390 }

391 }

392 }

393 }

394 }

395 }

396 }

397 }

398 }

399 }

400 }

401 }

402 }

403 }

404 }

405 }

406 }

407 }

408 }

409 }

410 }

411 }

412 }

413 }

414 }

415 }

416 }

417 }

418 }

419 }

420 }

421 }

422 }

423 }

424 }

425 }

426 }

427 }

428 }

429 }

430 }

431 }

432 }

433 }

434 }

435 }

436 }

437 }

438 }

439 }

440 }

441 }

442 }

443 }

444 }

445 }

446 }

447 }

448 }

449 }

450 }

451 }

452 }

453 }

454 }

455 }

456 }

457 }

458 }

459 }

460 }

461 }

462 }

463 }

464 }

465 }

466 }

467 }

468 }

469 }

470 }

471 }

472 }

473 }

474 }

475 }

476 }

477 }

478 }

479 }

480 }

481 }

482 }

483 }

484 }

485 }

486 }

487 }

488 }

489 }

490 }

491 }

492 }

493 }

494 }

495 }

496 }

497 }

498 }

499 }

500 }

501 }

502 }

503 }

504 }

505 }

506 }

507 }

508 }

509 }

510 }

511 }

512 }

513 }

514 }

515 }

516 }

517 }

518 }

519 }

520 }

521 }

522 }

523 }

524 }

525 }

526 }

527 }

528 }

529 }

530 }

531 }

532 }

533 }

534 }

535 }

536 }

537 }

538 }

539 }

540 }

541 }

542 }

543 }

544 }

545 }

546 }

547 }

548 }

549 }

550 }

551 }

552 }

553 }

554 }

555 }

556 }

557 }

558 }

559 }

560 }

561 }

562 }

563 }

564 }

565 }

566 }

567 }

568 }

569 }

570 }

571 }

572 }

573 }

574 }

575 }

576 }

577 }

578 }

579 }

580 }

581 }

582 }

583 }

584 }

585 }

586 }

587 }

588 }

589 }

590 }

591 }

592 }

593 }

594 }

595 }

596 }

597 }

598 }

599 }

600 }

601 }

602 }

603 }

604 }

605 }

606 }

607 }

608 }

609 }

610 }

611 }

612 }

613 }

614 }

615 }

616 }

617 }

618 }

619 }

620 }

621 }

622 }

623 }

624 }

625 }

626 }

627 }

628 }

629 }

630 }

631 }

632 }

633 }

634 }

635 }

636 }

637 }

638 }

639 }

640 }

641 }

642 }

643 }

644 }

645 }

646 }

647 }

648 }

649 }

650 }

651 }

652 }

653 }

654 }

655 }

656 }

657 }

658 }

659 }

660 }

661 }

662 }

663 }

664 }

665 }

666 }

667 }

668 }

669 }

670 }

671 }

672 }

673 }

674 }

675 }

676 }

677 }

678 }

679 }

680 }

681 }

682 }

683 }

684 }

685 }

686 }

687 }

688 }

689 }

690 }

691 }

692 }

693 }

694 }

695 }

696 }

697 }

698 }

699 }

700 }

701 }

702 }

703 }

704 }

705 }

706 }

707 }

708 }

709 }

710 }

711 }

712 }

713 }

714 }

715 }

716 }

717 }

718 }

719 }

720 }

721 }

722 }

723 }

724 }

725 }

726 }

727 }

728 }

729 }

730 }

731 }

732 }

733 }

734 }

735 }

736 }

737 }

738 }

739 }

740 }

741 }

742 }

743 }

744 }

745 }

746 }

747 }

748 }

749 }

750 }

751 }

752 }

753 }

754 }

755 }

756 }

757 }

758 }

759 }

760 }

761 }

762 }

763 }

764 }

765 }

766 }

767 }

768 }

769 }

770 }

771 }

772 }

773 }

774 }

775 }

776 }

777 }

778 }

779 }

780 }

781 }

782 }

783 }

784 }

785 }

786 }

787 }

788 }

789 }

790 }

791 }

792 }

793 }

794 }

795 }

796 }

797 }

798 }

799 }

800 }

801 }

802 }

803 }

804 }

805 }

806 }

807 }

808 }

809 }

810 }

811 }

812 }

813 }

814 }

815 }

816 }

817 }

818 }

819 }

820 }

821 }

822 }

823 }

824 }

825 }

826 }

827 }

828 }

829 }

830 }

831 }

832 }

833 }

834 }

835 }

836 }

837 }

838 }

839 }

840 }

841 }

842 }

843 }

844 }

845 }

846 }

847 }

848 }

849 }

850 }

851 }

852 }

853 }

854 }

855 }

856 }

857 }

858 }

859 }

860 }

861 }

862 }

863 }

864 }

865 }

866 }

867 }

868 }

869 }

870 }

871 }

872 }

873 }

874 }

875 }

876 }

877 }

878 }

879 }

880 }

881 }

882 }

883 }

884 }

885 }

886 }

887 }

888 }

889 }

890 }

891 }

892 }

893 }

894 }

895 }

896 }

897 }

898 }

899 }

900 }

901 }

902 }

903 }

904 }

905 }

906 }

907 }

908 }

909 }

910 }

911 }

912 }

913 }

914 }

915 }

916 }

917 }

918 }

919 }

920 }

921 }

922 }

923 }

924 }

925 }

926 }

927 }

928 }

929 }

930 }

931 }

932 }

933 }

934 }

935 }

936 }

937 }

938 }

939 }

940 }

941 }

942 }

943 }

944 }

945 }

946 }

947 }

948 }

949 }

950 }

951 }

952 }

953 }

954 }

955 }

956 }

957 }

958 }

959 }

960 }

961 }

962 }

963 }

964 }

965 }

966 }

967 }

968 }

969 }

970 }

971 }

972 }

973 }

974 }

975 }

976 }

977 }

978 }

979 }

980 }

981 }

982 }

983 }

984 }

985 }

986 }

987 }

988 }

989 }

990 }

991 }

992 }

993 }

994 }

995 }

996 }

997 }

998 }

999 }

1000 }

1001 }

1002 }

1003 }

1004 }

1005 }

1006 }

1007 }

1008 }

1009 }

1010 }

1011 }

1012 }

1013 }

1014 }

1015 }

1016 }

1017 }

1018 }

1019 }

1020 }

1021 }

1022 }

1023 }

1024 }

1025 }

1026 }

1027 }

1028 }

1029 }

1030 }

1031 }

1032 }

1033 }

1034 }

1035 }

1036 }

1037 }

1038 }

1039 }

1040 }

1041 }

1042 }

1043 }

1044 }

1045 }

1046 }

1047 }

1048 }

1049 }

1050 }

1051 }

1052 }

1053 }

1054 }

1055 }

1056 }

1057 }

1058 }

1059 }

1060 }

1061 }

1062 }

1063 }

1064 }

1065 }

1066 }

1067 }

1068 }

1069 }

1070 }

1071 }

1072 }

1073 }

1074 }

1075 }

1076 }

1077 }

1078 }

1079 }

1080 }

1081 }

1082 }

1083 }

1084 }

1085 }

1086 }

1087 }

1088 }

1089 }

1090 }

1091 }

1092 }

1093 }

1094 }

1095 }

1096 }

1097 }

1098 }

1099 }

1100 }

1101 }

1102 }

1103 }

1104 }

1105 }

1106 }

1107 }

1108 }

1109 }

1110 }

1111 }

1112 }

1113 }

1114 }

1115 }

1116 }

1117 }

1118 }

1119 }

1120 }

1121 }

1122 }

1123 }

1124 }

1125 }

1126 }

1127 }

1128 }

1129 }

1130 }

1131 }

1132 }

1133 }

1134 }

1135 }

1136 }

1137 }

1138 }

1139 }

1140 }

1141 }

1142 }

1143 }

1144 }

1145 }

1146 }

1147 }

1148 }

1149 }

1150 }

1151 }

1152 }

1153 }

1154 }

1155 }

1156 }

1157 }

1158 }

1159 }

1160 }

1161 }

1162 }

1163 }

1164 }

1165 }

1166 }

1167 }

1168 }

1169 }

1170 }

1171 }

1172 }

1173 }

1174 }

1175 }

1176 }

1177 }

1178 }

1179 }

1180 }

1181 }

1182 }

1183 }

1184 }

1185 }

1186 }

1187 }

1188 }

1189 }

1190 }

1191 }

1192 }

1193 }

1194 }

1195 }

1196 }

1197 }

1198 }

1199 }

1200 }

1201 }

1202 }

1203 }

1204 }

1205 }

1206 }

1207 }

1208 }

1209 }

1210 }

1211 }

1212 }

1213 }

1214 }

1215 }

1216 }

1217 }

1218 }

1219 }

1220 }

1221 }

1222 }

1223 }

1224 }

1225 }

1226 }

1227 }

1228 }

1229 }

1230 }

1231 }

1232 }

1233 }

1234 }

1235 }

1236 }

1237 }

1238 }

1239 }

1240 }

1241 }

1242 }

1243 }

1244 }

1245 }

1246 }

1247 }

1248 }

1249 }

1250 }

1251 }

1252 }

1253 }

1254 }

1255 }

1256 }

1257 }

1258 }

1259 }

1260 }

1261 }

1262 }

1263 }

1264 }

1265 }

1266 }

1267 }

1268 }

1269 }

1270 }

1271 }

1272 }

1273 }

1274 }

1275 }

1276 }

1277 }

1278 }

1279 }

1280 }

1281 }

1282 }

1283 }

1284 }

1285 }

1286 }

1287 }

1288 }

1289 }

1290 }

1291 }

1292 }

1293 }

1294 }

1295 }

1296 }

1297 }

1298 }

1299 }

1300 }

1301 }

1302 }

1303 }

1304 }

1305 }

1306 }

1307 }

1308 }

1309 }

1310 }

1311 }

1312 }

1313 }

1314 }

1315 }

1316 }

1317 }

1318 }

1319 }

1320 }

1321 }

1322 }

1323 }

1324 }

1325 }

1326 }

1327 }

1328 }

1329 }

1330 }

1331 }

1332 }

1333 }

1334 }

1335 }

1336 }

1337 }

1338 }

1339 }

1340 }

1341 }

1342 }

1343 }

1344 }

1345 }

1346 }

1347 }

1348 }

1349 }

1350 }

1351 }

1352 }

1353 }

1354 }

1355 }

1356 }

1357 }

1358 }

1359 }

1360 }

1361 }

1362 }

1363 }

1364 }

1365 }

1366 }

1367 }

1368 }

1369 }

1370 }

1371 }

1372 }

1373 }

1374 }

1375 }

1376 }

1377 }

1378 }

1379 }

1380 }

1381 }

1382 }

1383 }

1384 }

1385 }

1386 }

1387 }

1388 }

1389 }

1390 }

1391 }

1392 }

1393 }

1394 }

1395 }

1396 }

1397 }

1398 }

1399 }

1400 }

1401 }

1402 }

1403 }

1404 }

1405 }

1406 }

1407 }

1408 }

1409 }

1410 }

1411 }

1412 }

1413 }

1414 }

1415 }

1416 }

1417 }

1418 }

1419 }

1420 }

1421 }

1422 }

1423 }

1424 }

1425 }

1426 }

1427 }

1428 }

1429 }

1430 }

1431 }

1432 }

1433 }

1434 }

1435 }

1436 }

1437 }

1438 }

1439 }

1440 }

1441 }

1442 }

1443 }

1444 }

1445 }

1446 }

1447 }

1448 }

1449 }

1450 }

1451 }

1452 }

1453 }

1454 }

1455 }

1456 }

1457 }

1458 }

1459 }

1460 }

1461 }

1462 }

1463 }

1464 }

1465 }

1466 }

1467 }

1468 }

1469 }

1470 }

1471 }

1472 }

1473 }

1474 }

1475 }

1476 }

1477 }



Command Prompt

X + -

====>115  
s====>104  
h====>13  
====>10  
  
====>116  
t====>101  
e====>110  
n====>100  
d====>117  
u====>108  
l====>107  
k====>97  
a====>114  
r====>13  
====>10  
  
====>73  
I====>78  
N====>68  
D====>13  
====>10  
  
====>73  
I====>80  
P====>76  
L====>13  
====>10  
  
====>-1

Neha Dargajani



41

D:\100operations>

WTC  
CPU



Search

ENCL IN 22:25 02-07-2023

