A

Project Report

On

**“Find Folder Size”**

Prepared By:

Saloni S. CHAUDHARI(1741046)

Sayali S. CHAUDHARI(1741051)

Guided By:

**MS. SHRUTIKA MAHAJAN**



**GOVERNMENT COLLEGE OF ENGINEERING, JALGAON**

**DEPARTMENT OF COMPUTER ENGINEERING**

**CERTIFICATE**



This is to certify that the project report entitled, **“FIND FOLDER SIZE**”, which is being submitted here with for the result of the work completed by “**Sayali chaudhari and Saloni chaudhari**” under my supervision and guidance within the four walls of the institute and the same has not been submitted elsewhere for the award of any degree.

(Ms. Shrutika Mahajan) (Prof. D. V. Chaudhari)

Project Guide Head of Computer Department

**ACKNOWLEDGEMENT**

It is indeed a matter of great pleasure and privilege to be able to present this project report on **“FIND FOLDER SIZE**” under the valuable guidance of **SHRUTIKA MAHAJAN** mam for her valuable guidance, advice and constant aspiration to our work. Also here we would like to thank our honorable **Principal Dr. R.P. BORKAR** And our **HOD D.V.Chaudhari** who made all facilities and faculty of Technology available for our in the college premises. We are obliged! We are also thankful to all my teachers and principal of this institute for providing us constant support and facilities.

It has been great fun to work together with problem related with project financial support from our parents it’s gratefully acknowledged. We would thank our parents for their understanding and support during this year. Many thank to our group members for their patience, encouragement among us.

SALONI S. CHAUDHARI

SAYALI S. CHAUDHARI

**INDEX**

|  |  |  |
| --- | --- | --- |
| **Chapter**  **No** | **Title** | **Page no**. |
| 1 | Abstract | 1 |
| 2 | Introduction | 2 |
| 3 | Methodology | 3 |
| 4 | Source Code | 4 |
| 5 | Output | 7 |
| 6 | Conclusion | 9 |
| 7 | References | 10 |

**1.ABSTRACT**

As the name specifies “**FIND FOLDER SIZE**” is developed using SWING. A **FIND FOLDER SIZE** is a type of minute app. A **Find Folder Size** is a small project which can be used for to knowing the memory usage.

The easiest path to find the folder size ,this program is definitely used .

In this program ,we used the packages like “io.file”,“swing”.

1.

2.iNTROD**UCTION**

The project consist of almost all the aspects of java for instance Swing and File concept.

It implements various packages including swing and io.file and also number formats and provide dynamic way of formatting number in java.

We used the class like JOptionPane to provide standard dialogue boxes.

**2.**

**3 .METHODOLOGY**

We have project on “Find Folder Size”.First we do codding and taken output.That is shows one diaload box wich consist of two buttons “Ok” and “Cancle” and one space for accepting path of foalder whose size to be found.After setting path in this space we gives output on terminal which consist size of folder in kb,mb and gb. **3.**

|  |  |
| --- | --- |
|  |  |
|  |  |

**4.sOURCE CODE**

import javax.swing.\*;

import java.io.File;

import java.text.DecimalFormat;

import java.text.NumberFormat;

public class FindFolderSize {

int totalFolderCount=0;

int totalFileCount=0;

int totalFolder=0;

int totalFile=0;

public static void main(String args [])

{

/\*\*\* Define your folder here. This is the folder whose size

\* statistics you want to find out.\*/

//String folder = "D:/JAVA Content/Coding";

String folder=JOptionPane.showInputDialog(null,"Type input folder path");

try{

DecimalFormat fmt =new DecimalFormat("#.##");

FindFolderSize fg=new FindFolderSize();

/\*\*Calculating <span id="IL\_AD10" class="IL\_AD">the file</span> size. By default size in long

\* is returned.\*/ **4.**

long fileSizeByte=fg.getFileSize(new File(folder));

/\* Formatting the long value to calculate sizein

\* different units KB, MB and GB\*/

doublefileSizeKB=Double.valueOf(fmt.format(fileSizeByte /1024));

doublefileSizeMB=Double.valueOf(fmt.format(fileSizeByte /(1024\*1024)));

doublefileSizeGB=Double.valueOf(fmt.format(fileSizeByte /(1024\*1024\*1024)));

/\*\*Printing the statistics\*\*/

System.out.println("\n\n##############â€“Folder Statisticsâ€“#################");

System.out.println("Total Folder Size: ["+fileSizeByte+" Bytes] \n\t\t["+fileSizeKB+" KB] \n\t\t["+fileSizeMB+" MB] \n\t\t["+fileSizeGB+" GB]");

System.out.println("Total Number of Folders: "+fg.getTotalFolderCount());

System.out.println("Total Number of Files: "+fg.getTotalFileCount());

System.out.println("##########â€“End Of Folder Statisticsâ€“##############");

}catch (Exception e)

{

System.out.println("Exception Occurred:\"+e.getMessage());

}

}

/\*\*\* @return the totalFolderCount\*/

public int getTotalFolderCount() { **5.**

return totalFolderCount;

}

/\*\*

\* @return the totalFileCount

\*/

public int getTotalFileCount() {

return totalFileCount;

}

public long getFileSize(File folder)

{

totalFolder++;

System.out.println("Folder: " + folder.getName());

long foldersize = 0;

File[] filelist = folder.listFiles();

for (int i = 0; i < filelist.length; i++) {

if (filelist[i].isDirectory()) {

foldersize += getFileSize(filelist[i]);

} else {

totalFile++;

foldersize += filelist[i].length();

}

}

return foldersize;

}

public int getTotalFolder() {

return totalFolder;

}

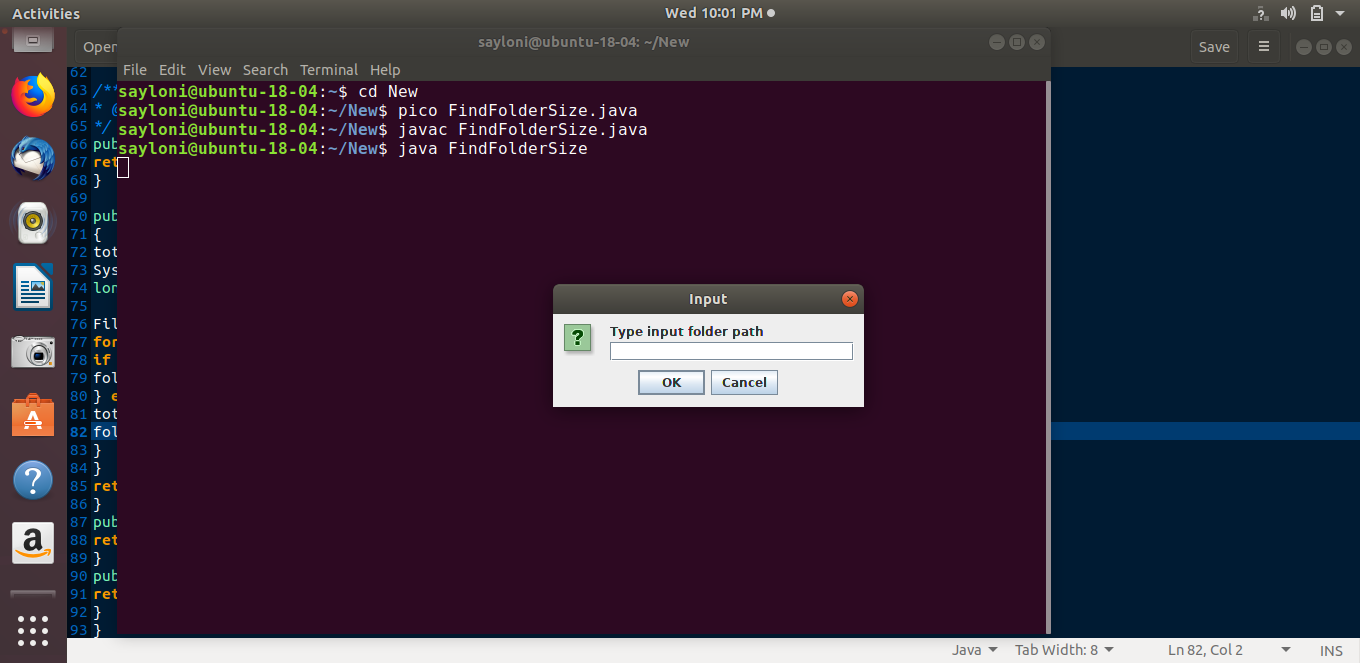
public int getTotalFile() {

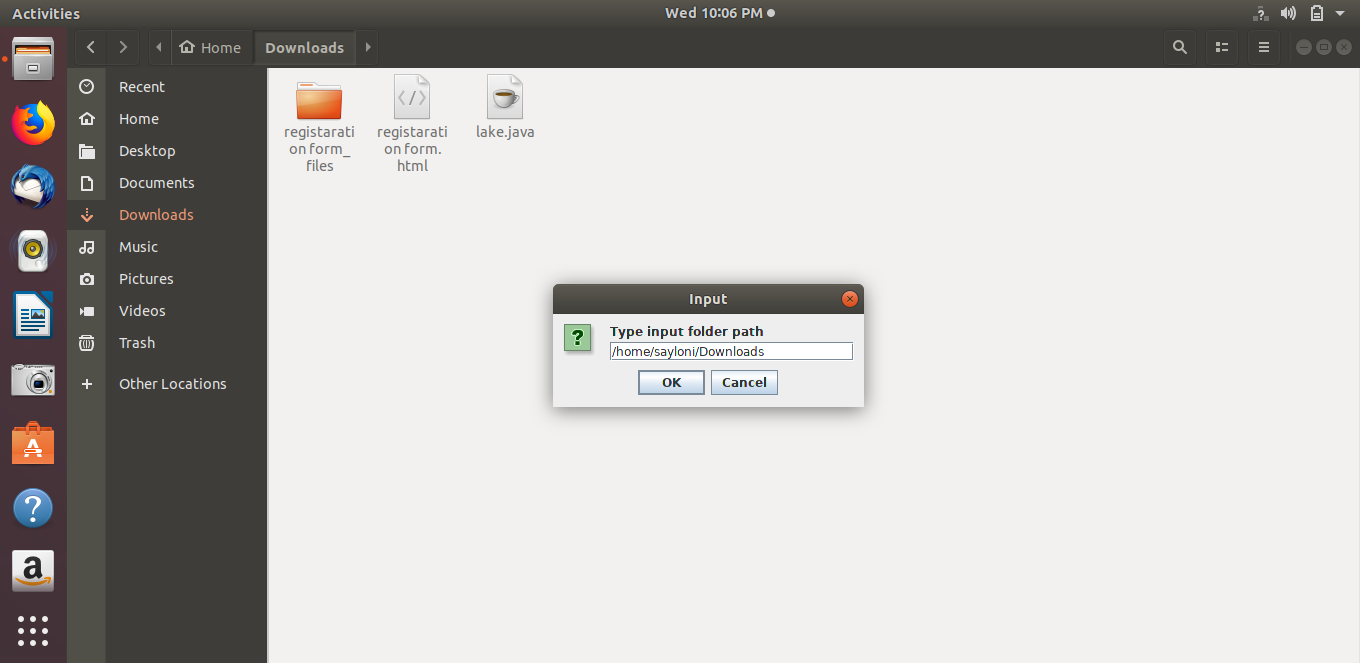
return totalFile;

}

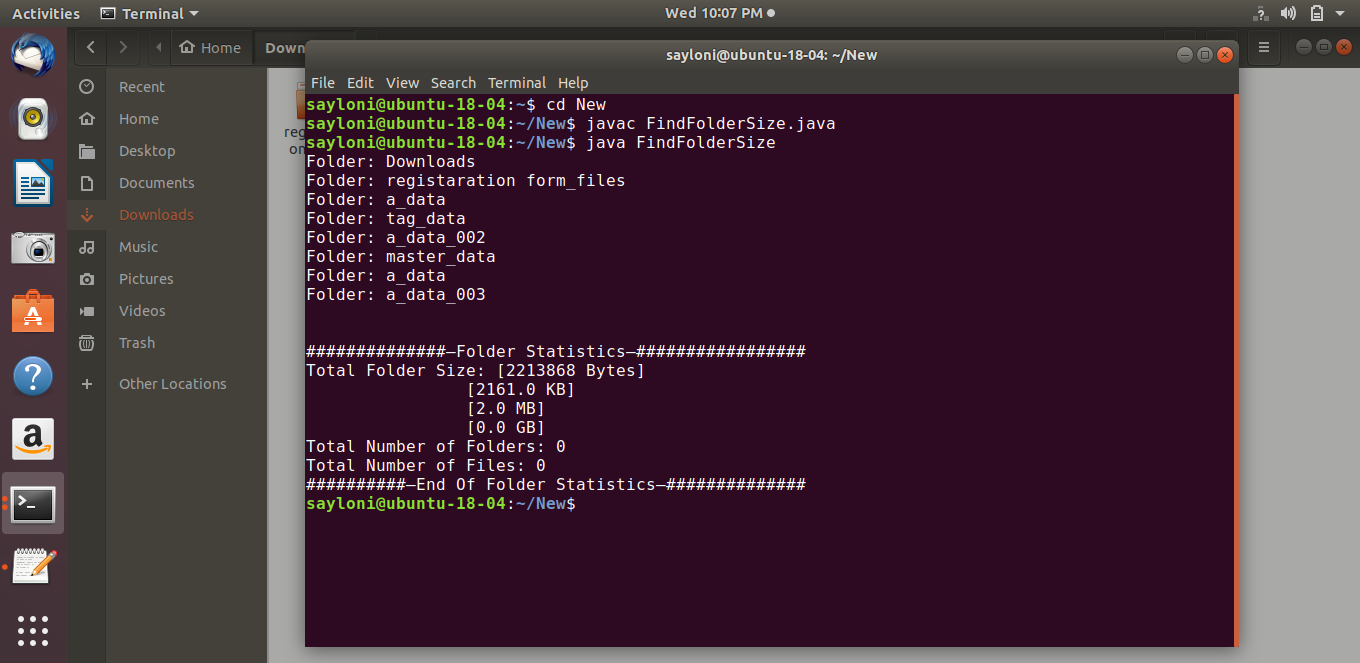
} **6.**

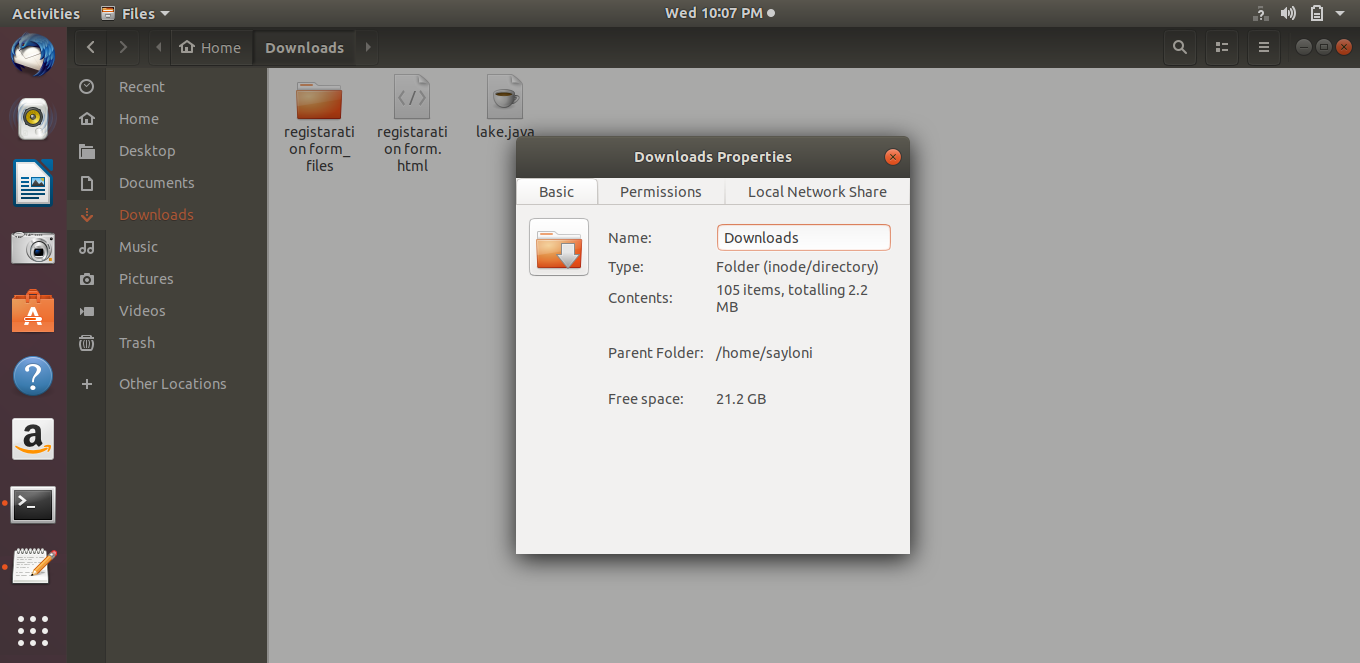
**5.OUTPUT**

****

****

**7.**

** OUTPUT SCREENSHOT**

****

**8.**

**6. CONCLUSION**

Thus, we developed the FIND FOLDER SIZE using the “SWING” and “JOptionPane Class”. Our FIND FOLDER SIZE is project which can be used for to knowing the memory usage. I surely says that it is useful to all the for folder

**9.**

7.REFERENCES

**Beginner:**

* Head First Java, 2nd Edition
* Thinking in Java (4th Edition)
* Think Java
* Introduction to Java by Sedgewick
* Java in a Nutshell
* Core Java Volume I--Fundamentals (9th Edition) (Core Series): Cay S. Horstmann
* Java How To Program (late objects) by Paul Deitel, Harvey Deitel

**Intermediate:**

* Effective Java (2nd Edition): Joshua Bloch
* Java Performance: Charlie Hunt, Binu John
* Head First Servlets and JSP
* SCJP by Kathy and Sierra
* Java - The Complete Reference by Herbert Schildt.
* Java Concurrency in Practice
* Java Performance
* The Java Programming Language, 4th Edition

**Advanced:**

* Java Puzzlers : Traps, Pitfalls, And Corner Cases

**10.**