1 איך יוצרים תיקייה בגאווה?

ב-Java, כדי ליצור תיקייה ב-Java תחילה אנו משתמשים ב-class file.(import java.io.File)

שנית, נקרא לתיקייה שברצוננו ליצור ונציין את הנתיב שבו נרצה ליצור את התיקייה עם שם המשתנה cm מסוג File.

שלישית, אנו משתמשים ב mkdir() כדי ליצור תיקייה חדשה ומשתמשים בתנאי if כדי לוודא שהתיקייה נוצרה.

<https://how.dev/answers/how-to-create-a-directory-using-java>

import java.io.File;

public class CmdEx {

public static void main(String[] args){

File cmd = new File ("C:\\Users\\Admin\\Downloads\\myNewFolder");

//تطبعلك إذا تم انشاء الملف او لا (الخطوة مش ضرورية ولكن للتاكيد)

if (cm.mkdir() == true) {

System.out.println("Directory has been created successfully");

}

else {

System.out.println("Directory cannot be created");

}

}

}

2 איך יוצרים קובץ?

ב־Java, ניתן ליצור קובץ חדש באמצעות המחלקה File והשיטה createNewFile(). פונקציה זו יוצרת קובץ חדש וריק. הפונקציה מחזירה true אם נתיב הקובץ המופשט אינו קיים ונוצר קובץ חדש. היא מחזירה false אם שם הקובץ כבר קיים.

<https://www.geeksforgeeks.org/file-createnewfile-method-in-java-with-examples/>

// createNewFile() method of File Class

import java.io.\*;

public class CmdEx {

public static void main(String args[])

{

try {

// Get the file

File f = new File("C:\\Users\\Admin\\Downloads\\myNewFolder\\Text");

// Create new file

// if it does not exist

if (f.createNewFile())

System.out.println("File created");

else

System.out.println("File already exists");

}

catch (Exception e) {

System.err.println(e);

}

}

}

3 איך בודקים אם קובץ/תיקייה קיימים ?

ב־Java, ניתן לבדוק אם קובץ או תיקייה קיימים באמצעות המחלקה File והשיטה exists().פונקציה זו קובעת האם קובץ או ספרייה המסומנים בשם הקובץ המופשט קיימים או לא. הפונקציה מחזירה אמת אם נתיב הקובץ המופשט קיים, אחרת מחזירה שקר.

<https://www.geeksforgeeks.org/file-exists-method-in-java-with-examples/>

// Java Program to Illustrate exists() method of File Class

// Importing input output classes

import java.io.\*;

public class CmdEx {

// Main driver method

public static void main(String args[])

{

// Getting the file by creating object of File class

File f = new File("C:\\Users\\Admin\\Downloads\\myNewFolder\\Text");

// Checking if the specified file exists or not

if (f.exists())

// Show if the file exists

System.out.println("Exists");

else

// Show if the file does not exists

System.out.println("Does not Exists");

}

}

4 איך מציגים את תוכן תיקייה?

רשימת כל הקבצים תיקייה ב-Java. במחלקה java.io.File המספקת מתודה מובנית listFiles() שתחזיר מערך של קבצים ותיקיות של הקבצים תיקייה.

<https://www.geeksforgeeks.org/how-to-list-all-files-in-a-directory-in-java/>

// Java program list all files in a directory using listFiles()

import java.io.File;

public class CmdEx {

public static void main(String[] args)

{

// Path of the specific directory

String directoryPath = "C:\\Users\\Admin\\Downloads\\myNewFolder";

// Using File class create an object for specific directory

File directory = new File(directoryPath);

// Using listFiles method we get all the files of a directory

// return type of listFiles is array

File[] files = directory.listFiles();

// Print name of the all files present in that path

if (files != null) {

for (File file : files) {

System.out.println(file.getName());

}

}

}

}

5 )רשות ) מה ההבדל בין File ל Path ?

1. Path הוא חלק מחבילת Java NIO (New Input/Output) שהוצגה ב-Java 7, המספקת פונקציונליות משופרת לטיפול בקבצים.

2. File הוא המחלקה הישנה יותר מ-Input/Output API המקורי ב-Java.

3. Path מספק שיטות טובות יותר לטיפול בנתיבי קבצים, כגון שילוב שני נתיבים או אחזור חלקים מנתיב.

4. שיטות File אינן תומכות בקישורים סמליים כמו שיטות Path.

5. ניתן להשתמש ב-Path עם מחלקות NIO אחרות כגון Files ו-FileSystems כדי לבצע פעולות קבצים מתקדמות.

<https://www.javaguides.net/2023/11/difference-between-path-and-file-in-java-nio.html>

import java.nio.file.Path;

import java.nio.file.Paths;

import java.io.File;

public class CmdEx {

public static void main(String[] args) {

// Using Path

Path path = Paths.get("C:\\Users\\Admin\\Downloads\\myNewFolder");

System.out.println("Path: " + path.toString());

// Using File

File file = new File("C:\\Users\\Admin\\Downloads\\myNewFolder\\Text");

System.out.println("File: " + file.getAbsolutePath());

}

}