**תרגיל minishell**

**שם הסטודנית: נור צאלח**

**ת"ז: 328124367**

**חלק 1:**

שאלה 1**:**

הקוד יוצר תיקייה באמצעות הפקודה Files.createDirectories(path).  
אם חלק מהתיקיות לא קיימות, הן ייווצרו באופן אוטומטי.  
אם התיקיות כבר קיימות, לא תתרחש שגיאה.

הקוד: import java.io.IOException ;

import java.nio.file.Files;

import java.nio.file.Path;

import java.nio.file.Paths;

public class DirectoryCreate {

public static void main(String[] args) {

String dir = "/home/mkyong/test2/test3/test4/";

try {

Path path = Paths.get(dir);

Files.createDirectories(path);

System.out.println("Directory is created!");

} catch (IOException e) {

System.err.println("Failed to create directory! " + e.getMessage());

}

}

}

קישור: <https://mkyong.com/java/how-to-create-directory-in-java/>

**2-** יוצרים קובץ בשם newFile.txt בתיקייה הנוכחית באמצעות Files.createFile().  
אם הקובץ כבר קיים או שיש שגיאה, מודפסת הודעת שגיאה למסך.  
הקוד משתמש במחלקות מ־java.nio.file כדי לטפל בקבצים.

הקוד: import java.io.IOException;

import java.nio.file.Files;

import java.nio.file.Path;

import java.nio.file.Paths;

public class DirectoryCreate {

public static void main(String[] args) {

String dir = "/home/mkyong/test2/test3/test4/";

try {

Path path = Paths.get(dir);

Files.createDirectories(path);

System.out.println("Directory is created!");

} catch (IOException e) {

System.err.println("Failed to create directory! " + e.getMessage());

}

}

}

קישור:<https://mkyong.com/java/how-to-create-directory-in-java/>

3-

בודים על ידי הקוד שבודק אם קובץ או תיקייה קיימים בנתיב שניתן (/home/mkyong/test/test.log).  
אם הקובץ קיים, הוא בודק אם זה קובץ רגיל או תיקייה, ומדפיס הודעה מתאימה.  
אם הנתיב לא קיים בכלל, הוא מדפיס שהקובץ לא קיים

הקוד:

package com.mkyong.io.file;

import java.nio.file.Files;  
import java.nio.file.Path;  
import java.nio.file.Paths;

public class FileExist {

public static void main(String[] args) {

Path path = Paths.get("/home/mkyong/test/test.log");

// check exists for file and directory

if (Files.exists(path)) {

if (Files.isRegularFile(path)) {

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

}

if (Files.isDirectory(path)) {

System.out.println("File exists, but it is a directory.");

}

} else {

System.out.println("File doesn't exist");

}

}

}

<https://mkyong.com/java/how-to-check-if-a-file-exists-in-java/> הקישור:

בכדי להציג את תוכן תיקייה בשפת Java, ניתן להשתמש במחלקה File, 4-שמייצגת קובץ או תיקייה במערכת הקבצים. בעזרת המתודה listFiles() אפשר לקבל את כל הקבצים והתיקיות שבתוכה, ולבצע עליהם פעולות כמו בדיקה אם זה קובץ או תיקייה והצגת הנתיב המלא. כך ניתן לדמות פקודת ls או dir בממשק שורת פקודה פשוט

הקוד

**package** test;

**import** java.io.File;

**import** java.io.IOException;

**public** **class** DirectoryContents {

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

File f = **new** File(**"."**); // current directory

File[] files = f.listFiles();

**for** (File file : files) {

**if** (file.isDirectory()) {

System.out.print(**"directory:"**);

} **else** {

System.out.print(**" file:"**);

}

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

}

}

}

.

File היא מחלקה ישנה בג'אווה שמאפשרת גישה לקבצים ותיקיות אך עם יכולות מוגבלות. -5  
Path היא מחלקה חדשה יותר שנותנת כלים מתקדמים, מדויקים וגמישים לעבודה עם מערכת הקבצים.

Path עובדת עם מחלקות נוספות שמאפשרות פעולות מורכבות יותר על קבצים ותיקיות.

<https://docs.oracle.com/javase/8/docs/api/java/io/File.html> הקישור: