**Q1. Given a path as input, print the count of no. of files in that path location. (You can extend your solution by counting directories, sub-directories and printing the count of each)**

**Ans -**

import java.io.File;

import java.util.Scanner;

public class Main {

public static int dir\_count = 0, sub\_dir\_count = 0;

public static int getFilesCount(File file, boolean in\_dir) {

File[] files = file.listFiles();

int count = 0;

for (File f : files)

if (f.isDirectory()) {

if(in\_dir)

sub\_dir\_count++; //if in\_dir is true then we are inside a directory hence sub\_dir\_count is increased.

else

dir\_count++;

count += getFilesCount(f,true);

}

else

count++;

return count;

}

public static void main(String[] args) {

String s;

System.out.println("Enter the path here");

Scanner sc = new Scanner(System.in);

s = sc.nextLine();

File f = new File(s); //enter path here

System.out.println("No. of files in path : " + getFilesCount(f,false));

System.out.println("No. of directories in path : " + dir\_count);

System.out.println("No. of sub-directories in path : " + sub\_dir\_count);

dir\_count = 0;

sub\_dir\_count = 0;

}

}

**Q2. Create a student table in MySQL DB from Java with the below mentioned columns in the student table. Read the xml document from input.txt (sample input.txt as provided below) and insert the students list from the xml document to the student table in MySQL.**

**Student Table Columns:**

**id: Number (Roll number from xml document) (primary key)**

**Name: String (firstname + middlename from xml document)**

**Surname: String (lastnaem from xml document)**

**Gender: string (gender from xml document)**

**Marks: Float (marks from xml document)**

**Ans –**

import org.w3c.dom.Document;

import org.w3c.dom.Element;

import org.w3c.dom.Node;

import org.w3c.dom.NodeList;

import javax.xml.parsers.DocumentBuilder;

import javax.xml.parsers.DocumentBuilderFactory;

import javax.xml.xpath.XPath;

import javax.xml.xpath.XPathConstants;

import javax.xml.xpath.XPathFactory;

import java.io.File;

import java.sql.\*;

import java.util.Arrays;

import java.util.List;

public class Student {

static final String JDBC\_DRIVER = "com.mysql.jdbc.Driver";

static final String DB\_URL = "jdbc:mysql://localhost:3306/Student\_Assignment";

static final String USER = "root";

static final String PASS = "MyNewPass";

public static void main(String[] args) {

Connection conn = null;

Statement statement = null;

try{

Class.forName("com.mysql.jdbc.Driver");

conn = DriverManager.getConnection(DB\_URL, USER, PASS);

System.out.println("Connected database successfully!!!");

statement = conn.createStatement();

String sql = "CREATE TABLE STUDENT " +

"(id INTEGER not NULL, " +

" Name VARCHAR(255), " +

" Surname VARCHAR(255), " +

" Gender VARCHAR(255), " +

" Marks FLOAT, " +

" PRIMARY KEY ( id ))";

statement.executeUpdate(sql);

System.out.println("Table creation Successful");

File file = new File("C:\\Users\\ankirai\\Desktop\\Student.xml");

DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();

DocumentBuilder builder = factory.newDocumentBuilder();

Document studentXmlDoc = builder.parse(file);

Element studentsRoot = studentXmlDoc.getDocumentElement();

NodeList students = studentsRoot.getElementsByTagName("student");

for(int i=0;i<students.getLength();i++) {

Element thisStudent = (Element) students.item(i);

String rollNo = thisStudent.getAttribute("rollno");

NodeList name = thisStudent.getElementsByTagName("name");

String studentName = null;

String surname = null;

for (int j = 0; j < name.getLength(); j++) {

Element thisName = (Element) name.item(j);

studentName = thisName.getElementsByTagName("firstname").item(0).getFirstChild().getNodeValue() + " " + thisName.getElementsByTagName("middlename").item(0).getFirstChild().getNodeValue();

surname = thisName.getElementsByTagName("lastname").item(0).getFirstChild().getNodeValue();

}

String gender = thisStudent.getElementsByTagName("gender").item(0).getFirstChild().getNodeValue();

String marks = thisStudent.getElementsByTagName("marks").item(0).getFirstChild().getNodeValue();

String sqlInsert = "INSERT INTO Student" + "(id, Name, Surname, Gender, Marks)" +

"VALUES(" + rollNo + ", '" + studentName + "', '" + surname + "', '" + gender + "'," + marks + ")";

statement.executeUpdate(sqlInsert);

}

}catch(SQLException se){

se.printStackTrace();

}catch(Exception e){

e.printStackTrace();

}finally{

try{

if(statement!=null)

conn.close();

}catch(SQLException se){

}// do nothing

try{

if(conn!=null)

conn.close();

}catch(SQLException se){

se.printStackTrace();

}

}

}

}



