import java.util.Arrays;

import java.util.Comparator;

public class EmployeeWorkHours {

public static void main(String[] args) {

// Example data: 8 employees, each with 7 days of work hours

double[][] workHours = {

{8, 9, 8, 7, 8, 6, 5},

{7, 8, 9, 8, 7, 8, 6},

{9, 8, 7, 8, 9, 7, 8},

{6, 7, 8, 9, 8, 7, 6},

{8, 8, 8, 8, 8, 8, 8},

{7, 6, 7, 6, 7, 6, 7},

{9, 9, 9, 9, 9, 9, 9},

{5, 5, 5, 5, 5, 5, 5}

};

// Array to store total hours for each employee

double[] totalHours = new double[workHours.length];

// Calculate total hours for each employee

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

double sum = 0;

for (int j = 0; j < workHours[i].length; j++) {

sum += workHours[i][j];

}

totalHours[i] = sum;

}

// Create an array of employee indexes

Integer[] employeeIndexes = new Integer[workHours.length];

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

employeeIndexes[i] = i;

}

// Sort employee indexes based on total hours in decreasing order

Arrays.sort(employeeIndexes, new Comparator<Integer>() {

@Override

public int compare(Integer o1, Integer o2) {

return Double.compare(totalHours[o2], totalHours[o1]);

}

});

// Display employees and their total hours in decreasing order

System.out.println("Employee\tTotal Hours");

for (int index : employeeIndexes) {

System.out.printf("Employee %d\t%.2f\n", index + 1, totalHours[index]);

}

}

}

import java.util.Scanner;

public class MatrixColumnSum {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

double[][] matrix = new double[3][4];

System.out.println("Enter a 3-by-4 matrix row by row:");

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 4; j++) {

matrix[i][j] = scanner.nextDouble();

}

}

for (int j = 0; j < 4; j++) {

double sum = 0;

for (int i = 0; i < 3; i++) {

sum += matrix[i][j];

}

System.out.printf("Sum of the elements at column %d is %.1f%n", j, sum);

}

scanner.close();

}

}

import java.util.Random;

public class MatrixFiller {

public static void main(String[] args) {

int size = 4;

int[][] matrix = new int[size][size];

Random random = new Random();

// Fill the matrix with random 0s and 1s

for (int i = 0; i < size; i++) {

for (int j = 0; j < size; j++) {

matrix[i][j] = random.nextInt(2); // Generates either 0 or 1

}

}

// Print the matrix

System.out.println("Generated Matrix:");

for (int i = 0; i < size; i++) {

for (int j = 0; j < size; j++) {

System.out.print(matrix[i][j] + " ");

}

System.out.println();

}

// Find the first row with the most 1s

int maxRowOnes = -1;

int rowWithMostOnes = -1;

for (int i = 0; i < size; i++) {

int currentRowOnes = 0;

for (int j = 0; j < size; j++) {

if (matrix[i][j] == 1) {

currentRowOnes++;

}

}

if (currentRowOnes > maxRowOnes) {

maxRowOnes = currentRowOnes;

rowWithMostOnes = i;

}

}

// Find the first column with the most 1s

int maxColOnes = -1;

int colWithMostOnes = -1;

for (int j = 0; j < size; j++) {

int currentColOnes = 0;

for (int i = 0; i < size; i++) {

if (matrix[i][j] == 1) {

currentColOnes++;

}

}

if (currentColOnes > maxColOnes) {

maxColOnes = currentColOnes;

colWithMostOnes = j;

}

}

// Print the results

System.out.println("First row with the most 1s: Row " + rowWithMostOnes + " with " + maxRowOnes + " ones.");

System.out.println("First column with the most 1s: Column " + colWithMostOnes + " with " + maxColOnes + " ones.");

}

}

import java.util.Arrays;

import java.util.Collections;

import java.util.List;

import java.util.ArrayList;

public class ArrayShuffler {

public static void shuffle(int[][] m) {

// Convert the 2D array into a list of rows

List<int[]> rowList = new ArrayList<>();

for (int[] row : m) {

rowList.add(row);

}

// Shuffle the list of rows

Collections.shuffle(rowList);

// Copy the shuffled rows back into the original 2D array

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

m[i] = rowList.get(i);

}

}

public static void main(String[] args) {

int[][] matrix = {

{1, 2, 3},

{4, 5, 6},

{7, 8, 9},

{10, 11, 12}

};

System.out.println("Original matrix:");

printMatrix(matrix);

shuffle(matrix);

System.out.println("Shuffled matrix:");

printMatrix(matrix);

}

private static void printMatrix(int[][] matrix) {

for (int[] row : matrix) {

System.out.println(Arrays.toString(row));

}

}

}

import java.util.Arrays;

import java.util.Collections;

import java.util.List;

import java.util.ArrayList;

public class MatrixShuffler {

public static void shuffle(int[][] m) {

// Convert the 2D array into a list of rows

List<int[]> rowList = new ArrayList<>();

for (int[] row : m) {

rowList.add(row);

}

// Shuffle the list of rows

Collections.shuffle(rowList);

// Copy the shuffled rows back into the original 2D array

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

m[i] = rowList.get(i);

}

}

public static void main(String[] args) {

int[][] m = {{1, 2}, {3, 4}, {5, 6}, {7, 8}, {9, 10}};

System.out.println("Original matrix:");

printMatrix(m);

shuffle(m);

System.out.println("Shuffled matrix:");

printMatrix(m);

}

private static void printMatrix(int[][] matrix) {

for (int[] row : matrix) {

System.out.println(Arrays.toString(row));

}

}

}