**Assignment # 4 - Files and two-dim arrays**

**Sai Kiran Reddy Gokula**

**Output:**

**A screenshot of a computer

Description automatically generated**

**Console Output:**

Number of lines in the file =350

Longest row in the file is: 20 ,with length of: 30

Max value in first row = 98

Max value in file = 100

**Source Code:**

package a4;

import java.util.stream.Stream;

import java.io.File;

import java.io.IOException;

import java.nio.file.Files;

import java.nio.file.Paths;

/\*\*

\* CPSC 24500-001- Object-Oriented Programming

\* Assignment 4: Files and two-dim arrays

\*/

public class Assignment4 {

/\*\*

\* @param filename

\* @return number of lines in a text file

\* @throws Exception

\*/

private static int getNoLines(String filename) throws Exception{

try (Stream<String> fileStream = Files.lines(Paths.get(filename))) {

return (int) fileStream.count();

}

}

/\*\*

\*

\* @param filename source file

\* @return two dim array (jagged array), where each row in the array contains the values in one line of the file,

\* the length of each row depends on the number of values in each line of the file.

\* @throws Exception

\*/

public static int[][] create2DArray(String filename) throws Exception {

//String[] lines = Files.readAllLines(Paths.get(filename)).toArray(new String[0]);

String[] lines = Files.lines(Paths.get(filename)).toArray(String[]::new);

// Initialize the 2D array

int[][] array = new int[lines.length][];

// Populate the array

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

String[] values = lines[i].split("\\s+");

array[i] = new int[values.length];

for (int j = 0; j < values.length; j++) {

array[i][j] = Integer.parseInt(values[j]);

}

}

return array;

}

/\*\*

\* To fetch the index of the longest row in the given two dim array.

\* @param two dim array

\* @return index of the longest row

\* @throws exception if the given array is null or empty & Array out of bound exception

\*/

public static int findLongestRow(int[][] array) throws Exception {

if (array == null || array.length == 0) {

throw new Exception("Array is null or empty");

}

int longestRow = 0;

int maxLength = array[0].length;

try {

for (int i = 1; i < array.length; i++) {

if (array[i].length > maxLength) {

longestRow = i;

maxLength = array[i].length;

}

}

}

catch (ArrayIndexOutOfBoundsException e) {

// Handle the exception if it occurs

System.out.println("Array index out of bounds exception occurred."+e);

}

return longestRow;

}

/\*\*

\* To get the maximum value in the given row of the two dim array.

\* @param two dim array (jagged array)

\* @param rowIndex value of the row

\* @return maximum value in the row

\* @throws IllegalArgumentException if the array is null or empty or if the row index is out of bound

\*/

private static int findMaxInRow(int[][] array, int rowIndex) throws IllegalArgumentException {

if (array == null || array.length == 0) {

throw new IllegalArgumentException("Array is null or empty");

}

if (rowIndex < 0 || rowIndex >= array.length) {

throw new IllegalArgumentException("Row index is out of bounds");

}

int[] row = array[rowIndex];

if (row.length == 0) {

throw new IllegalArgumentException("Row is empty");

}

int max = Integer.MIN\_VALUE;

for (int value : array[rowIndex]) {

if (value > max) {

max = value;

}

}

return max;

}

/\*\*

\* Finds the maximum value in the entire two dim array.

\* @param two dim array

\* @return maximum value in the array

\* @throws IllegalArgumentException if the array is null or empty

\*/

public static int findMax(int[][] array) throws IllegalArgumentException {

if (array == null || array.length == 0) {

throw new IllegalArgumentException("Array is null or empty");

}

int result = array[0][0]; // Initializing result to the first element of the array

// Iterates each row and column to find the maximum value

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

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

if (array[i][j] > result) {

result = array[i][j];

}

}

}

return result;

}

public static void main(String[] args) {

String filename = null;

if (args.length <1) {

System.out.println("usage: Assignment4 inputFilename ");

System.exit(0);

}

filename = args[0];

int arr[][] = null;

try {

System.out.println("Number of lines in the file ="+ getNoLines(filename));

arr = create2DArray(filename);

int longestRow = findLongestRow(arr);

System.out.println("Longest row in the file is: "+ (longestRow+1 )+" ,with length of: "

+arr[longestRow].length);

System.out.println("Max value in first row = "+ findMaxInRow(arr, 0));

System.out.println("Max value in file = "+ findMax(arr));

} catch (Exception e) {

System.out.print(e);

}

}

}