Joel Atkinson

September 1, 2025

Assignment 5.2

CSD402 Java for Programmers

Code:

//Joel Atkinson, September 1, 2025, Assignment 5.2 CSD402 Java for Programmers  
/\*The purpose of this assignment is to write 4 methods to find the location of the largest and smallest values in  
int and double array's.\*/  
  
public class ArrayLocator {  
 // Method to find the location of the largest value in a 2D double array  
 public static int[] locateLargest(double[][] arrayParam) {  
 // Check to see if array is null or empty  
 if (arrayParam == null || arrayParam.length == 0 || arrayParam[0].length == 0) {  
 return new int[]{-1, -1};  
 }  
  
 // Initialize max value with first element and set initial position  
 double max = arrayParam[0][0];  
 int rowIndex = 0;  
 int colIndex = 0;  
  
 // Loop through each row and column of the array and compare current element with min, update if larger  
 for (int i = 0; i < arrayParam.length; i++) {  
 for (int j = 0; j < arrayParam[i].length; j++) {  
 if (arrayParam[i][j] > max) {  
 max = arrayParam[i][j];  
 rowIndex = i;  
 colIndex = j;  
 }  
 }  
 }  
  
 // Return the [row,column] position of the largest value  
 return new int[]{rowIndex, colIndex};  
 }  
  
 // Method to find the location of the largest value in a 2D int array  
 public static int[] locateLargest(int[][] arrayParam) {  
 // Check if array is empty of null  
 if (arrayParam == null || arrayParam.length == 0 || arrayParam[0].length == 0) {  
 return new int[]{-1, -1};  
 }  
  
 // Initialize max value with the first element and initial position  
 int max = arrayParam[0][0];  
 int rowIndex = 0;  
 int colIndex = 0;  
  
 // Loop through each row and column of the array and compare current element with min, update if larger  
 for (int i = 0; i < arrayParam.length; i++) {  
 for (int j = 0; j < arrayParam[i].length; j++) {  
 if (arrayParam[i][j] > max) {  
 max = arrayParam[i][j];  
 rowIndex = i;  
 colIndex = j;  
 }  
 }  
 }  
  
 // Return the [row,column] position of the largest value  
 return new int[]{rowIndex, colIndex};  
 }  
  
 // Method to find the location of the smallest value in a 2D double array  
 public static int[] locateSmallest(double[][] arrayParam) {  
 // Check if array is null or empty  
 if (arrayParam == null || arrayParam.length == 0 || arrayParam[0].length == 0) {  
 return new int[]{-1, -1};  
 }  
  
 // Initialize min value with first element and set initial position  
 double min = arrayParam[0][0];  
 int rowIndex = 0;  
 int colIndex = 0;  
  
 // Loop through each row and column of the array and compare current element with min, update if smaller  
 for (int i = 0; i < arrayParam.length; i++) {  
 for (int j = 0; j < arrayParam[i].length; j++) {  
 if (arrayParam[i][j] < min) {  
 min = arrayParam[i][j];  
 rowIndex = i;  
 colIndex = j;  
 }  
 }  
 }  
  
 // Return the location [row,column] of the smallest value in the array  
 return new int[]{rowIndex, colIndex};  
 }  
  
 // Method to find the location of the smallest value in a 2D int array  
 public static int[] locateSmallest(int[][] arrayParam) {  
 // Check to see if array is null or empty  
 if (arrayParam == null || arrayParam.length == 0 || arrayParam[0].length == 0) {  
 return new int[]{-1, -1};  
 }  
  
 // Initialize the min value with the first element and set initial position  
 int min = arrayParam[0][0];  
 int rowIndex = 0;  
 int colIndex = 0;  
  
 // Loop trough each row and column of the array and compare current element with min, update if smaller  
 for (int i = 0; i < arrayParam.length; i++) {  
 for (int j = 0; j < arrayParam[i].length; j++) {  
 if (arrayParam[i][j] < min) {  
 min = arrayParam[i][j];  
 rowIndex = i;  
 colIndex = j;  
 }  
 }  
 }  
  
 // Return location [row,column] of smallest value  
 return new int[]{rowIndex, colIndex};  
 }  
  
 // Main method to test the locateLargest and locateSmallest methods  
 public static void main(String[] args) {  
 // Define the 2D array with double values for test  
 double[][] doubleArray = {  
 {7.6, 6.4, 9.5},  
 {5.8, 1.1, 4.2},  
 {3.7, 5.4, 6.1 }  
 };  
 // Find and store largest and smallest positions in double array  
 int[] largestDouble = *locateLargest*(doubleArray);  
 int[] smallestDouble = *locateSmallest*(doubleArray);  
 // Print the positions of largest and smallest in double array  
 System.*out*.println("Largest in double array at: [" + largestDouble[0] + ", " + largestDouble[1] + "]");  
 System.*out*.println("Smallest in double array at: [" + smallestDouble[0] + ", " + smallestDouble[1] + "]");  
  
 // Define the 2D array with int values for test  
 int[][] intArray = {  
 {45, 12, 31},  
 {95, 62, 45},  
 {57, 87, 19}  
 };  
 // Find and store the largest and smallest positions in int array  
 int[] largestInt = *locateLargest*(intArray);  
 int[] smallestInt = *locateSmallest*(intArray);  
 // Print the locations of the largest and smallest in int array  
 System.*out*.println("Largest in int array at: [" + largestInt[0] + ", " + largestInt[1] + "]");  
 System.*out*.println("Smallest in int array at: [" + smallestInt[0] + ", " + smallestInt[1] + "]");  
 }  
}

Screenshot of program running without errors:

A screenshot of a computer program

AI-generated content may be incorrect.