Частное учреждение образования

«Колледж бизнеса и права»

Тема: «Тестирование кода на языке Java.»

Цель: Научиться тестировать код на языке Java при помощи библиотеки

JUnit.

Учащийся М.Н. Гончаров.

группа: Т-992

2022

Вариант 6

Дана квадратная матрица. Найти сумму элементов, которые больше,

чем максимум главной диагонали. Поменять местами элементы

главной и побочной диагоналей.

// main and function

import java.util.Arrays;  
  
 */\*\*  
 \** ***@author*** *Goncharov Matthew  
 \** ***@version*** *0.1  
 \*/* public class laba\_18  
 {  
 public static void main(String[] args)  
 {  
 int[][] arr = *newArray*(5);  
 // Вывожу для проверки  
 System.*out*.println("\nPrint my array");  
 for (int[] line : arr)  
 {  
 System.*out*.println(Arrays.*toString*(line));  
 }  
 // Моя функция  
 arr = *replaceDiag*(arr);  
 System.*out*.println("\nReplace diagonal on array");  
 for (int[] line : arr)  
 {  
 System.*out*.println(Arrays.*toString*(line));  
 }  
 }  
 */\*\*  
 \* Создание массива  
 \** ***@param*** *size размер матрицы  
 \** ***@return*** *возвращает заполненый массив  
 \*/* public static int[][] newArray(int size)  
 {  
 int[][] arr = new int[size][size];  
 for (int i = 0; i < size; i++)  
 {  
 for (int j = 0; j < size; j++)  
 {  
 arr[i][j] = (int) (Math.*random*() \* 10);  
 }  
 }  
 if (arr == null || arr.length == 0) // Проверка на пустоту  
 {  
 return null;  
 }  
 return arr;  
 }  
 */\*\*  
 \* Функция для смены элементов на главной и побочной диагонали  
 \** ***@param*** *arr  
 \** ***@return*** *возвращает среднее арифмитическое  
 \*/* public static int[][] replaceDiag(int[][] arr)  
 {  
 if (arr == null || arr.length == 0)  
 {  
 return null;  
 }  
 int[] array\_glavn = new int[arr.length];  
 int[] array\_pobochn = new int[arr.length];  
  
 // Заполняю диагонали  
 for (int i = 0; i < arr.length; i++)  
 {  
 for (int j = 0 ; j < arr.length; j++)  
 {  
 if (i == j)  
 {  
 array\_glavn[i] = arr[i][j];  
 }  
 if (i + j == arr.length - 1)  
 {  
 array\_pobochn[i] = arr[i][j];  
 }  
 }  
 }  
 // replace  
 for (int i = 0; i < arr.length; i++)  
 {  
 for (int j = 0 ; j < arr.length; j++)  
 {  
 if (i == j)  
 {  
 arr[i][j] = array\_pobochn[i];  
 }  
 if (i + j == arr.length - 1)  
 {  
 arr[i][j] = array\_glavn[i];  
 }  
 }  
 }  
 return arr;  
 }  
 */\*\*  
 \* Функция по поиску max на диагонали  
 \** ***@param*** *arr  
 \** ***@return*** *max  
 \*/* public static int maxArr(int[][] arr)  
 {  
 int max = arr[0][0];  
 for (int i = 0; i < arr.length; i++)  
 {  
 for (int j = 0 ; j < arr.length; j++)  
 {  
 if (i == j)  
 {  
 if (max < arr[i][j])  
 {  
 max = arr[i][j];  
 }  
 }  
 }  
 }  
 return max;  
 }  
 }

// test

import org.junit.Assert;  
import org.junit.Test;  
  
  
public class tests {  
 @Test  
 public void TestNotNullCreate()  
 {  
 var array = laba\_18.*newArray*(5);  
 Assert.*assertNotNull*(array);  
 }  
 @Test  
 public void TestOnExistCreate()  
 {  
 var array = laba\_18.*newArray*(0);  
 Assert.*assertNull*(array);  
 }  
 @Test  
 public void TestOnExistWhenReplace()  
 {  
 int[][] arr = new int[0][0];  
 var array = laba\_18.*replaceDiag*(arr);  
 Assert.*assertEquals*(null,array);  
 }  
 @Test  
 public void TestValues()  
 {  
 int[][] arr = {  
 {1,0,0,2},  
 {0,1,2,0},  
 {0,2,1,0},  
 {2,0,0,1}};  
  
 var array = laba\_18.*replaceDiag*(arr);  
  
 // Итог  
 int[][] arr\_2 = {  
 {2,0,0,1},  
 {0,2,1,0},  
 {0,1,2,0},  
 {1,0,0,2}};  
  
 Boolean f = true;  
 // via flag  
 for (int i = 0; i < arr.length; i++)  
 {  
 for (int j = 0; j < arr.length; j++)  
 {  
 if (array[i][j] != arr\_2[i][j])  
 {  
 f = false;  
 break;  
 }  
 }  
 }  
 Assert.*assertEquals*(true,f);  
 }  
 @Test  
 public void TestValuesInvalid()  
 {  
 int[][] arr = {  
 {1,0,0,2},  
 {0,1,2,0},  
 {0,2,1,0},  
 {2,0,0,1}};  
  
 var array = laba\_18.*replaceDiag*(arr);  
  
 // Итог  
 int[][] arr\_2 = {  
 {-2,0,0,1},  
 {0,2,1,0},  
 {0,1,2,0},  
 {1,0,0,2}};  
  
 Boolean f = true;  
 // via flag  
 for (int i = 0; i < arr.length; i++)  
 {  
 for (int j = 0; j < arr.length; j++)  
 {  
 if (array[i][j] != arr\_2[i][j])  
 {  
 f = false;  
 break;  
 }  
 }  
 }  
 Assert.*assertNotEquals*(true,f);  
 }  
 @Test  
 public void TestMaxInArrayOnDiagonal()  
 {  
 int[][] arr = {  
 {0,0,0,0},  
 {0,0,0,0},  
 {0,0,9,0},  
 {0,0,0,0}};  
 var x = laba\_18.*maxArr*(arr);  
 Assert.*assertEquals*(9, x);  
 }  
}

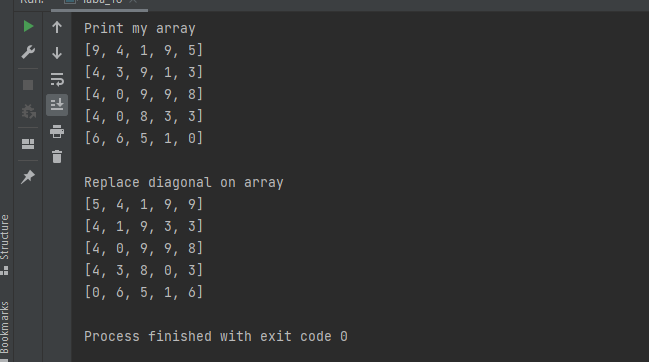
// output

Рисунок 1

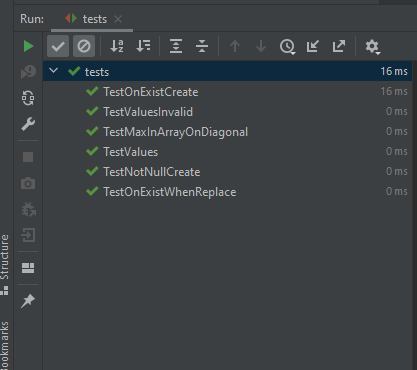


Рисунок 2