Министерство науки и высшего образования РФ

ФГАОУ ВО Пермский национальный исследовательский

политехнический университет

Кафедра «Вычислительная математика, механика и биомеханика»

Отчет по лабораторной работе № 5

тема «Массивы в Java»

по дисциплине «Информатика»

Выполнил: студент группы ИСТ-22-1б Михайлов Александр

Проверил: ассистент каф. ВММБ Нетбай Г.В.

Пермь, 2023

Оглавление

Оглавление

[Вариант 14 3](#_Toc136784846)

[Задание 1 3](#_Toc136784847)

[Постановка задачи 3](#_Toc136784848)

[Решение задачи 3](#_Toc136784849)

[Проверка результата 4](#_Toc136784850)

[Задание 2 5](#_Toc136784851)

[Постановка задачи 5](#_Toc136784852)

[Решение задачи 5](#_Toc136784853)

[Проверка результата 6](#_Toc136784854)

[Задание 3 7](#_Toc136784855)

[Постановка задачи 7](#_Toc136784856)

[Решение задачи 7](#_Toc136784857)

[Проверка результата 8](#_Toc136784858)

[Задание 4 9](#_Toc136784859)

[Постановка задачи 9](#_Toc136784860)

[Решение задачи 9](#_Toc136784861)

[Проверка результата 10](#_Toc136784862)

[Задание 5 11](#_Toc136784863)

[Постановка задачи 11](#_Toc136784864)

[Решение задачи 11](#_Toc136784865)

[Проверка результата 12](#_Toc136784866)

[Задание 6 13](#_Toc136784867)

[Постановка задачи 13](#_Toc136784868)

[Решение задачи 13](#_Toc136784869)

[Задание 7 15](#_Toc136784870)

[Постановка задачи 15](#_Toc136784871)

[Решение задачи 15](#_Toc136784872)

[Проверка результата 15](#_Toc136784873)

[Задание 8 16](#_Toc136784874)

[8.1. Постановка задачи 16](#_Toc136784875)

[8.2. Решение задачи, код программы 16](#_Toc136784876)

[8.3. Тестирование работы программы 17](#_Toc136784877)

[Задание 9 17](#_Toc136784878)

[9.1. Постановка задачи 18](#_Toc136784879)

[9.2. Решение задачи, код программы 18](#_Toc136784880)

[9.3. Тестирование работы программы с проверкой 19](#_Toc136784881)

[Задание 10 19](#_Toc136784882)

[10.1. Постановка задачи 20](#_Toc136784883)

[10.2. Решение задачи, код программы 20](#_Toc136784884)

[10.3. Тестирование работы программы с проверкой 21](#_Toc136784885)

[Задание 11 22](#_Toc136784886)

[11.1. Постановка задачи 22](#_Toc136784887)

[11.2. Решение задачи, код программы 22](#_Toc136784888)

[11.3. Тестирование работы программы с проверкой 23](#_Toc136784889)

[Задание 12 24](#_Toc136784890)

[12.1. Постановка задачи 24](#_Toc136784891)

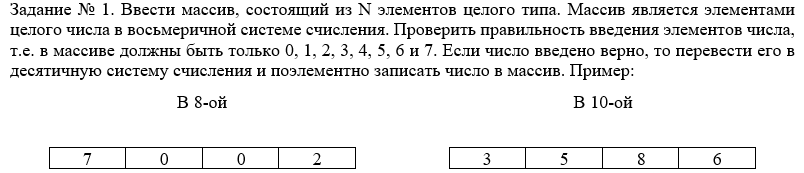
[12.2. Решение задачи, код программы 24](#_Toc136784892)

[12.3. Тестирование работы программы с проверкой 24](#_Toc136784893)

# Вариант 14

# Задание 1

## Постановка задачи



## Решение задачи

Код программы:

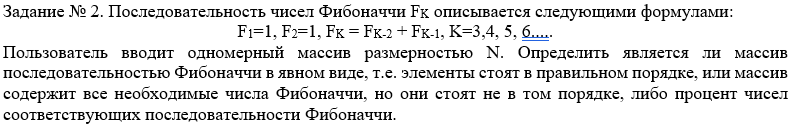
java.util.Scanner;  
  
public class Main {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
  
 System.*out*.print("Введите размер массива: ");  
 int n = scanner.nextInt();  
  
 int[] arr = new int[n];  
 boolean valid = true;  
  
 System.*out*.println("Введите элементы массива в восьмеричной системе счисления:");  
  
 for (int i = 0; i < n; i++) {  
 int num = scanner.nextInt();  
 if (num < 0 || num > 7) {  
 valid = false;  
 break;  
 }  
 arr[i] = num;  
 }  
  
 if (valid) {  
 int decimal = 0;  
 int power = n - 1;  
  
 for (int i = 0; i < n; i++) {  
 decimal += arr[i] \* Math.*pow*(8, power);  
 power--;  
 }  
  
 System.*out*.println("Число в десятичной системе счисления: " + decimal);  
 System.*out*.println("Элементы числа в массиве:");  
  
 for (int i = 0; i < n; i++) {  
 System.*out*.print(decimal % 10 + " ");  
 decimal /= 10;  
 }  
 } else {  
 System.*out*.println("Массив содержит некорректные элементы!");  
 }  
 }  
}

## Проверка результата

|  |  |
| --- | --- |
|  |  |

# Задание 2

## Постановка задачи



## Решение задачи

Код программы:

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Введите размер массива: ");

int n = scanner.nextInt();

int[] arr = new int[n];

System.out.println("Введите элементы массива:");

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

arr[i] = scanner.nextInt();

}

boolean isFibonacci = true;

int fibonacciCount = 0;

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

if (i == 0 || i == 1) {

if (arr[i] != 1) {

isFibonacci = false;

break;

}

} else {

int fibonacci = arr[i - 2] + arr[i - 1];

if (arr[i] != fibonacci) {

isFibonacci = false;

} else {

fibonacciCount++;

}

}

}

if (isFibonacci) {

System.out.println("Массив является последовательностью Фибоначчи!");

} else if (fibonacciCount == 0) {

System.out.println("Массив не является последовательностью Фибоначчи!");

} else {

double fibonacciPercentage = (double) fibonacciCount / n \* 100;

System.out.println("Массив содержит " + fibonacciCount + " чисел Фибоначчи (" + fibonacciPercentage + "%).");

}

}

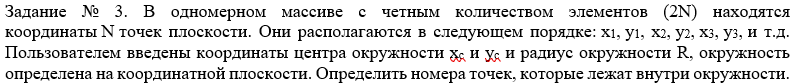
}

## Проверка результата

|  |  |
| --- | --- |
|  |  |

# Задание 3

## Постановка задачи



## Решение задачи

Код программы:

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Введите количество точек: ");

int n = scanner.nextInt();

double[] x = new double[n];

double[] y = new double[n];

System.out.println("Введите координаты точек:");

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

x[i] = scanner.nextDouble();

y[i] = scanner.nextDouble();

}

System.out.print("Введите координаты центра окружности: ");

double xc = scanner.nextDouble();

double yc = scanner.nextDouble();

System.out.print("Введите радиус окружности: ");

double r = scanner.nextDouble();

System.out.print("Номера точек, которые лежат внутри окружности: ");

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

double distance = Math.sqrt(Math.pow(x[i] - xc, 2) + Math.pow(y[i] - yc, 2));

if (distance < r) {

System.out.print((i + 1) + " ");

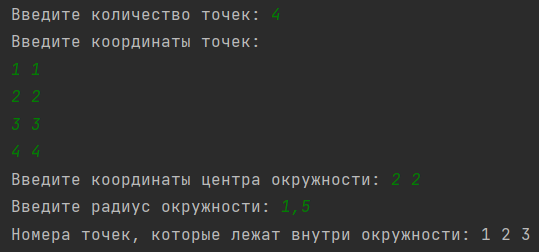
}

}

}

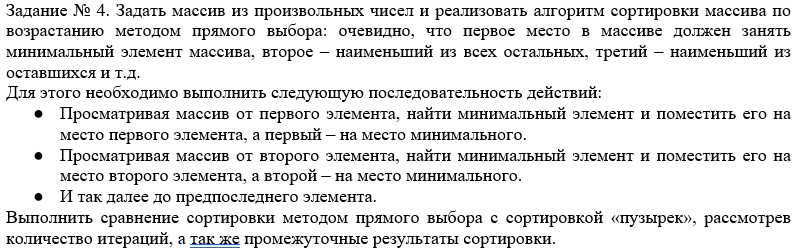
}

## Проверка результата



# Задание 4

## Постановка задачи



## Решение задачи

Код программы:

import java.util.Arrays;

public class Main {

public static void main(String[] args) {

int[] arr = {5, 2, 9, 1, 7, 4};

System.out.println("Исходный массив: " + Arrays.toString(arr));

selectionSort(arr);

System.out.println("Отсортированный массив методом прямого выбора: " + Arrays.toString(arr));

int[] arr2 = {5, 2, 9, 1, 7, 4};

bubbleSort(arr2);

System.out.println("Отсортированный массив методом пузырька: " + Arrays.toString(arr2));

}

public static void selectionSort(int[] arr) {

for (int i = 0; i < arr.length - 1; i++) {

int minIndex = i;

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

if (arr[j] < arr[minIndex]) {

minIndex = j;

}

}

int temp = arr[i];

arr[i] = arr[minIndex];

arr[minIndex] = temp;

}

}

public static void bubbleSort(int[] arr) {

boolean sorted = false;

int iterations = 0;

while (!sorted) {

sorted = true;

for (int i = 0; i < arr.length - 1; i++) {

if (arr[i] > arr[i + 1]) {

int temp = arr[i];

arr[i] = arr[i + 1];

arr[i + 1] = temp;

sorted = false;

}

}

iterations++;

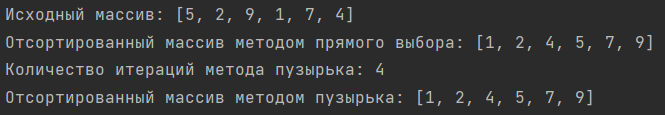
}

System.out.println("Количество итераций метода пузырька: " + iterations);

}

}

## Проверка результата



# Задание 5

## Постановка задачи



## Решение задачи

Код программы:

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Введите размер матрицы (n x n): ");

int n = scanner.nextInt();

int[][] matrix = new int[n][n];

System.out.println("Введите элементы матрицы:");

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

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

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

}

}

System.out.print("Введите номер столбца для замены: ");

int column = scanner.nextInt() - 1;

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

int temp = matrix[i][column];

matrix[i][column] = matrix[n - 1 - i][n - 1 - column];

matrix[n - 1 - i][n - 1 - column] = temp;

}

System.out.println("Матрица после замены:");

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

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

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

}

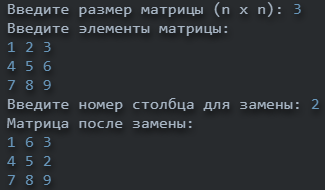
System.out.println();

}

}

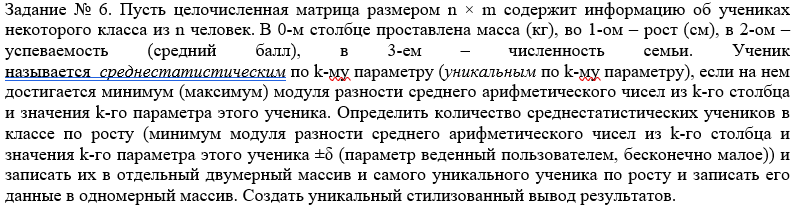
}

## Проверка результата



# Задание 6

## Постановка задачи



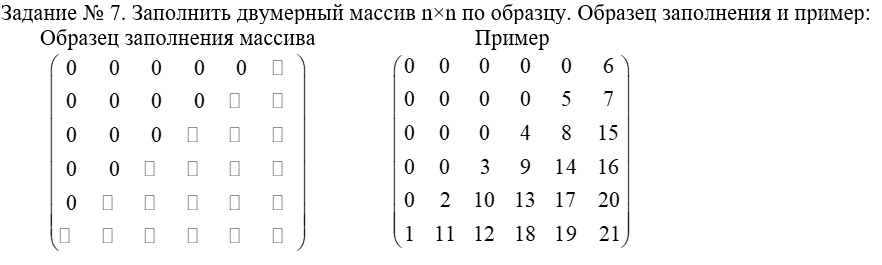
## Решение задачи

Код программы:

import java.util.Arrays;  
  
public class Main {  
 public static void main(String[] args) {  
  
 int n = 10;  
 int m = 4;  
 int[][] matrix = new int[n][m];  
 matrix[0] = new int[]{60, 170, 4, 4};  
 matrix[1] = new int[]{70, 180, 5, 3};  
 matrix[2] = new int[]{55, 165, 3, 2};  
  
  
 double sum = 0;  
 for (int i = 0; i < n; i++) {  
 sum += matrix[i][1];  
 }  
 double average = sum / n;  
  
 double delta = 0.1;  
  
 int count = 0;  
 int[][] result = new int[n][m];  
 int[] unique = new int[m];  
  
 double minDiff = Double.*MAX\_VALUE*;  
 for (int i = 0; i < n; i++) {  
 double diff = Math.*abs*(average - matrix[i][1]);  
 if (diff <= delta) {  
 result[count] = matrix[i];  
 count++;  
  
 double curDiff = Math.*abs*(average - matrix[i][1]);  
 if (curDiff < minDiff) {  
 minDiff = curDiff;  
 unique = matrix[i];  
 }  
 }  
 }  
  
 System.*out*.println("Количество среднестатистических учеников по росту: " + count);  
 System.*out*.println("Среднестатистические ученики по росту:");  
 for (int i = 0; i < count; i++) {  
 System.*out*.println(Arrays.*toString*(result[i]));  
 }  
 System.*out*.println("Самый уникальный ученик по росту: " + Arrays.*toString*(unique));  
 }  
}

# Задание 7

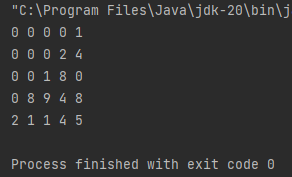
## Постановка задачи



## Решение задачи

import java.util.Arrays;  
  
public class Main {  
 public static void main(String[] args) {  
  
 int n = 5;  
 int[][] matrix = new int[n][n];  
  
 for (int i = 0; i < n; i++) {  
 for (int j = 0; j < n; j++) {  
 if (j < n - i - 1) {  
 matrix[i][j] = 0;  
 } else {  
 matrix[i][j] = (int) (Math.*random*() \* 10);  
 }  
 }  
 }  
  
 for (int i = 0; i < n; i++) {  
 for (int j = 0; j < n; j++) {  
 System.*out*.print(matrix[i][j] + " ");  
 }  
 System.*out*.println();  
 }  
 }  
 }

## Проверка результата



# Задание 8

## 8.1. Постановка задачи

Задание № 8. Заполнить двумерный массив n×n по образцу. Матрица заполняется автоматически для любого n. Создать метод заполнения блока матрицы и отпараметризировав заполнение заполнить при помощи метода всю матрицу. Образец заполнения:



## 8.2. Решение задачи, код программы

import java.util.Scanner;  
  
public class laba8\_8 {  
 public static void main(String[] args) {  
 Scanner in = new Scanner(System.*in*);  
 System.*out*.println("Введите n");  
 int n = in.nextInt();  
 int [][] arr = new int[n][n];  
 int k\_max = (n \* n - n)/2 + n;  
 *Shtuka*(arr,0,0,n/2 - 1, n);  
 *Shtuka*(arr,0,n/2,n/2 - 1, n);  
 *Shtuka*(arr,n/2,0,n/2 - 1, n);  
 *Shtuka*(arr,n/2,n/2,n/2 - 1, n);  
 *PrintArr*(arr);  
 }  
 public static void Shtuka(int [][] arr,int x, int y, int m, int n){  
 int k = 1;  
 for (int i = 0; i < n/2; i++){  
 for (int j = x + 2 \* i, b = y + i;j <= x + m - i;j++){  
 arr[b][j] = k++;  
 }  
 for (int j =x + m - i, b = y + i + 1; b <=y + m - 2 \* i;b++){  
 arr[b][j] = k++;  
 }  
 for (int j = x + m - 1 - i, b = y + m - 1 - 2 \* i; j > x + 2 \* i; j--, b--){  
 arr[b][j] = k++;  
 }  
 }  
 }  
 public static void PrintArr(int[][] arr){  
 for (int i = 0; i<= arr.length-1; i++) {  
 for (int j = 0; j <= arr[0].length-1; j++){  
 System.*out*.print(arr[i][j] + "\t");  
 }  
 System.*out*.println();  
 }  
 }  
}

## 8.3. Тестирование работы программы

# Задание 9

## 9.1. Постановка задачи

Задание № 9. Заполнить двумерный массив 2n×2n по образцу. Матрица заполняется автоматически для любого n. Создать метод заполнения блоков матрицы и отпараметризировав выполнить заполнение всей матрицы при помощи методов. Образец заполнения:



## 9.2. Решение задачи, код программы

import java.util.Scanner;  
  
public class oliver2 {  
 public static void main(String[] args) {  
 Scanner in = new Scanner(System.*in*);  
 System.*out*.println("input n");  
 int n = in.nextInt();  
 int [][] arr = new int[n][n];  
 *FillTheBlock\_1*(arr,0,0,n/2 - 1);  
 *FillTheBlock\_1*(arr,0,n/2,n - 1);  
 *FillTheBlock\_2*(arr,n/2, 0,n - 1, n/2 \* n/2 ,n/2, n);  
 *FillTheBlock\_2*(arr,n/2, n/2,n - 1, n/2 \* n/2 ,n, n);  
 *PrintArr*(arr);  
 }  
 public static void FillTheBlock\_1(int [][] arr, int x, int y, int m){  
 int k = 1;  
 for (int lp = 0,c = x; lp <= m/2; lp++,c += 2){  
 for (int i = y + 2 \* lp; i <= m; i ++){  
 arr[i][c] = k++;  
 }  
 for (int j = m, j\_y = c + 1; j >= y + 2 \* lp + 1; j--){  
 arr[j][j\_y] =k++;  
 }  
 }  
 }  
 public static void FillTheBlock\_2(int [][] arr,int x, int y, int m, int max, int end\_y, int size){ // x = 4 y = 0 m = 7  
 int k = 1;int lp = 0;  
 while (k < max){  
 for (int i = end\_y - lp - 1, j = m - lp; i >= y + lp; i--){  
 arr[i][j] = k++;  
 }  
 for (int i = m - lp - 1, j = y + lp; i >= x + lp; i--){  
 arr[j][i] = k++;  
 }  
 for (int i = y + lp + 1, j = x + lp; i < end\_y - lp; i++){  
 arr[i][j] = k++;  
 }  
 for (int i = x + lp + 1, j = end\_y - lp - 1; i < m - lp; i++){  
 arr[j][i] = k++;  
 }  
 lp++;  
 if ((size/2) % 2 != 0){  
 arr[y + lp][x + lp] = max;  
 }  
 }  
 }  
 public static void PrintArr(int[][] arr){  
 for (int[] ints : arr) {  
 for (int anInt : ints) {  
 System.*out*.print(anInt + "\t");  
 }  
 System.*out*.println();  
 }  
 }  
}

## 9.3. Тестирование работы программы с проверкой

# Задание 10

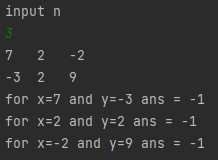
## 10.1. Постановка задачи

Задание № 10. Дана матрица с N точками в пространстве, 0-я строчка координаты х, 1-я строчка координаты y. Определить для каждой точки попала ли она в закрашенную область, с использованием массива N элементов (1 – точка попала внутрь области, 0 – точка на границе области, -1 – точка вне области). Области взять из лабораторной работы № 2 задание 5 (условия проверки реализовать как метод). Создать метод для стилизованного вывода ответов по попаданию точки в область в соответствии с массивом данных о попадании точки.

## 10.2. Решение задачи, код программы

import java.util.Random;  
import java.util.Scanner;  
  
public class oliver2 {  
 public static void main(String[] args) {  
 Scanner in = new Scanner(System.*in*);  
 System.*out*.println("input n");  
 int n = in.nextInt();  
 int[][] arr = new int[2][n];  
 *getArr*(arr);  
 *PrintArr*(arr);  
 *PrintAns*(n,arr);  
 }  
 public static void getArr( int[][] arr){  
 Random r = new Random();  
 for (int i = 0; i < arr.length; i++) {  
 for (int j = 0; j < arr[i].length; j++) {  
 arr[i][j] = r.nextInt(-10, 10);  
 }  
 }  
 }  
 public static void PrintArr( int[][] arr){  
 for (int[] ints : arr) {  
 for (int anInt : ints) {  
 System.*out*.print(anInt + "\t");  
 }  
 System.*out*.println();  
 }  
 }  
 public static boolean f1(double x, double y){  
 return ((y > -6 \* x - 54) & (y > 0.67 \* x - 0.67) & (y > 3.67 \* x + 14.33)  
 & (y < 2 \* x + 18) & (y < 0.25 \* x + 7.5))&(!((y > 1.4 \* x + 9.8) & (y < 6 \* x + 42 & y >= 0 & x <= 6)));  
 }  
 public static boolean f2(double x, double y){  
 return ((y > -1.67 \* x - 5.33) & (y > 2.5 \* x - 9.5) &  
 (y < 6 \* x + 10) & (y < 0.4 \* x + 4.4) || (y > - 0.33 \* x - 1 & x >= 3 & x <= 6))&(!((y < x + 2) & (y > - 0.83 \* x + 2)));  
 }  
 public static boolean f1edge (double x, double y){  
 return ((y == -6 \* x - 54) || (y == 0.67 \* x - 0.67) || (y == 3.67 \* x + 14.33)  
 || (y == 2 \* x + 18) || (y == 0.25 \* x + 7.5));  
 }  
 public static boolean f2edge(double x, double y){  
 return (y == -1.67 \* x - 5.33) || (y > 2.5 \* x - 9.5) ||  
 (y == 6 \* x + 10) || (y == 0.4 \* x + 4.4) || (y == -0.33 \* x - 1 & x >= 3 & x <= 6);  
 }  
 public static void PrintAns (int n, int[][] arr){  
 for (int i = 0; i < n; i++){  
 if ((*f1*(arr[0][i], arr[1][i])&*f1edge*(arr[0][i], arr[1][i])) || *f2*(arr[0][i], arr[1][i])&*f2edge*(arr[0][i], arr[1][i]))  
 System.*out*.println("for x=" + arr[0][i] + " and y=" + arr[1][i] + " ans = 0 \t");  
 else if (*f1*(arr[0][i], arr[1][i]) || *f2*(arr[0][i], arr[1][i])) {  
 System.*out*.println("for x=" + arr[0][i] + " and y=" + arr[1][i] + " ans = 1 \t");  
 }  
 else System.*out*.println("for x=" + arr[0][i] + " and y=" + arr[1][i] + " ans = -1 \t");  
 }  
 }  
}

## 10.3. Тестирование работы программы с проверкой



# Задание 11

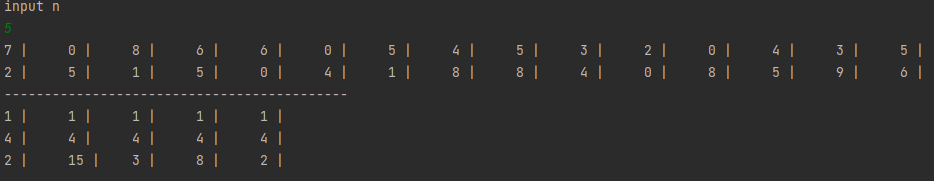
## 11.1. Постановка задачи

Задание № 11. Дан двумерный массив 2×3N, где N – количество предполагаемых треугольников. Пользователь вводит данные по координатам вершин треугольников в двумерный массив. Создать двумерный массив ответов для N предполагаемых треугольников: 0-я строчка результаты проверки о существовании треугольника с введенными вершинами (1 – существует; 0 – не существует); 1-я строчка классификация треугольников (1 – равносторонний; 2 – прямоугольный; 3 – равнобедренный; 4 – произвольный; 0 – если такого треугольника не существует); 2-я строчка площадь треугольника, если он существует, иначе 0. Создать метод для стилизованного вывода ответов. Проверить есть ли треугольники одинаковой площади, вывести ответ, если такие треугольники есть, то указать их номера.

## 11.2. Решение задачи, код программы

import java.util.\*;  
  
import static java.lang.Math.\*;  
  
public class oliver2 {  
 public static void main(String[] args){  
 Scanner in = new Scanner(System.*in*);  
 System.*out*.println("input n");  
 int n = in.nextInt();  
 int [][] arr = new int[2][3 \* n];  
 *getArr*(arr);  
 *PrintOutInt*(arr);  
 System.*out*.println("-------------------------------------------");  
 double s = 0; double ab; double bc; double ca;  
 double [][] ans = new double[3][n];  
 for (int j = 0, i = 0; j < 3 \* n; j += 3, i++){  
 ab = *sqrt*(*pow*(arr[0][j] - arr[0][j + 1], 2) + *pow*(arr[1][j] - arr[1][j + 1], 2));  
 bc = *sqrt*(*pow*(arr[0][j + 1] - arr[0][j + 2], 2) + *pow*(arr[1][j + 1] - arr[1][j + 2], 2));  
 ca = *sqrt*(*pow*(arr[0][j + 2] - arr[0][j], 2) + *pow*(arr[1][j + 2] - arr[1][j], 2));  
 s = 0.5 \* *abs*((arr[0][j] - arr[0][j + 2]) \* (arr[1][j + 1] - arr[1][j + 2]) - (arr[0][j + 1] - arr[0][j + 2]) \* (arr[1][j] - arr[1][j + 2]));  
 if (s == 0){  
 ans[0][i] = 0;  
 ans[1][i] = 0;  
 }  
 else {  
 ans[0][i] = 1;  
 ans[2][i] = s;  
 }  
 if ((ab == bc & bc == ca) & s != 0) ans[1][i] = 1;  
 else if ((ab == bc || bc == ca || ca == ab) & s != 0) ans[1][i] = 3;  
 else if ((*pow*(ab,2) == *pow*(bc,2) + *pow*(ca, 2) || *pow*(bc,2) == *pow*(ab,2) + *pow*(ca, 2) || *pow*(ca,2) == *pow*(bc,2) + *pow*(ab, 2)) & s!= 0) ans[1][i] = 2;  
 else if (s != 0) ans[1][i] = 4;  
  
 }  
 *PrintOut*(ans);  
  
 for (int i = 0; i < n; i++){  
 for (int j = 0; j < n; j++){  
 if (ans[2][i] == ans[2][j] & i != j) System.*out*.println("area of triangle " + (i + 1) + " equals to triangle " + (j + 1));  
 }  
 }  
 }  
 public static void getArr(int [][] arr){  
 Random r = new Random();  
 for (int i = 0; i < arr.length; i++) {  
 for (int j = 0; j < arr[i].length; j++) {  
 arr[i][j] = r.nextInt(10);  
 }  
 }  
 }  
 public static void PrintOut(double[][] arr){  
 for (double[] ints : arr) {  
 for (double anInt : ints) {  
 System.*out*.print((int)anInt + " | \t");  
 }  
 System.*out*.println();  
 }  
 }  
 public static void PrintOutInt(int[][] arr){  
 for (int[] ints : arr) {  
 for (int anInt : ints) {  
 System.*out*.print((int)anInt + " | \t");  
 }  
 System.*out*.println();  
 }  
 }  
}

## 11.3. Тестирование работы программы с проверкой



# Задание 12

## 12.1. Постановка задачи

Задание № 12. Дан двумерный квадратный массив A и двумерный квадратный массив обратный к первому A-1. При этом пользователь вводит размерность массива и данные сам и может допустить ошибку при вычислении обратной матрицы или при вводе. Поэтому выполнить проверку соответствуют ли массивы свойству A\* A-1=E, где E – это единичная матрица.

## 12.2. Решение задачи, код программы

import java.util.Scanner;  
import static java.lang.System.*exit*;  
  
public class oliver2 {  
 public static void main(String[] args){  
 Scanner in = new Scanner(System.*in*);  
 System.*out*.println("input n");  
 int n = in.nextInt();  
 double [][] arr = new double[n][n];  
 double [][] arr\_1 = new double[n][n];  
 System.*out*.println("input elements of array");  
 for (int i = 0; i < arr.length; i++) {  
 for (int j = 0; j < arr[i].length; j++) {  
 arr[i][j] = in.nextDouble();  
 }  
 }  
 System.*out*.println("input elements of array\_1");  
 for (int i = 0; i < arr\_1.length; i++) {  
 for (int j = 0; j < arr\_1[i].length; j++) {  
 arr\_1[i][j] = in.nextDouble();  
 }  
 }  
 int [][] e = new int[n][n];  
 for (int i = 0; i < n; i++){  
 for (int j = 0; j < n; j++){  
 for (int k = 0; k < n; k++){  
 e[i][j] += arr[i][k] \* arr\_1[k][j];  
 }  
 }  
 }  
 for (int i = 0; i < n; i++){  
 for (int j = 0; j < i; j++){  
 if (e[i][i] != 1 || e[i][j] != 0 || e[n - i - 1][n - 1 - j] != 0) {  
 System.*out*.println("Wrong input");  
 *exit*(0);  
 }  
 }  
 }  
 System.*out*.println("Everything is good!");  
 }  
}

## 12.3. Тестирование работы программы с проверкой

|  |  |
| --- | --- |
| № п.п. | Решение Java |
| 1 |  |
| 2 |  |