Министерство науки и высшего образования РФ

ФГАОУ ВО Пермский национальный исследовательский

политехнический университет

Кафедра «Вычислительная математика, механика и биомеханика»

Отчет по лабораторной работе № 8

тема «Одномерные массивы и матрицы»

по дисциплине «Информатика»

Выполнил: студент группы ИСТ-22-1б Васин М.А.

Проверил: ассистент каф. ВММБ Нетбай Г.В.

Пермь, 2023

**Содержание**

[Задание 1 4](#_Toc135178172)

[1.1. Постановка задачи 4](#_Toc135178173)

[1.2. Решение задачи, код программы 4](#_Toc135178174)

[1.3. Тестирование работы программы 5](#_Toc135178175)

[Задание 2 5](#_Toc135178176)

[2.1. Постановка задачи 5](#_Toc135178177)

[2.2. Решение задачи, код программы 5](#_Toc135178178)

[2.3. Тестирование работы программы 5](#_Toc135178179)

[Задание 3 5](#_Toc135178180)

[3.1. Постановка задачи 5](#_Toc135178181)

[3.2 Решение задачи, код программы 5](#_Toc135178182)

[3.3. Тестирование работы программы 6](#_Toc135178183)

[Задание 4 6](#_Toc135178184)

[4.1. Постановка задачи 6](#_Toc135178185)

[4.2. Решение задачи, код программы 7](#_Toc135178186)

[4.3. Тестирование работы программы 7](#_Toc135178187)

[Задание 5 8](#_Toc135178188)

[5.1. Постановка задачи 8](#_Toc135178189)

[5.2. Решение задачи, код программы 8](#_Toc135178190)

[5.3. Тестирование работы программы 9](#_Toc135178191)

[Задание 6 9](#_Toc135178192)

[6.1. Постановка задачи 9](#_Toc135178193)

[6.2. Решение задачи, код программы 9](#_Toc135178194)

[6.3. Тестирование работы программы 10](#_Toc135178195)

[Задание 7 10](#_Toc135178196)

[7.1. Постановка задачи 10](#_Toc135178197)

[7.2.1 Решение задачи, код программы 10](#_Toc135178198)

[7.3. Тестирование работы программы 11](#_Toc135178199)

[Задание 8 11](#_Toc135178200)

[8.1. Постановка задачи 11](#_Toc135178201)

[8.2. Решение задачи, код программы 11](#_Toc135178202)

[8.3. Тестирование работы программы 12](#_Toc135178203)

[Задание 9 12](#_Toc135178204)

[9.1. Постановка задачи 12](#_Toc135178205)

[9.2. Решение задачи, код программы 13](#_Toc135178206)

[9.3. Тестирование работы программы 14](#_Toc135178207)

[Задание 10 14](#_Toc135178208)

[10.1. Постановка задачи 14](#_Toc135178209)

[10.2. Решение задачи, код программы 14](#_Toc135178210)

[10.3. Тестирование работы программы 15](#_Toc135178211)

[Задание 11 15](#_Toc135178212)

[11.1. Постановка задачи 15](#_Toc135178213)

[11.2. Решение задачи, код программы 15](#_Toc135178214)

[11.3. Тестирование работы программы 15](#_Toc135178215)

[Задание 12 16](#_Toc135178216)

[12.1. Постановка задачи 16](#_Toc135178217)

[12.2. Решение задачи, код программы 16](#_Toc135178218)

[12.3. Тестирование работы программы 16](#_Toc135178219)

# Задание 1

## 1.1. Постановка задачи

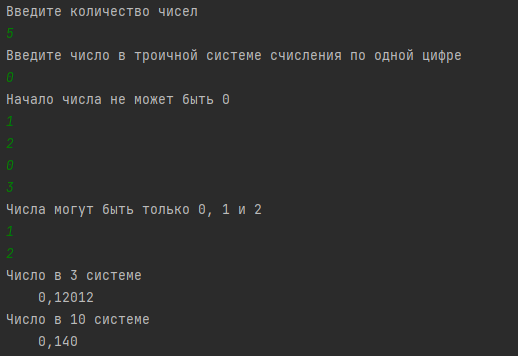
Ввести массив, состоящий из N элементов целого типа. Массив является элементами целого дробной части числа в троичной системе счисления. Проверить правильность введения элементов числа, т.е. в массиве должны быть только 0, 1 и 2. Считается, что в массиве ведены только те цифры, что стоят после запятой. Если число введено верно, то перевести его в десятичную систему счисления и поэлементно записать число в массив, так же только цифры после запятой (ограничиваясь 6 цифрами после запятой, если число при переводе не конечно). Пример:

|  |  |
| --- | --- |
| В 3-ой | В 10-ой |
| |  |  | | --- | --- | | 2 | 1 | | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 7 | 7 | 7 | 7 | 7 | 7 | |
| 0,21 | 0, 777777....... |

## 1.2. Решение задачи, код программы

import java.util.ArrayList;  
import java.util.Scanner;  
import static java.lang.Math.\*;  
public class z1 {  
 public static void main(String[] args) {  
 Scanner scan = new Scanner(System.*in*);  
 System.*out*.println("Введите количество чисел");  
 int n = scan.nextInt();  
 int doe\_ch [] = new int[n];  
 ArrayList< int[] > Chislo10 = new ArrayList< int[] >();  
 int sch = 0;  
 System.*out*.println("Введите число в троичной системе счисления по одной цифре");  
 for (int i = 0; i<n; i++) {  
 doe\_ch[i] = scan.nextInt();  
 while (i==0 & (doe\_ch[i]==0 || doe\_ch[i]>2 || doe\_ch[i]<0)) {  
 if (i==0 & doe\_ch[i]==0)  
 while (i==0 & doe\_ch[i]==0) {  
 System.*out*.println("Начало числа не может быть 0");  
 doe\_ch[i] = scan.nextInt();  
 }  
 if (i==0 & (doe\_ch[i] > 2 || doe\_ch[i] < 0))  
 while (i==0 & (doe\_ch[i]>2 || doe\_ch[i]<0)) {  
 System.*out*.println("Число может начинаться только с 1 и 2");  
 doe\_ch[i] = scan.nextInt();  
 }  
 }  
 if (i>0 & (doe\_ch[i]>2 || doe\_ch[i] <0))  
 while (i>0 & (doe\_ch[i]>2 || doe\_ch[i]<0)) {  
 System.*out*.println("Числа могут быть только 0, 1 и 2");  
 doe\_ch[i] = scan.nextInt();  
 }  
 }  
 int TempChislo10 = 0;  
 for (int i = n-1, j=0; i>=0; i--, j++) {  
 TempChislo10 += (int) (doe\_ch[i] \* *pow*(3, j)); }  
 while (TempChislo10 > 0) {  
 Chislo10.add(new int[1]);  
 Chislo10.get(sch)[0] = TempChislo10 % 10;  
 TempChislo10 /= 10;  
 sch++; }  
 int Desyt\_ch[] = new int [sch];  
 for (int i = sch-1, j=0; i>=0; i--, j++) {  
 Desyt\_ch[j] = Chislo10.get(i)[0]; }  
 System.*out*.print("Число в 3 системе\n\t0,");  
 for (int i=0; i<n; i++)  
 System.*out*.print(doe\_ch[i]);  
 System.*out*.print("\nЧисло в 10 системе\n\t0,");  
 for(int i=0; i<sch; i++)  
 System.*out*.print(Desyt\_ch[i]);  
 }  
}

## 1.3. Тестирование работы программы



# Задание 2

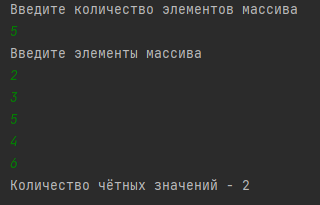
## 2.1. Постановка задачи

Ввести массив, состоящий из N элементов целого типа. Найти количество элементов четных по значению, которые стоят в четных по номерам ячейках массива.

## 2.2. Решение задачи, код программы

import java.util.Scanner;  
public class z2 {  
 public static void main(String[] args) {  
 Scanner scan = new Scanner(System.*in*);  
 System.*out*.println("Введите количество элементов массива");  
 int n = scan.nextInt();  
 int mass[] = new int[n];  
 System.*out*.println("Введите элементы массива");  
 for(int i =0; i<n; i++) {  
 mass[i] = scan.nextInt();  
 }  
 int sch = 0;  
 for(int i =0; i<n; i++) {  
 if(i%2==0)  
 if(mass[i]%2==0)  
 sch++;  
 }  
 System.*out*.println("Количество чётных значений - "+sch);  
 }  
}

## 2.3. Тестирование работы программы



# Задание 3

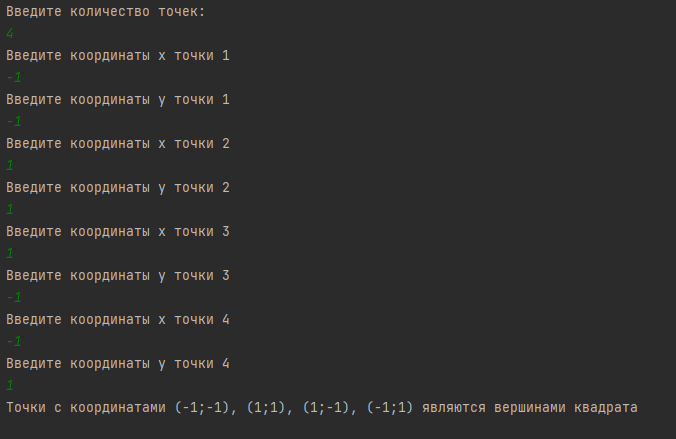
## 3.1. Постановка задачи

В одномерном массиве с четным количеством элементов (2N) находятся координаты N точек плоскости. Они располагаются в следующем порядке: x1, y1, х2, y2, x3, y3, и т.д. Определить номера точек, которые могут являться вершинами квадрата.

## 3.2 Решение задачи, код программы

import java.util.Scanner;  
public class z3 {  
 public static void main(String[] args) {  
 Scanner input = new Scanner(System.*in*);  
 System.*out*.println("Введите количество точек:");  
 int N = input.nextInt();  
 int[] coord = new int[2\*N];  
 for (int i = 0; i < 2 \* N; i++) {  
 if (i % 2 == 0) {  
 System.*out*.println("Введите координаты x точки " + ((i / 2) + 1));  
 } else {  
 System.*out*.println("Введите координаты y точки " + ((i / 2) + 1));  
 }  
 coord[i] = input.nextInt();  
 }  
  
 for (int i = 0; i < (2 \* N - 4); i += 2) {  
 for (int j = i + 2; j < (2 \* N - 3); j += 2) {  
 for (int k = j + 2; k < (2 \* N - 2); k += 2) {  
 for (int l = k + 2; l < (2 \* N - 1); l += 2) {  
 if (((coord[i] == coord[j]) & (coord[k] == coord[l]) &  
 (coord[i + 1] == coord[l + 1]) & (coord[j + 1] == coord[k + 1])) ||  
 ((coord[i] == coord[l]) & (coord[k] == coord[l]) &  
 (coord[i + 1] == coord[j + 1]) & (coord[l + 1] == coord[k + 1])) ||  
 ((coord[i] == coord[l]) & (coord[k] == coord[j]) &  
 (coord[i + 1] == coord[k + 1]) & (coord[j + 1] == coord[l + 1]))) {  
 System.*out*.println("Точки с координатами (" + coord[i] + ";" + coord[i + 1] + "), (" +  
 coord[j] + ";" + coord[j + 1] + "), (" + coord[k] + ";" + coord[k + 1] +  
 "), (" + coord[l] + ";" + coord[l + 1] + ") являются вершинами квадрата");  
 }  
 }  
 }  
 }  
 }  
 }  
}

## 3.3. Тестирование работы программы



# Задание 4

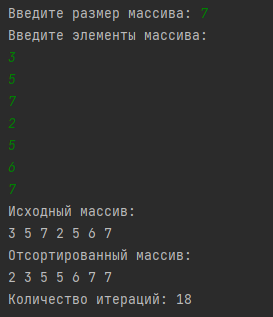
## 4.1. Постановка задачи

Задать массив из произвольных чисел и реализовать алгоритм «шейкер-сортировки». Образно алгоритм можно описать так: на каждом шаге основного цикла рассматривается массив данных A[left,right], по двум внутренним циклам: 1) от left к right A[i]>=A[i+1] тогда они меняются местами, иначе ничего не происходит, после окончания цикла right = right-1; 2) от right к left A[i-1]>=A[i] тогда они меняются местами, иначе ничего не происходит, после окончания массива left = left+1. После 1 шага минимальный и максимальный элемент в исходном массиве перетекают к краям, минимальный в — A[left], максимальный – в A[right]. Остановкой итерационной процедуры является равенство 0 перестановок элементов в массиве. Выполнить сравнение «шейкер-сортировки» с сортировкой «пузырек», рассмотрев количество итераций, а так же промежуточные результаты сортировки.

## 4.2. Решение задачи, код программы

import java.util.Scanner;  
public class z4 {  
 public static void main(String[] args) {  
 Scanner input = new Scanner(System.*in*);  
 System.*out*.print("Введите размер массива: ");  
 int size = input.nextInt();  
  
 int[] array = new int[size];  
 System.*out*.println("Введите элементы массива:");  
 for (int i = 0; i < size; i++) {  
 array[i] = input.nextInt();  
 }  
  
 System.*out*.println("Исходный массив:");  
 *printArray*(array);  
  
 int iterations = *shakerSort*(array);  
  
 System.*out*.println("Отсортированный массив:");  
 *printArray*(array);  
 System.*out*.println("Количество итераций: " + iterations);  
 }  
  
 public static int shakerSort(int[] array) {  
 int left = 0;  
 int right = array.length - 1;  
 boolean swapped;  
 int iterations = 0;  
  
 do {  
 swapped = false;  
  
 for (int i = left; i < right; i++) {  
 if (array[i] > array[i + 1]) {  
 *swap*(array, i, i + 1);  
 swapped = true;  
 }  
 iterations++;  
 }  
 right--;  
  
 for (int i = right; i > left; i--) {  
 if (array[i - 1] > array[i]) {  
 *swap*(array, i - 1, i);  
 swapped = true;  
 }  
 iterations++;  
 }  
 left++;  
 } while (swapped);  
  
 return iterations;  
 }  
  
 public static void swap(int[] array, int i, int j) {  
 int temp = array[i];  
 array[i] = array[j];  
 array[j] = temp;  
 }  
  
 public static void printArray(int[] array) {  
 for (int num : array) {  
 System.*out*.print(num + " ");  
 }  
 System.*out*.println();  
 }  
}

## 4.3. Тестирование работы программы



# Задание 5

## 5.1. Постановка задачи

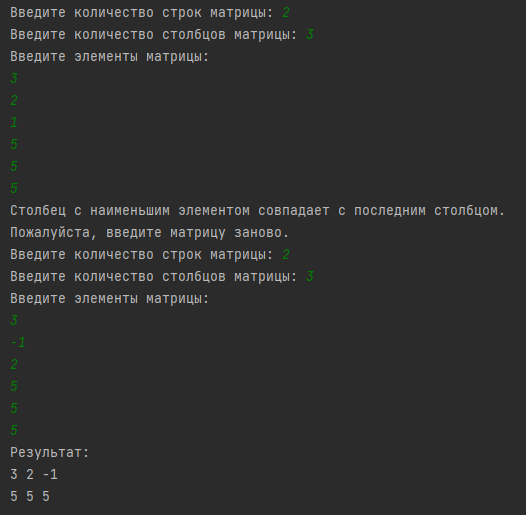
В заданной матрице поменять местами последний столбец и столбец, содержащий наименьший элемент матрицы. Если столбец, содержащий наименьший элемент матрицы, совпадает с последним столбцом, тогда выполнить генерацию двумерного массива еще раз и т.д. пока столбец, содержащий наименьший элемент матрицы, не будет совпадать с последним столбцом.

## 5.2. Решение задачи, код программы

import java.util.Scanner;  
public class z5 {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
  
 boolean swap = true;  
 while (swap) {  
 System.*out*.print("Введите количество строк матрицы: ");  
 int rows = scanner.nextInt();  
  
 System.*out*.print("Введите количество столбцов матрицы: ");  
 int columns = scanner.nextInt();  
  
 int[][] matrix = new int[rows][columns];  
  
 System.*out*.println("Введите элементы матрицы:");  
 for (int i = 0; i < rows; i++) {  
 for (int j = 0; j < columns; j++) {  
 matrix[i][j] = scanner.nextInt();  
 }  
 }  
  
 int minColumn = *findMinColumn*(matrix);  
  
 if (minColumn == columns - 1) {  
 System.*out*.println("Столбец с наименьшим элементом совпадает с последним столбцом.");  
 System.*out*.println("Пожалуйста, введите матрицу заново.");  
 } else {  
 *swapColumns*(matrix, minColumn, columns - 1);  
 swap = false;  
  
 System.*out*.println("Результат:");  
 *printMatrix*(matrix);  
 }  
 }  
 }  
  
 private static int findMinColumn(int[][] matrix) {  
 int minColumn = 0;  
 int minElement = matrix[0][0];  
  
 for (int j = 1; j < matrix[0].length; j++) {  
 if (matrix[0][j] < minElement) {  
 minElement = matrix[0][j];  
 minColumn = j;  
 }  
 }  
  
 return minColumn;  
 }  
  
 private static void swapColumns(int[][] matrix, int column1, int column2) {  
 for (int i = 0; i < matrix.length; i++) {  
 int temp = matrix[i][column1];  
 matrix[i][column1] = matrix[i][column2];  
 matrix[i][column2] = temp;  
 }  
 }  
  
 private static void printMatrix(int[][] matrix) {  
 for (int[] row : matrix) {  
 for (int value : row) {  
 System.*out*.print(value + " ");  
 }  
 System.*out*.println();  
 }  
 }  
}

}

## 5.3. Тестирование работы программы



# Задание 6

## 6.1. Постановка задачи

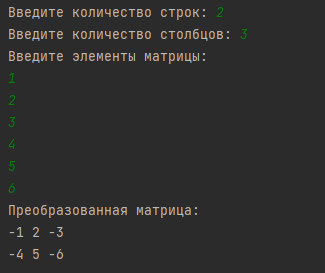
Пусть дана действительная матрица размером n × m. Требуется преобразовать матрицу: если в строчке есть отрицательные элементы, то поэлементно вычесть последнюю строку, если в строке нет отрицательных элементов, тогда заменить каждый четный по столбцу элемент на противоположный по знаку.

## 6.2. Решение задачи, код программы

import java.util.Scanner;  
public class z6 {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
  
 System.*out*.print("Введите количество строк: ");  
 int n = scanner.nextInt();  
  
 System.*out*.print("Введите количество столбцов: ");  
 int m = scanner.nextInt();  
  
 int[][] matrix = new int[n][m];  
  
 System.*out*.println("Введите элементы матрицы:");  
   
 for (int i = 0; i < n; i++) {  
 for (int j = 0; j < m; j++) {  
 matrix[i][j] = scanner.nextInt();  
 }  
 }  
   
 for (int i = 0; i < n; i++) {  
 boolean hasNegative = false;  
 for (int j = 0; j < m; j++) {  
 if (matrix[i][j] < 0) {  
 hasNegative = true;  
 break;  
 }  
 }  
  
 if (hasNegative) {  
 for (int j = 0; j < m; j++) {  
 matrix[i][j] -= matrix[n - 1][j];  
 }  
 } else {  
 for (int j = 0; j < m; j += 2) {  
 matrix[i][j] = -matrix[i][j];  
 }  
 }  
 }  
   
 System.*out*.println("Преобразованная матрица:");  
 for (int i = 0; i < n; i++) {  
 for (int j = 0; j < m; j++) {  
 System.*out*.print(matrix[i][j] + " ");  
 }  
 System.*out*.println();  
 }  
 }  
}

}

## 6.3. Тестирование работы программы



# Задание 7

## 7.1. Постановка задачи

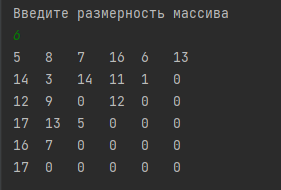
Заполнить двумерный массив n×n по образцу. Образец заполнения и пример:

|  |  |
| --- | --- |
| Образец заполнения массива | Пример |
|  |  |

## 7.2.1 Решение задачи, код программы

import java.util.Random;  
import java.util.Scanner;  
public class z7 {  
 public static void main(String[] args) {  
 Random run = new Random();  
 Scanner sc = new Scanner(System.*in*);  
 System.*out*.println("Введите размерность массива");  
 int n = sc.nextInt();  
 int g = n;  
 int mas[][] = new int [n][n];  
 while (g !=0){  
 for(int i = 0; i< mas.length; i++){  
 for( int j = 0; j<g; j++){  
 mas [i][j]= run.nextInt(20);  
 }  
 g = g-1;  
 }  
 }  
 *G*(mas);  
 }  
  
 public static void G(int[][] A){  
 for(int i=0; i<A.length; i++){  
 for(int j=0; j<A.length; j++) {  
 System.*out*.print(A[i][j] + "\t");  
 }  
 System.*out*.println();  
 }  
 System.*out*.println();  
 }  
}

## 7.3. Тестирование работы программы



# Задание 8

## 8.1. Постановка задачи

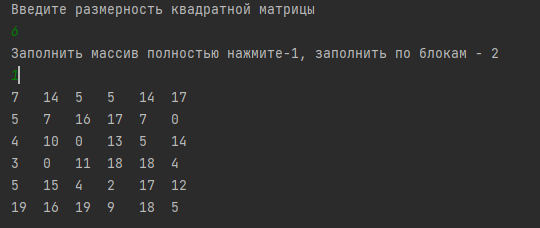
Заполнить двумерный массив n×n по образцу. Матрица заполняется автоматически для любого n. Создать метод заполнения блока матрицы и отпараметризировав заполнение заполнить при помощи метода всю матрицу. Образец заполнения:

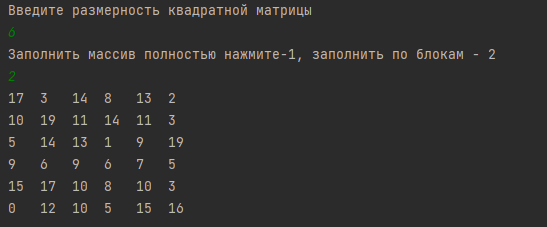


## 8.2. Решение задачи, код программы

import java.util.Random;  
import java.util.Scanner;  
public class z8 {  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
 System.*out*.println("Введите размерность квадратной матрицы");  
 int n = sc.nextInt();  
 System.*out*.println("Заполнить массив полностью нажмите-1, заполнить по блокам - 2");  
 int vibor = sc.nextInt();  
 if (vibor == 1){  
 *vibor1*(n);  
 }  
 if (vibor == 2){  
 *vibor2*(n);  
 }  
 }  
 public static void vibor1( int n){  
 int mas[][] = new int[n][n];  
 Random run = new Random();  
 for (int i = 0; i < (n) ; i++) {  
 for (int j = 0; j < (n) ; j++) {  
 mas[i][j] = run.nextInt(20);  
 }  
 }  
 *G*(mas);  
 }  
 public static void vibor2( int n){  
 int mas[][] = new int[n][n];  
 Random run = new Random();  
 for (int i = 0; i < n/2 ; i++) {  
 for (int j = 0; j < n/2 ; j++) {  
 mas[i][j] = run.nextInt(20);  
 }  
 }  
 for(int i = n/2; i<(n); i++){  
 for( int j = n/2; j<(n); j++){  
 mas [i][j]= run.nextInt(20);  
 }  
 }  
 for(int i = 0; i < n/2 ; i++){  
 for( int j = n/2; j<n; j++){  
 mas [i][j]= run.nextInt(20);  
 }  
 }  
 for(int i = n/2; i<n; i++){   
 for( int j = 0; j < n/2 ; j++){  
 mas [i][j]= run.nextInt(20);  
 }  
 }  
 *G*(mas);  
 }  
 public static void G(int[][] A){  
 for(int i=0; i<A.length; i++){  
 for(int j=0; j<A.length; j++) {  
 System.*out*.print(A[i][j] + "\t");  
 }  
 System.*out*.println();  
 }  
 System.*out*.println();  
 }  
}

## 8.3. Тестирование работы программы





# Задание 9

## 9.1. Постановка задачи

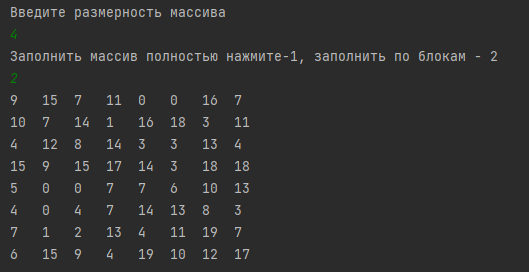
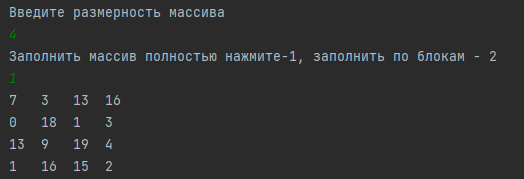
Заполнить двумерный массив 2n×2n по образцу. Матрица заполняется автоматически для любого n. Создать метод заполнения блоков матрицы и отпараметризировав выполнить заполнение всей матрицы при помощи методов. Образец заполнения:



## 9.2. Решение задачи, код программы

import java.util.Random;  
import java.util.Scanner;  
public class z9 {  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
 System.*out*.println("Введите размерность массива");  
 int n = sc.nextInt();  
 System.*out*.println("Заполнить массив полностью нажмите-1, заполнить по блокам - 2");  
 int vibor = sc.nextInt();  
 if (vibor == 1){  
 *vibor1*(n);  
 }  
 if (vibor == 2){  
 *vibor2*(n);  
 }  
 }  
 public static void vibor1( int n){  
 int mas[][] = new int[n][n];  
 Random run = new Random();  
 for (int i = 0; i < (n) ; i++) {  
 for (int j = 0; j < (n) ; j++) {  
 mas[i][j] = run.nextInt(20);  
 }  
 }  
 *G*(mas);  
 }  
 public static void vibor2(int n) {  
 int mas[][] = new int[2 \* n][2 \* n];  
 Random run = new Random();  
  
 for (int i = 0; i < n; i++) {  
 for (int j = 0; j < n; j++) {  
 mas[i][j] = run.nextInt(20);  
 }  
 }  
  
 for (int i = n; i < (2 \* n); i++) {  
 for (int j = n; j < (2 \* n); j++) {  
 mas[i][j] = run.nextInt(20);  
 }  
 }  
  
 for (int i = 0; i < n; i++) {  
 for (int j = n; j < (2 \* n); j++) {  
 mas[i][j] = run.nextInt(20);  
 }  
 }  
  
 for (int i = n; i < (2 \* n); i++) {  
 for (int j = 0; j < n; j++) {  
 mas[i][j] = run.nextInt(20);  
 }  
 }  
  
 *G*(mas);  
 }  
  
 public static void G(int[][] A) {  
 for (int i = 0; i < A.length; i++) {  
 for (int j = 0; j < A.length; j++) {  
 System.*out*.print(A[i][j] + "\t");  
 }  
 System.*out*.println();  
 }  
 System.*out*.println();  
 }  
}

## 9.3. Тестирование работы программы



# Задание 10

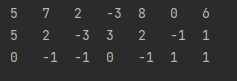
## 10.1. Постановка задачи

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

public class z10 {  
 public static void main(String[] args) {  
 int N[][] = {{5, 7, 2, -3, 8, 0, 6},  
 {5, 2, -3, 3, 2, -1, 1},  
 {0, 0, 0, 0, 0, 0, 0}};  
 for (int i = 0; i < 7; i++) {  
 int x = N[0][i];  
 int y = N[1][i];  
 if (*figura\_1\_obl*(x, y) | *figura\_2\_obl*(x, y)) {  
 N[2][i] = 1;  
  
 } else if (*figura\_1\_gr*(x, y) | *figura\_2\_gr*(x, y)) {  
 N[2][i] = 0;  
 } else {  
 N[2][i] = -1;  
 }  
 }  
 *G*(N);  
 }  
  
 public static boolean figura\_1\_obl(int x, int y) {  
  
 if ((x > 5 & y > 0 & y < (-2.5) \* x + 17.5) | (y > x - 7 & y < 0 & y > (-  
 1 / 3f) \* x + 1) | (y > 0 & y > (-x) - 3 & y < (-1 / 3f) \* x + (5 / 3f))  
 | (y > -1 & y < x - 3 & y < (-1 / 3f) \* x + 1) | (y < -1 &  
 Math.*pow*(x - 4, 2) + Math.*pow*(y + 1, 2) < 4)) {  
 return true;  
 }  
 return false;  
 }  
  
 public static boolean figura\_1\_gr(int x, int y) {  
  
 if ((x == 5 | y == 0 | y == -2.5 \* x + 17.5 | y == x - 7 | y == 0 | y == (-  
 1 / 3f) \* x + 1 | y == 0 | y == -x - 3 | y == (-1 / 3f) \* x + (5 / 3f)  
 | y == -1 | y == x - 3 | y == (-1 / 3f) \* x + 1 | Math.*pow*(x -  
 4, 2) + Math.*pow*(y + 1, 2) == 4) & (y >= -3 & y <= 5 & x >= 2 & x <= 7)) {  
 return true;  
 }  
 return false;  
 }  
  
 public static boolean figura\_2\_obl(int x, int y) {  
  
 if ((y > 3.5 \* x - 3 & y < 2 \* x & y > -x - 3) | (y > 2 \* x & y > -x - 3 & y < ((1 / 3f) \* x)  
 | (y > (1 / 3f) \* x & y < x + 2 & y < -x)  
 | (y > -x & y > 2 \* x & y < 0.5 \* x + 1.5) | (y > 0.5 \* x + 1.5 & y < -  
 0.5 \* x + 2.5 & x > -1) | (x > -1 & Math.*pow*(x + 1, 2) + Math.*pow*(y - 3, 2) < 4))) {  
 return true;  
 }  
 return false;  
 }  
  
 public static boolean figura\_2\_gr(int x, int y) {  
  
 if ((y == 3.5 \* x - 3 | y == 2 \* x | y == -x - 3 | y == 2 \* x | y == -x - 3 |  
 y == (1 / 3f) \* x | y == (1 / 3f) \* x | y == x + 2 | y == -x  
 | y == -x | y == 2 \* x | y == 0.5 \* x + 1.5 | y == 0.5 \* x + 1.5 | y == -  
 0.5 \* x + 2.5 | x == -1 | x == -1 | Math.*pow*(x + 1, 2) + Math.*pow*(y - 3, 2) == 4)  
 & (y >= -3 & y <= 5 & x >= -3 & x <= 2)) {  
 return true;  
 }  
 return false;  
 }  
  
 public static void G(int[][] A) {  
 for (int i = 0; i < 3; i++) {  
 for (int j = 0; j < 7; j++) {  
 System.*out*.print(A[i][j] + "\t");  
 }  
 System.*out*.println();  
 }  
 System.*out*.println();  
 }  
}

## 10.3. Тестирование работы программы



# Задание 11

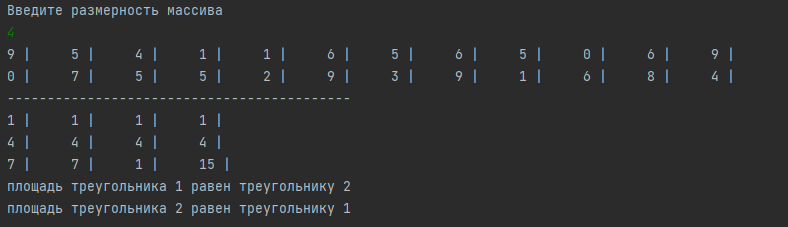
## 11.1. Постановка задачи

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 z11 {  
 public static void main(String[] args){  
 Scanner in = new Scanner(System.*in*);  
 System.*out*.println("Введите размерность массива");  
 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("площадь треугольника " + (i + 1) + " равен треугольнику " + (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



# Задание 12

## 12.1. Постановка задачи

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 z12 {  
 public static void main(String[] args) {  
 Scanner in = new Scanner(System.*in*);  
 System.*out*.println("Введите размерность массива");  
 int n = in.nextInt();  
 double[][] arr = new double[n][n];  
 double[][] arr\_1 = new double[n][n];  
  
 System.*out*.println("Введите элементы массива");  
 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("Введите элементы обратного массива");  
 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("Повторите попытку");  
 *exit*(0);  
 }  
 }  
 }  
  
 System.*out*.println("Всё получилось");  
 }  
}

## 12.3. Тестирование работы программы

