**Задание 1.** Составьте алгоритм вычисления суммы элементов матрицы, лежащих выше главной диагонали

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: ");  
 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();  
 }  
 }  
 int sum = 0;  
 for (int i = 0; i < n; i++) {  
 for (int j = i + 1; j < n; j++) {  
 sum += matrix[i][j];  
 }  
 }  
  
 // Вывод результата  
 System.*out*.println("Сумма элементов выше главной диагонали: " + sum);  
 }  
}

**Задание 2.** Составьте алгоритм, проверяющий, является ли матрица единичной.

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: ");  
 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();  
 }  
 }  
  
 boolean edinichnost = true;  
 for (int i = 0; i < n; i++) {  
 for (int j = 0; j < n; j++) {  
 if (i == j) {  
 if (matrix[i][j] != 1) {  
 edinichnost = false;  
 }  
 } else {  
 if (matrix[i][j] != 0) {  
 edinichnost = false;  
 }  
 }  
 }  
 }  
   
 if (edinichnost) {  
 System.*out*.println("Матрица является единичной.");  
 } else {  
 System.*out*.println("Матрица не является единичной.");  
 }  
 }  
}

**Задание 3.** Составьте алгоритм вывода строки матрицы, сумма значений элементов которой наибольшая.

import java.util.Scanner;  
  
public class Main {  
  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
  
 System.*out*.print("Введите количество строк: ");  
 int stroka = scanner.nextInt();  
 System.*out*.print("Введите количество столбцов: ");  
 int stolbec = scanner.nextInt();  
  
 int[][] matrix = new int[stroka][stolbec];  
  
 System.*out*.println("Введите элементы матрицы:");  
 for (int i = 0; i < stroka; i++) {  
 for (int j = 0; j < stolbec; j++) {  
 matrix[i][j] = scanner.nextInt();  
 }  
 }  
  
 int[] maxstr = *MaxSum*(matrix);  
 System.*out*.println("Строка с наибольшей суммой элементов: ");  
 for (int value : maxstr) {  
 System.*out*.print(value + " ");  
 }  
 System.*out*.println();  
 }  
  
 public static int[] MaxSum(int[][] matrix) {  
 int maxSum = Integer.*MIN\_VALUE*;  
 int maxIndex = -1;  
  
 for (int i = 0; i < matrix.length; i++) {  
 int sum = 0;  
 for (int j = 0; j < matrix[i].length; j++) {  
 sum += matrix[i][j];  
 }  
 if (sum > maxSum) {  
 maxSum = sum;  
 maxIndex = i;  
 }  
 }  
  
 return matrix[maxIndex];  
 }  
}

**Задание 4.** Составьте алгоритм вывода столбца матрицы, сумма значений элементов которого наименьшая.

import java.util.Scanner;  
  
public class Main {  
  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
   
 System.*out*.print("Введите количество строк: ");  
 int stroka = scanner.nextInt();  
 System.*out*.print("Введите количество столбцов: ");  
 int stolbec = scanner.nextInt();  
   
 int[][] matrix = new int[stroka][stolbec];  
   
 System.*out*.println("Введите элементы матрицы:");  
 for (int i = 0; i < stroka; i++) {  
 for (int j = 0; j < stolbec; j++) {  
 matrix[i][j] = scanner.nextInt();  
 }  
 }  
   
 int[] minstl = *MinSum*(matrix);  
 System.*out*.println("Столбец с наименьшей суммой элементов: ");  
 for (int value : minstl) {  
 System.*out*.print(value + " ");  
 }  
 }  
  
 public static int[] MinSum(int[][] matrix) {  
 int minsum = Integer.*MAX\_VALUE*;  
 int minIndex = -1;  
  
 for (int j = 0; j < matrix[0].length; j++) {  
 int sum = 0;  
 for (int i = 0; i < matrix.length; i++) {  
 sum += matrix[i][j];  
 }  
 if (sum < minsum) {  
 minsum = sum;  
 minIndex = j;  
 }  
 }  
  
 int[] mins = new int[matrix.length];  
 for (int i = 0; i < matrix.length; i++) {  
 mins[i] = matrix[i][minIndex];  
 }  
  
 return mins;  
 }  
}

**Задание 5.** Составьте алгоритм преобразования исходной матрицы так, чтобы первый элемент каждой строки был заменен средним арифметическим элементов этой строки.

import java.util.Scanner;  
  
public class Main {  
  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
   
 System.*out*.print("Введите количество строк: ");  
 int stroka = scanner.nextInt();  
 System.*out*.print("Введите количество столбцов: ");  
 int stolbec = scanner.nextInt();  
   
 double[][] matrix = new double[stroka][stolbec];  
   
 System.*out*.println("Введите элементы матрицы:");  
 for (int i = 0; i < stroka; i++) {  
 for (int j = 0; j < stolbec; j++) {  
 matrix[i][j] = scanner.nextDouble();  
 }  
 }  
   
 *srednee*(matrix);  
   
 System.*out*.println("Преобразованная матрица:");  
 for (double[] row : matrix) {  
 for (double value : row) {  
 System.*out*.print(value + " ");  
 }  
 System.*out*.println();  
 }  
 }  
  
 public static void srednee(double[][] matrix) {  
 for (int i = 0; i < matrix.length; i++) {  
 double sum = 0;  
 for (int j = 0; j < matrix[i].length; j++) {  
 sum += matrix[i][j];  
 }  
 double sr = sum / matrix[i].length;  
 matrix[i][0] = sr;  
 }  
 }  
}

**Задание 6.** Задана матрица А(m,n). Составить алгоритм формирования вектора P(n), в котором записаны номера строк максимальных элементов каждого столбца.

import java.util.Scanner;  
  
public class Main {  
  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
   
 System.*out*.print("Введите количество строк: ");  
 int stroka = scanner.nextInt();  
 System.*out*.print("Введите количество столбцов: ");  
 int stolbec = scanner.nextInt();  
   
 double[][] matrix = new double[stroka][stolbec];  
   
 System.*out*.println("Введите элементы матрицы:");  
 for (int i = 0; i < stroka; i++) {  
 for (int j = 0; j < stolbec; j++) {  
 matrix[i][j] = scanner.nextDouble();  
 }  
 }  
   
 int[] P = *vector*(matrix);  
   
 System.*out*.println("Вектор P(n) с номерами строк максимальных элементов каждого столбца:");  
 for (int index : P) {  
 System.*out*.print(index + " ");  
 }  
 }  
  
 public static int[] vector(double[][] matrix) {  
 int[] P = new int[matrix[0].length];  
  
 for (int j = 0; j < matrix[0].length; j++) {  
 double maxVal = Double.*NEGATIVE\_INFINITY*;  
 int maxIndex = -1;  
 for (int i = 0; i < matrix.length; i++) {  
 if (matrix[i][j] > maxVal) {  
 maxVal = matrix[i][j];  
 maxIndex = i;  
 }  
 }  
 P[j] = maxIndex;  
 }  
  
 return P;  
 }  
}