**import java.util.Scanner;**

**public class inverseMatrix{**

**public static void main (String[]args) {**

**/\* intro, enter the size of the matrix.**

**"n" will be the number of rows, and "m" the number of columns. \*/**

**System.*out*.println("What is the size of your matrix? ");**

**int n = 0;**

**int m = 0;**

**// type the number of rows "n", if you will type a "char" the program will stop running.**

**Scanner scan = new Scanner(System.*in*);**

**boolean isAnInt = scan.hasNextInt();**

**if (isAnInt) {**

**n = scan.nextInt();**

**} else {**

**System.*out*.println("This is not an Integer");**

**System.*exit*(0);**

**}**

**// type number (columns) "m", if you will type a "char" the program will stop running.**

**boolean isAnInt2 = scan.hasNextInt();**

**if (isAnInt2) {**

**m = scan.nextInt();**

**} else {**

**System.*out*.println("This is not an Integer");**

**System.*exit*(0);**

**}**

**// create the matrix, type your numbers.**

**// Your numbers will be stored raw by raw from the 1st raw.**

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

**System.*out*.println("Your matrix is " + n + " rows \* " + m + " columns, which is " + n \* m + " numbers");**

**System.*out*.println(" ");**

**System.*out*.println("Enter your numbers : ");**

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

**for (int j = 0; j < m; j++) {**

**matrix[i][j] = scan.nextInt();**

**}**

**}**

**System.*out*.println("This is your matrix " + n + " \* " + m + ":");**

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

**for (int j = 0; j < m; j++) {**

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

**}**

**System.*out*.println(" ");**

**}**

**/\*transpose matrix 2x2.**

**A new matrix "matrix2x2T" has been created to store the values of the transpose of the matrix. The original matrix created “matrix[n][m] will swap its elements (rows[j] and columns[i]) with the matrix “matrix2x2T[n][m]” ([i] rows and [j]columns) \*/**

**if( n == 2){**

**if( m == 2){**

**System.*out*.println(" ");**

**System.*out*.println("The transpose of the matrix is : ");**

**int[][] matrix2x2T = new int[n][m];**

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

**for (int j = 0; j < m; j++) {**

**matrix2x2T[i][j] = (matrix[j][i]);**

**System.*out*.print(matrix2x2T[i][j] + " ");**

**}**

**System.*out*.println(" ");**

**}**

**System.*out*.println(" ");**

**/\* Determinant for the matrix 2x2.**

**Calculations if the size of the matrix will be 2x2.**

**Do not forget that if the determinant is equals to 0 it is not possible to calculate the inverse.\*/**

**int det;**

**det = (matrix[0][0] \* matrix[1][1]) - (matrix[0][1] \* matrix[1][0]);**

**if (det == 0) {**

**System.*out*.println(" ");**

**System.*out*.println("Impossible to calculate the inverse because:");**

**}**

**System.*out*.println("The determinant of the matrix is = " + det);**

**System.*out*.println(" ");**

**/\* Inverse of matrix 2x2.**

**A new matrix "matrix2x2I", has been created to store the values of the inverse of matrix. The new matrix use the formula (1/det)\*matrix2x2T (transposed matrix).**

**Numbers have been formatted. \*/**

**double [][] matrix2x2I = new double [n][m];**

**System.*out*.println("The inverse of your matrix is : ");**

**for (int i = 0; i < 2; i++){**

**for (int j = 0; j < 2; j++){**

**matrix2x2I[i][j] = (1.0/det)\*matrix2x2T[i][j];**

**System.*out*.format(String.*format*("%.3f",matrix2x2I[i][j]) + " ");**

**}**

**System.*out*.println(" ");**

**}**

**System.*exit*(0);**

**}**

**}**

**//cofactors matrix "3x3".**

**int c11 = (matrix[1][1] \* matrix[2][2]) - (matrix[2][1] \* matrix[1][2]);**

**int c12 = - ((matrix[1][0] \* matrix[2][2]) - (matrix[2][0] \* matrix[1][2]));**

**int c13 = (matrix[1][0] \* matrix[2][1]) - (matrix[1][1] \* matrix[2][0]);**

**int c21 = - ((matrix[0][1] \* matrix[2][2]) - (matrix[0][2] \* matrix[2][1]));**

**int c22 = (matrix[0][0] \* matrix[2][2]) - (matrix[2][0] \* matrix[0][2]);**

**int c23 = - ((matrix[0][0] \* matrix[2][1]) - (matrix[2][0] \* matrix[0][1]));**

**int c31 = (matrix[0][1] \* matrix[1][2]) - (matrix[1][1] \* matrix[0][2]);**

**int c32 = - ((matrix[0][0] \* matrix[1][2]) - (matrix[1][0] \* matrix[0][2]));**

**int c33 = (matrix[0][0] \* matrix[1][1]) - (matrix[0][1] \* matrix[1][0]);**

**/\* transpose of matrix 3x3.**

**Cofactors have been assigned to a new matrix "matrixT".**

**A new matrix "matrixT2 has been created to store the values of the transpose of a matrix. The original matrix created “matrixT[n][m] will swap its elements (rows[j] and columns[i]) with the matrix “matrixT2[n][m]” ([i] rows and [j]columns)**

**\*/**

**int [][] matrixT = { {c11, c12, c13} , {c21, c22, c23} , {c31, c32, c33} };**

**int [][] matrixT2 = new int[n][m];**

**System.*out*.println(" ");**

**System.*out*.println("The transpose of the matrix is : ");**

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

**for (int j = 0; j < m; j++){**

**matrixT2[i][j] = matrixT[j][i] ;**

**System.*out*.print( (matrixT2[i][j]) + " ");**

**}**

**System.*out*.println(" ");**

**}**

**System.*out*.println(" ");**

**/\* Tom's determinant matrix 3x3 alternative**

**Calculations if the size of the matrix will be 3x3.**

**Do not forget that if the determinant is equals to 0 it is not possible to calculate the inverse**

**int a,b,c,d,e,f;**

**a = (matrix[0][0] \* matrix[1][1] \* matrix[2][2]);**

**b = (matrix[1][0] \* matrix[2][1] \* matrix[0][2]);**

**c = (matrix[2][0] \* matrix[0][1] \* matrix[1][2]);**

**d = (matrix[0][2] \* matrix[1][1] \* matrix[2][0]);**

**e = (matrix[1][2] \* matrix[2][1] \* matrix[0][0]);**

**f = (matrix[2][2] \* matrix[0][1] \* matrix[1][0]);**

**int det = (a + b + c)-(d + e + f);**

**System.out.println("The determinant of your matrix 3x3 is : " + det);**

**if(determinant == 0){**

**System.out.println("impossible to continue");**

**System.out.println(" ");**

**\*/**

**// Determinant matrix if the size of the matrix will be 3x3**

**// Do not forget that if the determinant is equals to 0 it is not possible to calculate the inverse.**

**int x,y,z;**

**x=(matrix[0][0] \* (matrix[1][1] \* matrix[2][2] - matrix[1][2] \* matrix[2][1]));**

**y=(matrix[0][1] \* (matrix[1][0] \* matrix[2][2] - matrix[1][2] \* matrix[2][0]));**

**z=(matrix[0][2] \* (matrix[1][0] \* matrix[2][1] - matrix[1][1] \* matrix[2][0]));**

**int determinant = x-y+z;**

**if(determinant == 0){**

**System.*out*.println("Impossible to calculate the inverse, because:");**

**}**

**System.*out*.println("The determinant of your matrix is " + determinant);**

**System.*out*.println(" ");**

**/\*the inverse of the matrix.**

**A new matrix has been created to invert the matrix "matrixT2".**

**The new matrix use the formula (1/det)\*matrixT2 (transposed matrix).**

**Numbers have been formatted. \*/**

**System.*out*.println("The inverse of the matrix is :");**

**double [][] inverseM = new double[n][m];**

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

**for (int j = 0; j < m; j++) {**

**inverseM[i][j] = (1.0/determinant) \* matrixT2[i][j];**

**System.*out*.format(String.*format*("%.3f ", inverseM[i][j]));**

**}**

**System.*out*.println(" ");**

**}**

**}**

**} //Giorgio Basile**