public class SparseMatrix {  
  
        */\*\*  
         \* AN - одномерный массив, который содержит все ненулевые элементы матрицы  
         \* A,перечисленные в строчном порядке.  
         \*/*  
        private double[] AN;  
  
        */\*\*  
         \* JA - одномерный массив, который содержит столько же элементов, сколько и  
         \* массив AN и для каждого из них указывает, в каком столбце находиться  
         \* данный элемент.  
         \*/*  
        private int[] JA;  
  
        */\*\*  
         \* IA - одномерный массив, элементы которого указывают на позиции, с которых  
         \* начинается описание очередной строки.  
         \*/*  
        private int[] IA;  
                  
        */\*\*  
         \* Формирование разреженной матрицы по количеству ненулевых элементов и  
         \* порядку матрицы  
         \*   
         \* @param n -  
         \*            порядок матрицы (количество строк или столбцов)  
         \* @param nz -  
         \*            количество ненулевых элементов  
         \*   
         \*/*  
        public SparseMatrix(int n, int nz) {                      
                AN = new double[nz];  
                JA = new int[nz];  
                IA = new int[n + 1];              
        }       
  
    
        public SparseMatrix mult (SparseMatrix B) {  
                SparseMatrix C = new SparseMatrix(IA.length - 1, (IA.length - 1)  
                                \* (IA.length - 1));  
                int n = 0;  
                C.IA[0] = 0;  
                for (int i = 0; i < IA.length - 1; i++) {  
  
                        for (int j = 0; j < B.IA.length - 1; j++) {  
  
                                for (int s = IA[i]; s < IA[i + 1]; s++) {  
                                        if (j == JA[s]) {  
                                                for (int k =  B.IA[j]; k < B.IA[j + 1]; k++) {  
                                                        C.AN[B.JA[k] + n] += B.AN[k]\* AN[s];  
                                                }  
                                                break;  
                                        }  
  
                                }  
  
                                C.JA[j + n] = j;  
                        }  
                        n = n + IA.length-1;  
                        C.IA[i + 1] = n;  
                }  
                return C;  
        }