**Posicao.java**

|  |
| --- |
| 1 **public class** Posicao { 2 **public int** linha; 3 **public int** coluna; 4  5 **public** String toString() { 6 **return** "linha=" + linha + " coluna=" + coluna; 7 } 8 } |

**VetorAmigo.java**

|  |
| --- |
| 01  02 **public class**VetorAmigo { 03  04    **public static void**main(String[] args) { 05       **int**[][] matriz1 = {{0,0,0},{0,0,0},{0,0,0}}; 06       **int**[][] matriz2 = {{0,0,0,0},{0,0,0,0},{0,0,0,0},{0,0,0,0}}; 07       **int**[][] matriz3 = {{0,0,0,0},{0,0,0,0},{0,0,0,0}}; 08  09       **int**[][] matriz4 = **new int**[27][]; 10       **for**(**int**i = 0; i < matriz4.length; i++) { 11          matriz4[i] = **new int**[5]; 12       } 13        14       testaMatriz(matriz1); 15       testaMatriz(matriz2); 16       testaMatriz(matriz3); 17       testaMatriz(matriz4); 18    } 19  20    **public static void**testaMatriz(**int**[][] matriz) { 21       System.out.println("matriz " + matriz.length + " x " + matriz[0].length); 22       **int**linhas = matriz.length \* matriz[0].length; 23        24       **for**(**int**j = 1; j <= linhas; j++) { 25          Posicao p = calculaPosicao( matriz, j); 26          System.out.println("numero " + j + " = posicao : " + p); 27       } 28        29       System.out.println(""); 30    } 31  32    /\*\* 33     \* calculo da linha : 34     \*  35     \* Divindo o numero pela quantidade de elementos de cada linha 36     \* obtemos um número entre a linha anterior e a desejada, com a  37     \* soma de 0.99 seguido de um truncamento (conversão para inteiro  38     \* desprezando as casas decimais) realizamos o arredontamento para cima 39     \* e diminuimos 1 para usar como posição de vetor. 40     \*   41     \* calculo da coluna 42     \*  43     \* diminuimos o numero pela linha vezes a quantidade de elementos que  44     \* existem em cada linha reduzindo o número a primeira linha, e diminuimos  45     \* de 1 para usar como posição de vetor  46     \*  47     \* @param matriz 48     \* @param i 49     \* @return Posicao 50     \*/ 51    **public static**Posicao calculaPosicao(**int**[][] matriz, **int**i) { 52       Posicao p = **new**Posicao(); 53       p.linha = (**int**)(((**double**)i / matriz[0].length)+0.99)-1;  54       p.coluna = i - ( p.linha \* matriz[0].length )-1;  55       **return**p; 56    } 57 } |