

```

1      /*
        File: T_dm.cpp

        Описать класс T_dm - двумерный массив.

        Решение. Сергей Митрофанов
    */

6      # include <iostream.h>
7      # include <conio.h>
8      # include <stdio.h>

9      // begin class T_dm

10     // двумерный массив (матрица)
11     class T_dm
12     {
13     private:
14         int str; // число строк
15         int stb; // число столбцов
16         int i, j; // индексы двумерного массива
17         int ** DM; // адрес начала двумерного массива

18     public:
19         T_dm (int, int); // конструктор инициализации
20         T_dm (); // деструктор

21         int out_str () { return str; } // возврат числа строк
22         int out_stb () { return stb; } // возврат числа столбцов

23         void record_dm (int, int, int); // запись значения в матрицу

24         void print_dm (); // печать двумерного массива
25     };

26     T_dm :: T_dm (
27         int str, //число строк
28         int stb // число столбцов
29     )
30     {
31         // конструктор двумерного массива

32         // int DM [str] [stb] - массив массивов
33         // Это массив из str элементов, каждый из которых есть
34         // массив из stb int-ов

35         T_dm :: str = str;
36         T_dm :: stb = stb;

```

```

37     // выделим память под такой двумерный массив
38     // интерпритируем такую матрицу в памяти
39     // зададим не сам двумерный массив, а его модель

40     // выделим память под str указателей на строки
41     // каждый такой указатель бует указывать на строку из stb int-ов

42     DM = new int * [str];

43     // получим для каждого указателя определенное значение адреса
44     // именно с него будет начинаться определенная строка
45     // двумерной матрицы

46     for (i = 0; i < str; ++i)
47         DM [i] = new int [stb];

48     // обнулим матрицу

49     for (i = 0; i < str; ++ i)
50         for (j = 0; j < stb; ++ j)
51             DM [i] [j] = 0;

52     cout << "Work C." << endl;

53 }

54 T_dm :: T_dm ()
55 {
56     // Деструктор

57     // Освободим память, занятую моделью матрицы
58     // сделаем это в два этапа:
59     // 1. Освободим память, занятую строками массива

60     for (i = 0; i < str ; ++ i)
61         delete [] DM [i];
62     // 2. Освободим память, занятую указателями на строки массива
63     delete [] DM;

64     cout << endl << "Work D." << endl;
65 }

66 void T_dm :: print_dm () // печать двумерного массива
67 {
68     // Метод
69     // Печать марицы

70     cout << endl;
71     for (i = 0; i < str; ++ i)
72     {

```

```

73     for (j = 0; j < stb; ++ j)
74         cout << DM [i] [j] << " ";
75     cout << endl;
76 }

77 }

78 void T_dm :: record_dm (
79     int i, // индекс строки
80     int j, // индекс столбца
81     int z // значение элемента матрицы
82 )
83 {
84     // запись значения в матрицу
85     DM [i] [j] = z;
86 }

87 // end class T_dm

88 int main ()
89 {
90     // Заполнить матрицу вот так
91     //
92     //         1 2 3 4
93     //         0 1 2 3
94     //         0 0 1 2
95     //         0 0 0 1

96     clrscr ();

97     int i, j;
98     // создадим объект
99     T_dm Matrix (7 , 7);

100    // заполним матрицу
101    for (i = 0; i < Matrix.out_str (); ++ i)
102        for (j = 0; j < Matrix.out_stb (); ++ j)
103            if (i <= j)
104                Matrix.record_dm (i, j, j - i + 1);
105    // напечатаем матрицу
106    Matrix.print_dm ();

107    // посмотреть, как сработал деструктор
108    cout << endl << "Enter, Alt + F5" << endl;

109    getch ();
110    return 0;
111 }

```

Listing данной задачи опубликован в сети Internet по адресу

<http://www.Best-Listing.ru/color-3-task-8.html>

Sergey Mitrofanov, 27.10.13, 15:05

E-mail: [infostar@mail.ru](mailto:infostar@mail.ru)

© <http://www.Best-Listing.ru/>, 2006–2013