



**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

**«МИРЭА – Российский технологический университет»**

**РТУ МИРЭА**

---

**ОТЧЕТ  
ПО ПРАКТИЧЕСКОЙ РАБОТЕ №2  
по дисциплине  
«Структуры и алгоритмы обработки данных»**

Выполнил студент группы ИКБО-13-21

Дамарад Д.В.

Принял  
старший преподаватель

Скворцова Л.А.

Практическая  
работа выполнена  
«Зачтено»

«\_\_»\_\_2021 г.

«\_\_»\_\_2021 г.

Москва 2022

## СОДЕРЖАНИЕ

<b>1</b>	<b><i>ПОСТАНОВКА ЗАДАЧИ.....</i></b>	<b><i>3</i></b>
<b>2</b>	<b><i>ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ.....</i></b>	<b><i>4</i></b>
<b>3</b>	<b><i>ВЫВОДЫ.....</i></b>	<b><i>13</i></b>

## **1 ПОСТАНОВКА ЗАДАЧИ**

Даны две матрицы, которые представлены двумерными массивами в двух вариациях: статический и динамический. Найти их произведение.

## 2 ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ

Декомпозиция:

Задачу следует разбить на следующие подзадачи:

- Заполнение матрицы.
- Вывод матрицы.
- Умножение матриц.

Операция умножения двух матриц выполняема только в том случае, если число столбцов в первом сомножителе равно числу строк во втором. ROWS и COLS – константы, отвечающие за максимальное количество строк и столбцов в матрице, представленной в виде двумерного статического массива.

Определение функций (на примере статического двумерного массива):

### 1. Заполнение матрицы.

Предусловие: int matrix[][COLS] – матрица, которая будет заполняться одним из двух способов, int n – количество строк в матрице, int m – количество столбцов в матрице.

Постусловие: заполненная матрица размера nxm.

void fillmatrix(int matrix[][COLS], int n, int m).

### 2. Вывод матрицы.

Предусловие: int matrix[][COLS] – матрица, которая будет выведена на экран, int n – количество строк в матрице, int m – количество столбцов в матрице.

Постусловие: выведенная матрица размера nxm.

void showmatrix(int matrix[][COLS], int n, int m).

### 3. Умножение матриц.

Предусловие: int matrix1[][COLS] – матрица 1, int matrix2[][COLS] – матрица 2, int n – количество строк в первой матрице, int m – количество столбцов во второй матрице.

Постусловие: результат умножения двух матриц – новая матрица размера

pxm.

Разработка алгоритмов функций и представление их на псевдокоде:

1. Заполнение матрицы.

```
void fillmatrix(int matrix[][COLS], int n, int m) {
    int y;
    cout << "Как вы желаете заполнить матрицу?" << endl << "1 - Датчиком
случайных чисел" << endl << "2 - С клавиатуры" << endl;
    y ← input;
    switch (y) {
    case 1: {
        for для всех элементов строк матрицы{
            for для всех элементов столбцов матрицы{
                matrix[i][j] ← rand() % 100;
            }
        }
        выход из цикла;
    }
    case 2: {
        for для всех элементов строк матрицы {
            for для всех элементов столбцов матрицы {
                cout << "Введите " << i + 1 << j + 1 << " элемент
матрицы: ";
                matrix[i][j] ← input;
            }
        }
        выход из цикла;
    }
    }
}
```

2. Вывод матрицы.

```
void showmatrix(int matrix[][COLS], int n, int m) {
    for для всех элементов строк матрицы {
        for для всех элементов столбцов матрицы {
            cout << matrix[i][j] << ' ';
        }
        cout << переход на след строку;
    }
}
```

### 3. Умножение матриц.

```
void matrix_multiplication(int matrix1[][COLS], int matrix2[][COLS],int n, int m)
{
    int newmatrix[ROWS][COLS];
    for для всех элементов строк матрицы
    {
        for для всех элементов столбцов матрицы
        {
            newmatrix[i][j] ← 0;
            for (int k ← 0; k < m; k++)
                newmatrix[i][j] += matrix1[i][k] * matrix2[k][j];
        }
    }
    showmatrix(newmatrix, n, m);
}
```

Тесты функций:

1) fillmatrix

Номер теста	Исходные данные	Ожидаемый результат	Результат программы	Тест пройден/не пройден
1	n=2 m=2 y=2 2 3 4 5	a11 = 2 a12 = 3 a21 = 4 a22 = 5	a11 = 2 a12 = 3 a21 = 4 a22 = 5	+
2	n=1 m=2 y=2 6 7	a11 = 6 a12 = 7	a11 = 6 a12 = 7	+

2) showmatrix

Номер теста	Исходные данные	Ожидаемый результат	Результат программы	Тест пройден/не пройден
1	a11 = 2 a12 = 3 a21 = 4 a22 = 5	2 3 4 5	2 3 4 5	+
2	a11 = 6 a12 = 7	6 7	6 7	+

3) matrix\_multiplication

Номер теста	Исходные данные	Ожидаемый результат	Результат программы	Тест пройден/не пройден
1	Матрица 1: 2 3 4 5 Матрица 2: 6 7	12 24	12 24	+

Номер теста	Исходные данные	Ожидаемый результат	Результат программы	Тест пройден/не пройден
2	Матрица 1: 2 2 2 2 Матрица 2: 2 2 2 2	8 8 8 8	8 8 8 8	+



Стоит отметить, что все функции для реализации задачи на двумерном динамическом массиве схожи, изменены параметры функций.

## Полный код программы

### matrix\_multiplication(static).cpp

```
#include <iostream>
const int ROWS = 3;
const int COLS = 3;
using namespace std;
void fillmatrix(int matrix[][COLS], int n, int m) {
    int y;
    cout << "Как вы желаете заполнить матрицу?" << endl << "1 - Датчиком случайных чисел" << endl << "2 - С клавиатуры" << endl;
    cin >> y;
    switch (y) {
        case 1: {
            for (int i = 0; i < n; i++) {
                for (int j = 0; j < m; j++) {
                    matrix[i][j] = rand() % 100;
                }
            }
            break;
        }
        case 2: {
            for (int i = 0; i < n; i++) {
                for (int j = 0; j < m; j++) {
                    cout << "Введите " << i + 1 << j + 1 << " элемент матрицы: ";
                    cin >> matrix[i][j];
                }
            }
            break;
        }
    }
}
void showmatrix(int matrix[][COLS], int n, int m) {
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            cout << matrix[i][j] << ' ';
        }
        cout << '\n';
    }
}
void matrix_multiplication(int matrix1[][COLS], int matrix2[][COLS], int n, int m) {
    int newmatrix[ROWS][COLS];
    for (int i = 0; i < n; i++) {
        {
            for (int j = 0; j < m; j++) {
                {
                    newmatrix[i][j] = 0;
                    for (int k = 0; k < m; k++)
                        newmatrix[i][j] += matrix1[i][k] * matrix2[k][j];
                }
            }
        }
        showmatrix(newmatrix, n, m);
    }
}
int main()
{
    setlocale(LC_ALL, "rus");
    int matrix1[ROWS][COLS];
```

```

int matrix2[ROWS][COLS];
int n1, m1, n2, m2;
cout << "Операция умножения двух матриц выполнима только в том случае, если число
столбцов в первом сомножителе равно числу строк во втором!" << endl;
cout << "Введите количество строк в матрице 1 (максимальное значение = 3): ";
cin >> n1;
cout << "Введите количество столбцов в матрице 1 (максимальное значение = 3): ";
cin >> m1;
if (n1 <= 0 or m1 <= 0 or n1>ROWS or m1>COLS){
    cout << "Количество строк и столбцов > 0 и <= 3";
    return 1;
}
cout << "Заполните матрицу 1" << endl;
fillmatrix(matrix1, n1, m1);
cout << "Матрица 1: " << endl;
showmatrix(matrix1, n1, m1);
cout << "Введите количество строк в матрице 2 (максимальное значение = 3): ";
cin >> n2;
cout << "Введите количество столбцов в матрице 2 (максимальное значение = 3): ";
cin >> m2;
if (n2 <= 0 or m2 <= 0 or n2 > ROWS or m2 > COLS) {
    cout << "Количество строк и столбцов > 0 и <= 3";
    return 1;
}
cout << "Заполните матрицу 2" << endl;
fillmatrix(matrix2, n2, m2);
cout << "Матрица 2: " << endl;
showmatrix(matrix2, n2, m2);
if (m1 != n2)
{
    cout << "Умножение невозможно, потому что число столбцов в первом сомножителе
не равно числу строк во втором!";
    return 0;
}
else {
    cout << "Результат произведения матрицы 1 на матрицу 2: " << endl;
    matrix_multiplication(matrix1, matrix2, n1, m2);
}
return 0;
}

```

## matrix\_multiplication(static).cpp

```

#include <iostream>
using namespace std;
void fillmatrix(int **matrix, int n, int m) {
    int y;
    cout << "Как вы желаете заполнить матрицу?" << endl << "1 - Датчиком случайных
чисел" << endl << "2 - С клавиатуры" << endl;
    cin >> y;
    switch (y) {
    case 1: {
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < m; j++) {
                matrix[i][j] = rand() % 100;
            }
        }
        break;
    }
    case 2: {
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < m; j++) {
                cout << "Введите " << i + 1 << j + 1 << " элемент матрицы: ";

```

```

        cin >> matrix[i][j];
    }
    }
    break;
}
}

void showmatrix(int **matrix, int n, int m) {
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            cout << matrix[i][j] << ' ';
        }
        cout << '\n';
    }
}

void matrix_multiplication(int **matrix1, int **matrix2, int n, int m) {
    int** newmatrix=new int*[n];
    for (int i = 0; i < n; i++) {
        newmatrix[i] = new int[m];
    }
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++)
        {
            newmatrix[i][j] = 0;
            for (int k = 0; k < m; k++)
                newmatrix[i][j] += matrix1[i][k] * matrix2[k][j];
        }
    }
    showmatrix(newmatrix, n, m);
}

int main()
{
    setlocale(LC_ALL, "rus");
    int n1, m1, n2, m2;
    cout << "Операция умножения двух матриц выполняема только в том случае, если число
столбцов в первом сомножителе равно числу строк во втором!" << endl;
    cout << "Введите количество строк в матрице 1: ";
    cin >> n1;
    cout << "Введите количество столбцов в матрице 1: ";
    cin >> m1;
    if (n1 <= 0 or m1 <= 0) {
        cout << "Количество строк и столбцов > 0";
        return 1;
    }
    int** matrix1 = new int*[n1];
    for (int i = 0; i < n1; i++) {
        matrix1[i] = new int[m1];
    }
    fillmatrix(matrix1, n1, m1);
    cout << "Матрица 1: " << endl;
    showmatrix(matrix1, n1, m1);
    cout << "Введите количество строк в матрице 2: ";
    cin >> n2;
    cout << "Введите количество столбцов в матрице 2: ";
    cin >> m2;
    if (n2 <= 0 or m2 <= 0) {
        cout << "Количество строк и столбцов > 0";
        return 1;
    }
    int** matrix2 = new int*[n2];
    for (int i = 0; i < n2; i++) {
        matrix2[i] = new int[m2];
    }
}

```

```

fillmatrix(matrix2, n2, m2);
cout << "Матрица 2: " << endl;
showmatrix(matrix2, n2, m2);
if (m1 != n2){
    cout << "Умножение невозможно, потому что число столбцов в первом сомножителе
не равно числу строк во втором!";
    return 0;
}
else {
    cout << "Результат произведения матрицы 1 на матрицу 2: " << endl;
    matrix_multiplication(matrix1, matrix2, n1, m2);
}
return 0;

```

### **3 ВЫВОДЫ**

В ходе работы на языке C++ было разработано приложение, реализующее задачу в соответствии с персональным вариантом в 2 вариациях: на двумерном статическом массиве и двумерном статическом массиве.