



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ _____ «Информатика и системы управления»

КАФЕДРА _____ «Теоретическая информатика и компьютерные технологии»

Лабораторная работа № 8
по курсу «Языки и методы программирования»
«Разработка шаблона класса»
Вариант 7

Студент группы ИУ9-21Б Шиятов Н.

Преподаватель Посевин Д. П.

Москва 2023

1 Задание

Необходимо составить шаблон класса, разместив его в отдельном заголовочном файле. Проверку работоспособности класса требуется организовать в функции `main`, размещённой в файле «`main.cpp`».

Класс: `Matrix<M,N>` – целочисленная матрица размера $M \times N$ с операцией, возвращающей ссылку на указанный элемент. Если $M = N$, то для матрицы должна быть доступна операция возведения в квадрат.

2 Результаты

Исходный код программы представлен в листингах 1– 2.

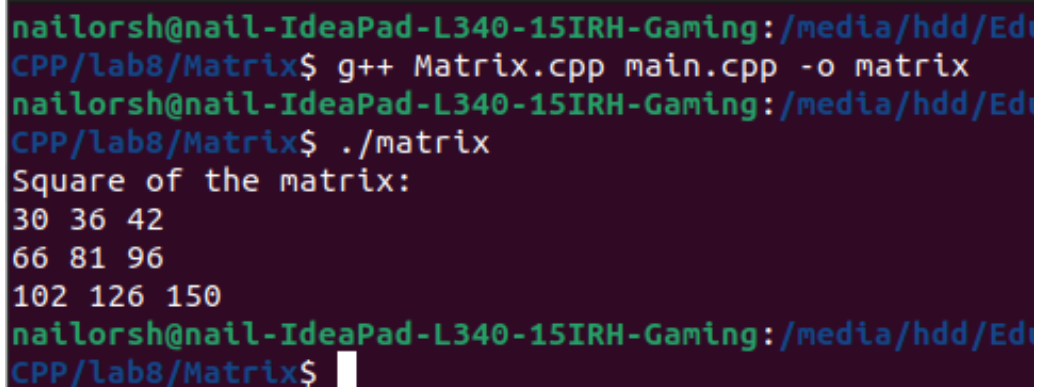
Результат запуска представлен на рисунке 1.

Листинг 1 — Класс Matrix

```
1 #include <iostream>
2 using namespace std;
3
4
5 template <int M, int N>
6 class Matrix {
7 private:
8     int matrix[M][N];
9
10 public:
11     Matrix() {
12         for (int i = 0; i < M; ++i) {
13             for (int j = 0; j < N; ++j) {
14                 matrix[i][j] = 0;
15             }
16         }
17     }
18
19     int* operator [] (int index) {
20         return matrix[index];
21     }
22
23     void print() {
24         for (int i = 0; i < M; ++i) {
25             for (int j = 0; j < N; ++j) {
26                 cout << matrix[i][j] << " ";
27             }
28             cout << '\n';
29         }
30     }
31
32     template <int K>
33     Matrix<M,K> operator*(Matrix<N,K> other) {
34         Matrix<M,K> result;
35         for (int i = 0; i < M; ++i) {
36             for (int j = 0; j < K; ++j) {
37                 for (int k = 0; k < N; ++k) {
38                     result[i][j] += matrix[i][k] * other[k][j];
39                 }
40             }
41         }
42         return result;
43     }
44
45     void square() {
46         *this = operator*(*this);
47     }
48 };
```

Листинг 2 — Проверка работоспособности

```
1 #include "Matrix.cpp"
2
3 #include <iostream>
4 using namespace std;
5
6 int main() {
7     const int M = 3;
8     const int N = 3;
9
10    Matrix<M,N> matrix1;
11    matrix1[0][0] = 1;
12    matrix1[0][1] = 2;
13    matrix1[0][2] = 3;
14    matrix1[1][0] = 4;
15    matrix1[1][1] = 5;
16    matrix1[1][2] = 6;
17    matrix1[2][0] = 7;
18    matrix1[2][1] = 8;
19    matrix1[2][2] = 9;
20
21    if (M == N) {
22        cout << "Square of the matrix:\n";
23        matrix1.square();
24    } else {
25        cout << "The square() function is not available\n";
26        return 0;
27    }
28
29    matrix1.print();
30
31    return 0;
32 }
```



```
nailorsh@nail-IdeaPad-L340-15IRH-Gaming:/media/hdd/Ed
CPP/lab8/Matrix$ g++ Matrix.cpp main.cpp -o matrix
nailorsh@nail-IdeaPad-L340-15IRH-Gaming:/media/hdd/Ed
CPP/lab8/Matrix$ ./matrix
Square of the matrix:
30 36 42
66 81 96
102 126 150
nailorsh@nail-IdeaPad-L340-15IRH-Gaming:/media/hdd/Ed
CPP/lab8/Matrix$
```

Рис. 1 — Результат