



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 16»

Студент группы ИУ9-21Б: Пенкин А. Д.

Преподаватель: Посевин Д. П.

Москва 2023

1 Цель

Целью данной работы является изучение шаблонов классов языка C++.

2 Условие

$\text{Matrix}<T,N>$ – квадратная матрица размера N , элементы которой имеют тип T . Матрица должна иметь следующие операции: 1. запись значения в элемент с индексами (i, j) ; 2. чтение значения из элемента с индексами (i, j) ; 3. построение новой матрицы путём удаления i -той строки и j -того столбца. В $\text{Matrix}<\text{bool},N>$ при $N = 8$ матрица должна быть представлена 64-разрядным целым числом, в котором каждому элементу соответствует один бит.

Последовательность символов ASCII с операциями: 1. получение количества символов; 2. получение ссылки на i -тый символ; 3. вставка нового символа в i -тую позицию последовательности; 4. проверка, является ли последовательность палиндромом.

3 Код решения

1. lab8.cpp

```
#include <iostream>
#include "Matrix.h"
#include <math.h>
#include <string>
#include <vector>

int main()
{
    std::cout << "define new int matrix with dim 10:\n";
    Matrix<int, 10>* a = new Matrix<int, 10>(0);
    a->printMatrix();

    std::cout << "\n\n set some values in matrix: \n";

    a->setValue(1, 3, 12);
    a->setValue(2, 4, 1);
    a->setValue(4, 0, 3);
    a->setValue(8, 5, -4);
    a->setValue(5, 9, 55);
```

```

a->printMatrix();

std::cout << "\n\n element from position(1, 3): " << a->getValue(1, 3);

Matrix<int, 9>* b = a->minor(1, 1);

std::cout << "\n\n minor from (1, 1):\n";
b->printMatrix();

std::cout << "\n\n string matrix:\n";
Matrix<std::string, 8>* c = new Matrix<std::string, 8>("WOW");
c->printMatrix();
}

```

2. Matrix.h

```

#pragma once
#include <iostream>
#include <vector>
#include <math.h>
#include <string>

template <typename T, int N>
class Matrix
{
private:
    std::vector<std::vector<T>>> a;
    long long a1;
public:
    Matrix(T x);
    bool isBoolSmall;
    void print();
    void printMatrix();
    void setValue(int a, int b, T v);
    T getValue(int a, int b);
    Matrix<T, N - 1>* minor(int a, int b);
};

template <typename T, int N>

```

```

void Matrix<T, N>::print()
{
    std::cout << N << "\n";
}

```

```

template <typename T, int N>
Matrix<T, N>::Matrix(T x) {
    this->isBoolSmall = false;
    this->a1 = 0;
    this->a.resize(N);
    for (int i = 0; i < N; i++) {
        this->a[i].resize(N);
        for (int j = 0; j < N; j++) {
            this->a[i][j] = x;
        }
    }
}

```

```

template <typename T, int N>
void Matrix<T, N>::printMatrix() {
    if (this->isBoolSmall) {
        for (int i = 0; i < N; i++) {
            for (int j = 0; j < N; j++) {
                std::cout << (long long)(a1 / pow(2, (8 * i + j))) % 2 << " ";
            }
            std::cout << "\n";
        }
    }
    else {
        for (int i = 0; i < N; i++) {
            for (int j = 0; j < N; j++) {
                std::cout << a[i][j] << " ";
            }
            std::cout << "\n";
        }
    }
}

```

```

template <typename T, int N>
void Matrix<T, N>::setValue(int a, int b, T v) {
    if (this->isBoolSmall) {
        a1 |= (long long)pow(2, 8 * a + b);
        if (!v) {
            a1 /= (long long)pow(2, 8 * a + b);
        }
    }
    else {
        this->a[a][b] = v;
    }
}

```

```

template <typename T, int N>
T Matrix<T, N>::getValue(int a, int b) {
    if (this->isBoolSmall) {
        return (long long)(a1 / pow(2, (8 * a + b))) % 2 == 1;
    }
    else {
        return this->a[a][b];
    }
}

```

```

template <typename T, int N>
Matrix<T, N - 1>* Matrix<T, N>::minor(int x, int y) {
    Matrix<T, N - 1>* b = new Matrix<T, N - 1>(this->a[0][0]);
    if (b->isBoolSmall) {
        int m = 0, l = 0;
        for (int i = 0; i < N; i++) {
            if (i == x) { continue; }
            else { m++; }
            for (int j = 0; j < N; j++) {
                a1 *= (long long)pow(2, 8 * m + l);
            }
            std::cout << "\n";
        }
    }
}

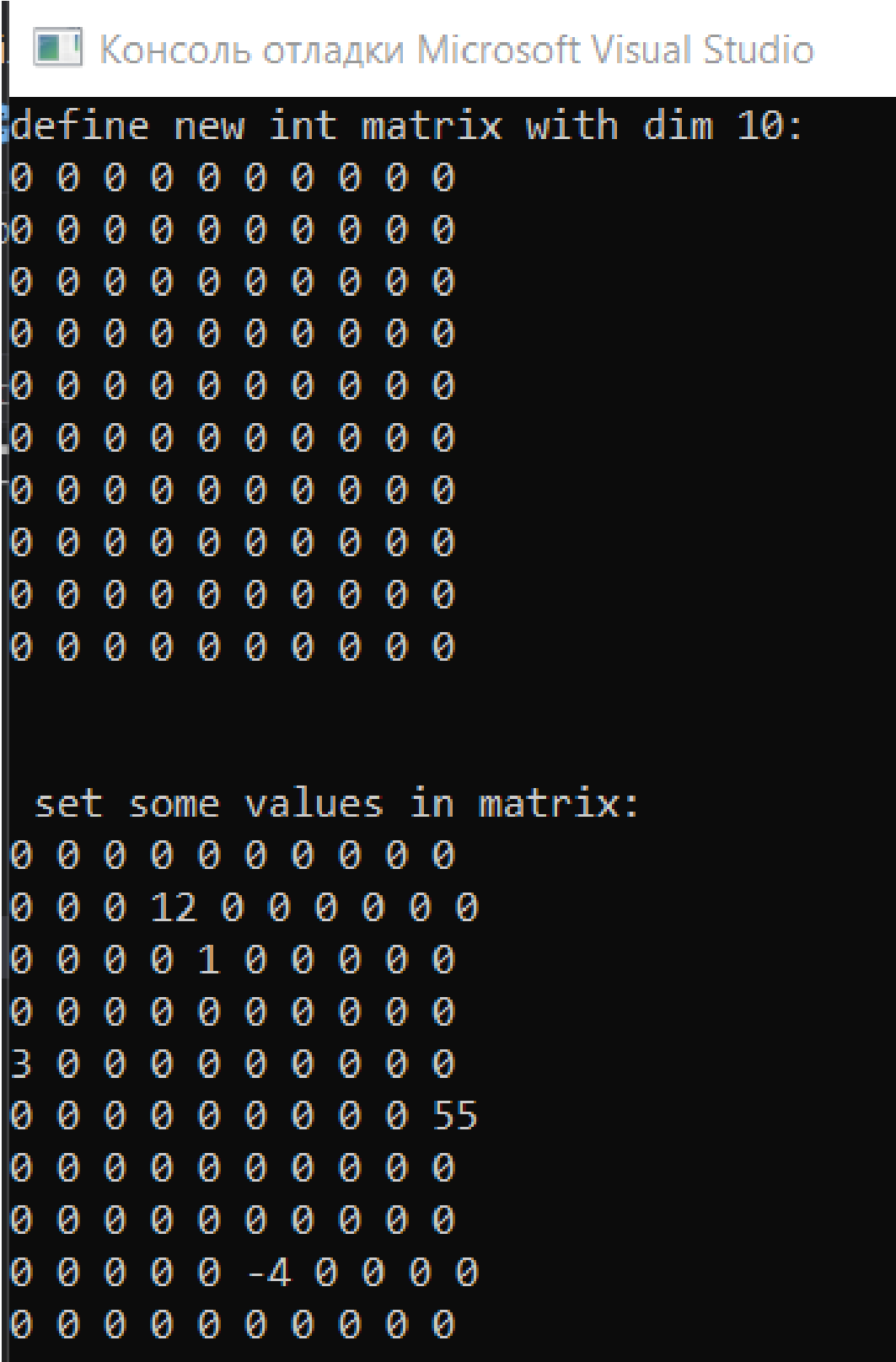
```

```

        }
    }
    else {
        int m = 0, l = 0;
        for (int i = 0; i < N; i++) {
            if (i == x) { continue; }
            for (int j = 0; j < N; j++) {
                if (j == y) { continue; }
                b->setValue(m, l, a[i][j]);
                l++;
            }
            m++;
            l = 0;
        }
    }
    return b;
}

```

4 Результаты работы программы



Консоль отладки Microsoft Visual Studio

```
define new int matrix with dim 10:  
0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0  
  
set some values in matrix:  
0 0 0 0 0 0 0 0 0 0  
0 0 0 12 0 0 0 0 0 0  
0 0 0 0 1 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0  
3 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 55  
0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 -4 0 0 0 0  
0 0 0 0 0 0 0 0 0 0
```

Рис. 1 — создание и изменение матрицы

```

element from position(1, 3): 12

minor from (1, 1):
0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0 0
0 0 0 0 0 0 0 0 0
3 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 55
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 -4 0 0 0 0
0 0 0 0 0 0 0 0 0

```

Рис. 2 — взятие элемента и минора

```

string matrix:
WOW WOW WOW WOW WOW WOW WOW WOW
WOW WOW WOW WOW WOW WOW WOW WOW
WOW WOW WOW WOW WOW WOW WOW WOW
WOW WOW WOW WOW WOW WOW WOW WOW
WOW WOW WOW WOW WOW WOW WOW WOW
WOW WOW WOW WOW WOW WOW WOW WOW
WOW WOW WOW WOW WOW WOW WOW WOW
WOW WOW WOW WOW WOW WOW WOW WOW

C:\Users\User\source\repos\lab8\Debug

```

Рис. 3 — матрица строк