МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования
«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»
РУТ(МИИТ)

Кафедра «Вычислительные системы, сети и информационная безопасность»

ОТЧЕТ ПО ДИСЦИПЛИНЕ

«АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ»

ЛАБОРАТОРНАЯ РАБОТА №2

Направление: 10.03.01 *Информационная безопасность*

Профиль: *Безопасность компьютерных систем*

Выполнил:
студент группы УИБ-115
Шевченко Димитрий

Проверил:

_____
(должность, ФИО)

_____
(должность, ФИО)

Москва 2022 г.

Задание №2

Задание: Задана матрица целых чисел. Сформировать новую матрицу, поменяв местами строки и столбцы исходной.

1 Таблица имён:

Функция main:

Исходные данные		
rows	Числовой	Количество строк
cols	Числовой	Количество столбцов
Рабочие переменные		
matrix	Числовой	Введённая матрица
Результат		
new_matrix	Числовой	Транспонированная матрица

Функция create_matrix:

Входные данные		
rows	Числовой	Количество строк
cols	Числовой	Количество столбцов
Рабочие переменные		
i	Числовой	Счётчик цикла
Результат		
matrix	Числовой	Созданная матрица

Функция input_matrix_elements:

Входные данные		
rows	Числовой	Количество строк
cols	Числовой	Количество столбцов
matrix	Числовой	Матрица для ввода
Рабочие переменные		
i	Числовой	Счётчик цикла
j	Числовой	Счётчик цикла

Функция output_matrix:

Входные данные		
rows	Числовой	Количество строк
cols	Числовой	Количество столбцов
matrix	Числовой	Матрица для вывода
Рабочие переменные		
i	Числовой	Счётчик цикла
j	Числовой	Счётчик цикла

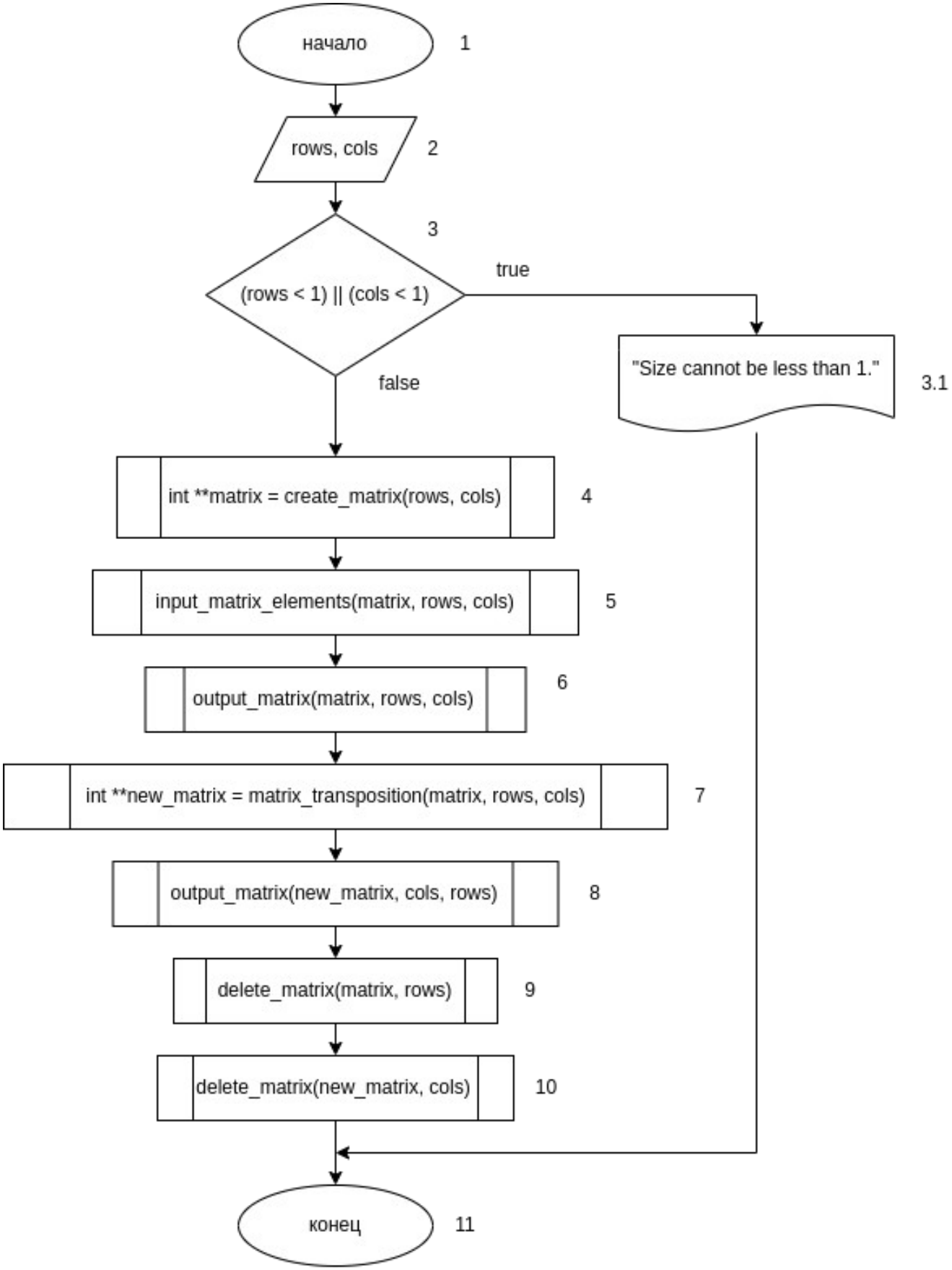
Функция matrix_transposition:

Входные данные		
rows	Числовой	Количество строк
cols	Числовой	Количество столбцов
matrix	Числовой	Изначальная матрица
Рабочие переменные		
i	Числовой	Счётчик цикла
j	Числовой	Счётчик цикла
Результат		
new_matrix	Числовой	Транспонированная матрица

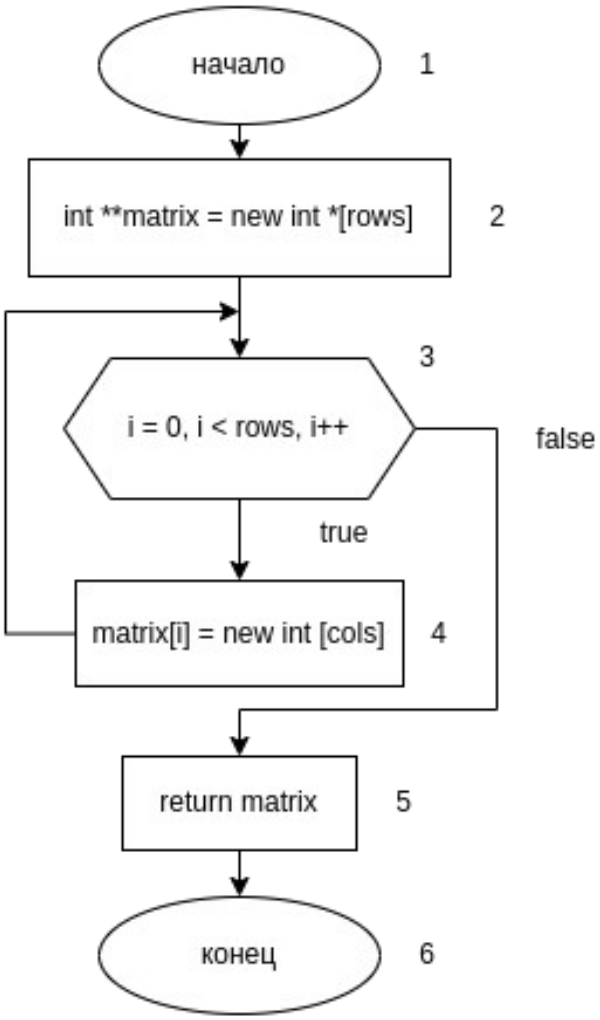
Функция delete_matrix:

Входные данные		
iterations	Числовой	Количество итераций цикла
matrix	Числовой	Матрица для удаления
Рабочие переменные		
i	Числовой	Счётчик цикла
j	Числовой	Счётчик цикла

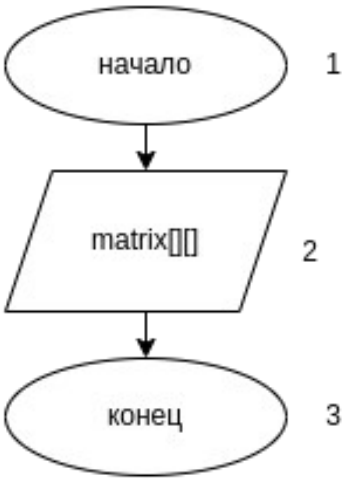
2 Блок-схема:


начало
1
rows, cols
2
3
(rows < 1) || (cols < 1)
true
"Size cannot be less than 1."
3.1
false
int **matrix = create_matrix(rows, cols)
4
input_matrix_elements(matrix, rows, cols)
5
output_matrix(matrix, rows, cols)
6
int **new_matrix = matrix_transposition(matrix, rows, cols)
7
output_matrix(new_matrix, cols, rows)
8
delete_matrix(matrix, rows)
9
delete_matrix(new_matrix, cols)
10
конец
11

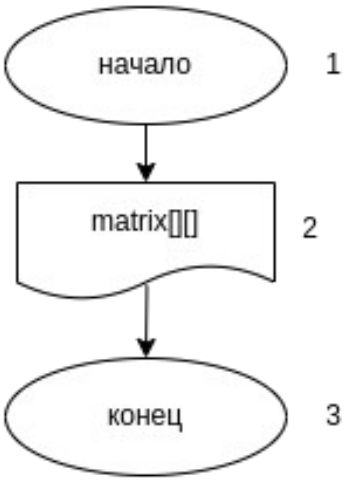
int create_matrix(int rows, int cols)**



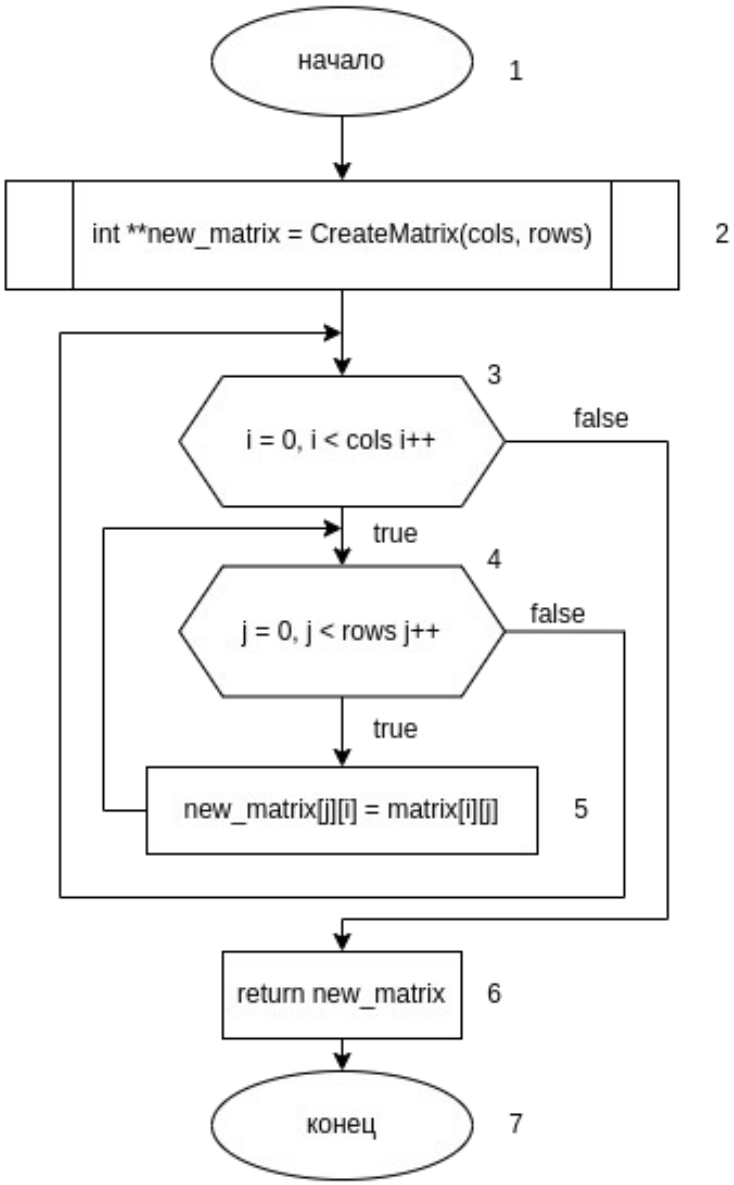
void input_matrix_elements(int **matrix, int rows, int cols)



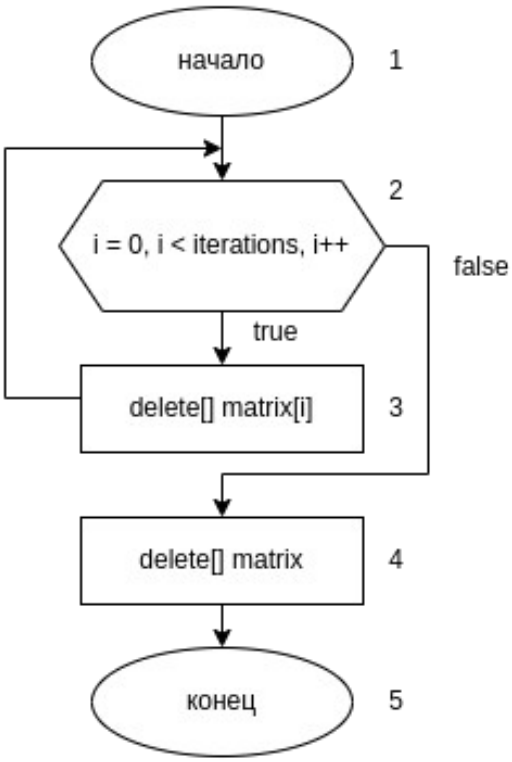
void output_matrix(int **matrix, int rows, int cols)



int matrix_transposition(int **matrix, int rows, int cols)**



void delete_matrix(int **matrix, int iterations)



3 Отладочные примеры:

Вариант I

1. Начало
2.
Ввод rows = 0
Ввод cols = -5
3. Так как 0 и -5 меньше 1, то
    3.1. Вывод «Size cannot be less than 1.»
11. Конец

Вариант II

1. Начало
2.
Ввод rows = 3
Ввод cols = 3
3. Так как 3 и 3 больше 1, то
4. int **matrix = create_matrix(3, 3)

    1. Начало
    2. int **matrix = new int *[3]
    3. Цикл от 0 до 3 с шагом 1 и переменной i
        4. matrix[i] = new int [3]
    5. return matrix
    6. Конец

5. Ввод элементов matrix
1
2
3

4
5
6

7
8
9

6. Вывод matrix
1 2 3
4 5 6
7 8 9

7. int **new_matrix = matrix_transposition(matrix, 3, 3)

    1. Начало
    2. int **new_matrix = create_matrix(3, 3)
    3. Цикл от 0 до 3 с шагом 1 и переменной i
        4. Цикл от 0 до 3 с шагом 1 и переменной j
            5. new_matrix[j][i] = matrix[i][j]
    6. return new_matrix
    7. Конец

8. Вывод new_matrix

1	4	7
2	5	8
3	6	9

9. delete_matrix(matrix, 3)

    1. Начало
    2. Цикл от 0 до 3 с шагом 1 и переменной i
        3. delete[] matrix[i]
    4. delete[] matrix
    5. Конец

10. delete_matrix(new_matrix, 3)
11. Конец

<u>Вариант III</u>

1. Начало
2.
Ввод rows = 4
Ввод cols = 2
3. Так как 4 и 2 больше 1, то
4. int **matrix = create_matrix(4, 2)

    1. Начало
    2. int **matrix = new int *[4]
    3. Цикл от 0 до 4 с шагом 1 и переменной i
        4. matrix[i] = new int [2]
    5. return matrix
    6. Конец

5. Ввод элементов matrix
10
5

21
652

45
99

106
1007

6. Вывод matrix
10 5
21 652
45 99
106 1007

7. int **new_matrix = matrix_transposition(matrix, 4, 2)

1. Начало
2. int **new_matrix = create_matrix(2, 4)
3. Цикл от 0 до 4 с шагом 1 и переменной i
    4. Цикл от 2 до 3 с шагом 1 и переменной j
        5. new_matrix[j][i] = matrix[i][j]
6. return new_matrix
7. Конец

8. Вывод new_matrix
10 21 45 106
5 652 99 1007

9. delete_matrix(matrix, 4)

1. Начало
2. Цикл от 0 до 4 с шагом 1 и переменной i
    3. delete[] matrix[i]
4. delete[] matrix
5. Конец

10. delete_matrix(new_matrix, 2)
11. Конец

4 Код программы:

```
#include <iostream> // подключение библиотеки функции ввода-вывода
using namespace std; // подключение пространства имён std

int** create_matrix(int rows, int cols){ // создание матрицы
	int **matrix = new int *[rows]; // создание массива matrix с размером rows
	for (int i = 0; i < rows; i++){ // цикл от 0 до rows
		matrix[i] = new int [cols]; // создание массива с размером cols в текущем элементе
	}
	return matrix; // возвращение созданной матрицы
}

void input_matrix_elements(int **matrix, int rows, int cols){ // ввод элементов матрицы
	cout << endl << "Input elements of matrix: " << endl;
	for (int i = 0; i < rows; i++){
		for (int j = 0; j < cols; j++){
			cin >> matrix[i][j];
		}
		cout << endl;
	}
}
```

```
void output_matrix(int **matrix, int rows, int cols){ // вывод элементов матрицы
	for (int i = 0; i < rows; i++){
		for (int j = 0; j < cols; j++){
			cout << matrix[i][j] << "\t";
		}
		cout << endl;
	}
}

int** matrix_transposition(int **matrix, int rows, int cols){ // транспонирование матрицы
	int **new_matrix = create_matrix(cols, rows); // создание матрицы new_matrix с размером cols на rows
	for (int i = 0; i < rows; i++){ // цикл от 0 до rows
		for (int j = 0; j < cols; j++){ // цикл от 0 до cols
			new_matrix[j][i] = matrix[i][j]; // присваиваем элементу new_matrix под индексами [j][i] значение matrix под индексами [i][j]
		}
	}
	return new_matrix; // возвращаем транспонированную матрицу
}

void delete_matrix(int **matrix, int iterations){ // удаление матрицы
	for (int i = 0; i < iterations; i++){ // цикл от 0 до iterations
		delete[] matrix[i]; // удаление текущего элемента матрицы
	}
	delete[] matrix; // удаление матрицы
}
```

```
int main(){
	int rows, cols; // строки и столбцы матрицы

	// ввод размера матрицы
	cout << "Input rows of matrix: " << endl;
	cin >> rows;

	cout << "Input cols of matrix: " << endl;
	cin >> cols;

	// если размер меньше 0, то завершаем программу
	if ((rows < 1) || (cols < 1)){
		cout << "Size cannot be less than 1." << endl;
		return 0;
	}

	// инициализация матрицы с указанными размерами
	int **matrix = create_matrix(rows, cols);

	// ввод матрицы
	input_matrix_elements(matrix, rows, cols);

	// вывод матрицы
	cout << endl << "Entered matrix: " << endl;
	output_matrix(matrix, rows, cols);

	// Транспонирование введённой матрицы
	int **new_matrix = matrix_transposition(matrix, rows, cols);
```

```
    // вывод полученной матрицы
    cout << endl << "Transposed matrix: " << endl;
    output_matrix(new_matrix, cols, rows);

    // очистка ОЗУ
    delete_matrix(matrix, rows);
    delete_matrix(new_matrix, cols);

    // завершение программы
    return 0;
}
```

5 Результат выполнения работы программы:

1)

```
Input rows of matrix:
0
Input cols of matrix:
-5
Size cannot be less than 1.
```

2)

```
Input rows of matrix:
3
Input cols of matrix:
3

Input elements of matrix:
1
2
3

4
5
6

7
8
9


Entered matrix:
1       2       3
4       5       6
7       8       9

Transposed matrix:
1       4       7
2       5       8
3       6       9
```

3)

```
Input rows of matrix:
4
Input cols of matrix:
2

Input elements of matrix:
10
5

21
652

45
99

106
1007


Entered matrix:
10      5
21      652
45      99
106     1007

Transposed matrix:
10      21      45      106
5       652     99      1007
```

6 Вывод:

В ходе выполнения работы были изучены функции, а также методы их использования.

Была проделана работа по созданию простейшего UI, вводу и выводу данных, построение отдельных элементов программы и защите блок-схем с программным кодом.

На контрольных примерах мы убедились, что программа работает корректно и отвечает заданным в ТЗ требованиям.

Был оформлен комплект документации на программный код.