**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 №3**

*Направление:* 10.03.01*Информационная безопасность*

*Профиль:**Безопасность компьютерных систем*

Выполнил:  
студент группы УИБ-112

Орлов Андрей Витальевич

Проверил:

Старший преподаватель Никольская Марина Николаевна

(должность, ФИО)

Старший преподаватель Цыганова Наталия Алексеевна

(должность, ФИО)

Москва 2021 г.

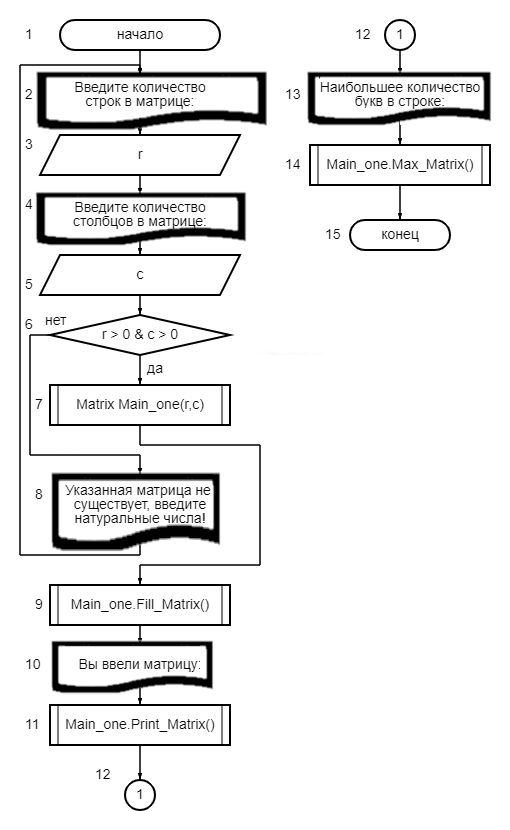
**Задание №3**

Задание: Задана символьная матрица. Найти номер первой по порядку строки, содержащей наибольшее количество букв

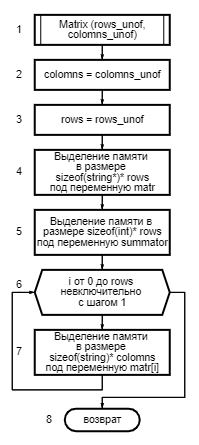
1. Таблица имён:

|  |  |  |
| --- | --- | --- |
| Исходные данные | | |
| r | целочисленный | Количество строк в матрице |
| c | целочисленный | Количество столбцов в матрице |
| Main\_one | Экземпляр класса Matrix | Исходно задаваемая пользователем  матрица строкового вида |
| Рабочие переменные | | |
| colomns | целочисленный | Количество столбцов в матрице |
| rows | целочисленный | Количество строк в матрице |
| matr | двойной указатель string | Исходно заданная матрица |
| summator | указатель целочисл. типа | Массив с кол-вом букв в исходной матрице |
| summator[i] | целочисленный | Количество букв в строке i исходной матрицы |
| i | целочисленный | Переменная – счётчик в циклах |
| j | целочисленный | Переменная - счётчик в циклах |
| matr[i][j] | строковый | Элемент исходной строковой матрицы |
| c | автоматический | Переменная для перебора слов в циклах |
| max\_amount | целочисленный | Счётчик букв в строке матрицы |
| Результат | | |
| max\_row | целочисленный | Строка с наибольшим количеством букв |

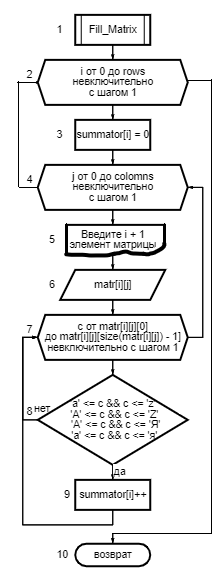
1. Блок-схема:



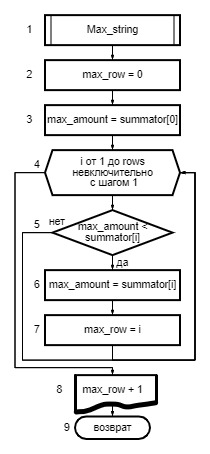
Блок-схема №1 – Основная фунцкия программы (main)



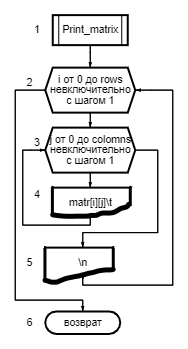
Блок-схема №2 – Конструктор класса Matrix



Блок-схема №3 – Метод заполнения матрицы



Блок-схема №4 – Выборка строки с наибольшим кол-вом букв



Блок-схема №5 – Метод вывода матрицы на экран

1. Отладочный пример:

1 вариант.

1) Начало  
2) Вывод на экран – “Введите количество строк в матрице: ”  
3) Ввод с клавиатуры r = 2  
4) Вывод на экран - “Введите количество столбцов в матрице: ”  
5) Ввод с клавиатуры c = 2  
6) 2 > 0 & 2 > 0, да, переход к блоку 7  
7) Переход к выполнению конструктора класса, блок-схеме №2.  
2а) Переприсваивание переменных colomns = 2  
3a) Переприсваивание переменных rows = 2  
4a) Выделение памяти в размере sizeof(string\*)\*2 под переменную matr  
5a) Выделение памяти в размере sizeof(int)\*2 под переменную summator  
6a) Цикл: Переменная i = 0  
7a) Выделение памяти в размере sizeof(string)\*2 под переменную matr[0]  
6a) Цикл: Переменная i = 1  
7a) Выделение памяти в размере sizeof(string)\*2 под переменную matr[1]  
6a) Цикл: Переменная i = 2, конец цикла, переход к блоку 8а  
8а) Возврат в основную блок-схему  
9а) Переход к выполнению метода класса, блок-схеме №3.  
2б) Цикл: Переменная i = 0  
3б) Переприсваивание переменной summator[0] = 0  
4б) Цикл: Переменная j = 0  
5б) Вывод на экран – “Введите 1 элемент матрицы: ”  
6б) Ввод matr[0][0] = “r”  
7б) Цикл: Переменная c = “r”  
7б) Проверка на букву – да, переход к блоку 8б  
8б) Переприсваивание переменной summator[0] += 1 = 1  
7б) Конец цикла, возвращение к блоку 4б  
4б) Цикл: Переменная j = 1  
5б) Вывод на экран – “Введите 2 элемент матрицы: ”  
6б) Ввод matr[0][1] = “r”  
7б) Цикл: Переменная c = “r”  
8б) Проверка на букву – да, переход к блоку 8б  
9б) Переприсваивание переменной summator[0] += 1 = 2  
7б) Конец цикла, возвращение к блоку 4б  
4б) Конец цикла, возврат к блоку 2б  
2б) Цикл: Переменная i = 1  
3б) Переприсваивание переменной summator[1] = 0  
4б) Цикл: Переменная j = 0  
5б) Вывод на экран – “Введите 3 элемент матрицы: ”  
6б) Ввод matr[1][0] = “1”  
7б) Цикл: Переменная c = “1”  
8б) Проверка на букву – нет, переход к блоку 6б  
7б) Конец цикла, возвращение к блоку 4б  
4б) Цикл: Переменная j = 1  
5б) Вывод на экран – “Введите 4 элемент матрицы: ”  
6б) Ввод matr[1][1] = “1”  
7б) Цикл: Переменная c = “1”  
8б) Проверка на букву – нет, переход к блоку 6б  
7б) Конец цикла, возвращение к блоку 4б  
5б) Конец цикла, возврат к блоку 2б  
3б) Конец цикла, переход к блоку 9б  
10б) Возвращение в основную блок-схему №1  
10) Вывод на экран – “Вы ввели матрицу: ”  
11) Переход к выполнению метода класса, блок-схеме №5  
2г) Цикл: Переменная i = 0  
3г) Цикл: Переменная j = 0  
4г) Вывод на экран – “r\t”  
3г) Цикл: Переменная j = 1  
4г) Вывод на экран – “r\t”  
3г) Цикл завершён, переход к блоку 2г  
2г) Цикл: Переменная i = 1  
3г) Цикл: Переменная j = 0  
4г) Вывод на экран – “1\t”  
3г) Цикл: Переменная j = 1  
4г) Вывод на экран – “1\t”  
3г) Цикл завершён, переход к блоку 2г  
2г) Цикл завершён, переход к блоку 6г  
6г) Возврат в основную блок-схему №1  
13) Вывод на экран – “Наибольшее количество букв в строке: ”  
14) Переход к выполнению метода класса, блок-схеме №4  
2в) Переприсваивание переменных max\_row = 0  
3в) Переприсваивание переменных max\_amount = 2  
4в) Цикл: Переменная i = 1  
5в) Условие - 2 < 0, нет, переход к блоку 4в  
4в) Цикл завершён, переход к блоку 8в  
8в) Вывод на экран – “1”  
9в) Возврат в основную блок-схему №1  
15) Конец программы.

1. Код программы:

#include <iostream>

#include <string>

using namespace std;

class Matrix {

private:

int colomns, rows;

string\*\* matr;

int\* summator;

public:

//Выделяем память под матрицу

Matrix(int rows\_unof, int colomns\_unof) {

colomns = colomns\_unof, rows = rows\_unof;

matr = new string \* [rows];

summator = new int[rows];

for (int i = 0; i < rows; i++)

matr[i] = new string[colomns];

}

//Заполняем матрицу и массив с количеством букв в строках

void Fill\_Matrix() {

for (int i = 0; i < rows; i++) {

summator[i] = 0;

for (int j = 0; j < colomns; j++) {

printf("Введите %d элемент матрицы: ", i \* colomns + j + 1);

cin >> matr[i][j];

for (auto c : matr[i][j])

if ('a' <= c && c <= 'z' ||

'A' <= c && c <= 'Z' ||

'А' <= c && c <= 'Я' ||

'а' <= c && c <= 'я')

summator[i]++;

}

}

}

//Выбираем наибольший элемент среди массива с количеством букв в строках

void Max\_string() {

int max\_row = 0, max\_amount = summator[0];

for (int i = 1; i < rows; i++) {

if (max\_amount < summator[i]) {

max\_amount = summator[i];

max\_row = i;

}

}

cout << max\_row + 1 << endl;

}

//Выводим всё содержимое матрицы на экран пользователю

void Print\_matrix() {

for (int i = 0; i < rows; i++) {

for (int j = 0; j < colomns; j++)

cout << matr[i][j] << "\t ";

cout << endl;

}

}

};

int main() {

//Для избегания искажений с латиницей

//подключаем кодировки русского языка

setlocale(LC\_ALL, "rus");

system("chcp 1251 >> null");

int r, c;

// Запрос пользователю на ввод основных переменных

while (true) {

cout << "Введите количество строк в матрице: ";

cin >> r;

cout << "Введите количество столбцов в матрице: ";

cin >> c;

if (r > 0 && c > 0)

break;

cout << "Указанная матрица не существует, введите натуральные числа!" << endl;

}

// Поэтапно вызываем мнемоничные методы для создаваемого экземпляра класса

Matrix Main\_one(r, c);

Main\_one.Fill\_Matrix();

cout << "Вы ввели матрицу: " << endl;

Main\_one.Print\_matrix();

cout << "Наибольшее количество букв в строке ";

Main\_one.Max\_string();

system("pause");

return 0;

}

1. Результат выполнения работы программы:

Изображение выглядит как текст

Автоматически созданное описание

Отладка №1

1. Вывод:

В ходе выполнения работы были изучены основные способы обёртки

матриц в своём единообразии в отдельный класс, в котором можно создавать различные методы для повышения читаемости кода в главной функции main(), названия для методов мнемоничны, поэтому глядя на вызов метода, мы уже сможем понять, каково его предназначение в программе.

На контрольных примерах мы убедились, что код работает корректно.  
Был оформлен комплект документации на программный код.