**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра АПУ**

Курсовая РАБОТА

**по дисциплине «Программирование»**

|  |  |  |
| --- | --- | --- |
| Студент гр. 4391 |  | Мухачев Д. О. |
| Преподаватель |  | Писарев А. С. |

Санкт-Петербург

2024

**ЗАДАНИЕ**

**НА КУРСОВУЮ РАБОТУ**

|  |
| --- |
| Студент Мухачев Д. О. |
| Группа 4391 |
| Исходные данные:  Программа должна оперировать тремя квадратными матрицами (А, В и С), размерность которых задается (или определяется) на фазе работы программы.  Необходимо:  1. Реализовать динамические структуры данных и алгоритмы их обработки, позволяющие поддерживать выполнение следующих функций:  1) консольный ввод/вывод данных о матрицах А, В и С;  2) файловый ввод/вывод данных о матрицах А, В и С;  3) интерактивное редактирование элементов матриц;  2. Скопировать в матрицу С столбцы матриц А или В в зависимости от того, какой из столбцов содержит большее количество нулей  Программа должна поддерживать систему меню и подменю, пункты которых соответствуют выполнению функций, предусмотренных заданием.  Содержание пояснительной записки:  «Содержание», «Введение», «Цель и задачи», «Описание кода»,  «Тестирование», «Заключение», «Список использованных  источников», «Приложение А. Полный текст программы» |

|  |  |  |
| --- | --- | --- |
| Предполагаемый объем пояснительной записки:  Не менее 00 страниц. | | |
| Дата выдачи задания: 7 ноября 2024 | | |
| Дата сдачи реферата: 18 декабря 2024 | | |
| Дата защиты реферата: 20 декабря 2024 | | |
| Студент |  | Мухачев Д. О. |
| Преподаватель |  | Писарев А. С. |

**Аннотация**

Курсовая работа включает разработку программы на языке C++, которая демонстрирует работу с тремя квадратными матрицами (A, B и C) с динамически определяемой размерностью. В проекте реализованы функции для консольного и файлового ввода/вывода данных о матрицах, а также интерактивного редактирования их элементов. Основное внимание уделяется алгоритму, который заменяет столбцы матрицы C аналогичными столбцами из матриц A или B в зависимости от количества нулей в соответствующих столбцах. Если в столбце матрицы A больше нулей, чем в столбце матрицы B, соответствующий столбец из A заменяет столбец в C; в противном случае происходит замена на столбец из B. Результаты работы программы подтверждают успешное выполнение всех операций с матрицами и корректное отображение изменённой матрицы C. Программа имеет удобный пользовательский интерфейс, обеспечивающий простое взаимодействие через консоль.

**содержание**

[**ВВЕДЕНИЕ** 6](#_Toc185496636)

[**1. ЦЕЛЬ И ЗАДАЧИ** 7](#_Toc185496637)

[1.1. Цель 7](#_Toc185496638)

[1.2. Задачи 7](#_Toc185496639)

[**2. ОПИСАНИЕ КОДА** 7](#_Toc185496640)

[2.1. Подключение библиотек 7](#_Toc185496641)

[2.2. Функция для ввода элементов матрицы из файла 8](#_Toc185496642)

[2.3. Функция для ввода элементов матрицы с клавиатуры 8](#_Toc185496643)

[2.4. Функция для вывода матрицы в файл: 9](#_Toc185496644)

[2.5. Функция вывода матрицы на экран: 10](#_Toc185496645)

[2.6. Алгоритм замены столбцов матрицы С на основе количества нулей в столбцах матриц А и В 10](#_Toc185496646)

[2.11. Основная функция 11](#_Toc185496647)

[2.12. Основное меню программы 12](#_Toc185496648)

[**3. ТЕСТИРОВАНИЕ** 14](#_Toc185496649)

[3.1. Ввод данных и успешное тестирование 14](#_Toc185496650)

[3.2. Отображение сохранённых матриц в файле 15](#_Toc185496651)

[**ЗАКЛЮЧЕНИЕ** 16](#_Toc185496652)

[**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ** 17](#_Toc185496653)

[**ПРИЛОЖЕНИЕ А. ПОЛНЫЙ ТЕКСТ ПРОГРАММЫ** 18](#_Toc185496654)

# **ВВЕДЕНИЕ**

В условиях современного мира, где обработка данных становится все более актуальной, матричные операции играют ключевую роль в различных областях науки и техники. Данная курсовая работа посвящена разработке программы на языке C++, которая осуществляет работу с тремя квадратными матрицами, обозначенными как A, B и C. Размерность этих матриц определяется на этапе выполнения программы, что обеспечивает гибкость и адаптивность в работе с различными наборами данных.

Целью проекта является создание функционального программного обеспечения, которое включает в себя консольный и файловый ввод/вывод данных о матрицах, а также интерактивное редактирование их элементов. Одной из ключевых задач является реализация алгоритма, который заменяет столбцы матрицы C аналогичными столбцами из матриц A или B в зависимости от количества нулей в соответствующих столбцах. Это требует применения динамических структур данных и эффективных алгоритмов для оптимизации производительности и управления памятью.

Программа также включает систему меню и подменю, что позволяет пользователям легко выбирать необходимые операции и управлять процессом работы с матрицами. Таким образом, проект направлен на создание комплексного решения для обработки данных в виде матриц, что может быть полезно в различных вычислительных задачах и исследованиях. Код, представленный в работе, не только решает конкретную задачу, но и углубляет понимание принципов работы с динамическими структурами данных, а также теоретических и практических аспектов реализации алгоритмов на языке C++.

## **1. ЦЕЛЬ И ЗАДАЧИ**

### 1.1. Цель

Разработка программы для работы с тремя квадратными матрицами (A, B и C), которая включает в себя создание, редактирование, сохранение и загрузку данных, а также выполнение анализа на основе суммы элементов столбцов.

### 1.2. Задачи

Создать функцию файлового вывода матрицы

Реализовать функцию консольного вывода матрицы.

Разработать функцию файлового ввода матрицы.

Создать функцию консольного ввода матрицы.

Разработать алгоритм, который копирует в матрицу С столбцы матриц А или В в зависимости от того, какой из столбцов содержит большее количество нулей

Создать меню и подменю для управления программой и выполнения всех предусмотренных функций.

## **2. ОПИСАНИЕ КОДА**

### 2.1. Подключение библиотек

#include <iostream>: Подключает библиотеку для работы с вводом и выводом в стандартный поток (консоль).

#include <fstream>: Подключает библиотеку для работы с файловым вводом и выводом.

#include <string>: Подключает библиотеку для работы со строками.

#include <io.h> и #include <fcntl.h> используются для поддержки UNICODE.

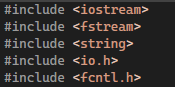


Рисунок 1 – Подключенные библиотеки

### 2.2. Функция для ввода элементов матрицы из файла

Была написана функция fin\_mat, которая предназначена для ввода элементов квадратной матрицы из файла.

Указатель на double (double\* mat), представляет собой двумерный массив (в данном случае — матрицу). Этот параметр позволяет функции работать с динамически выделенной памятью для матрицы. Целочисленный параметр (int n), который указывает размерность матрицы (так как матрица квадратная, это число будет равно количеству строк и столбцов). Используется поток ввода файла fin, который функция получает в качестве параметра. Далее был написан цикл для ввода элементов. Цикл (for (int i = 0; i < n \* n; i++)) проходит по всем элементам матрицы. Переменная i представляет индекс текущего элемента так, как будто матрица является одномерным массивом. Далее было написано считывание значения из файла с пропуском разделителя «;» формата CSV, и сохранение его в соответствующую ячейку i матрицы .

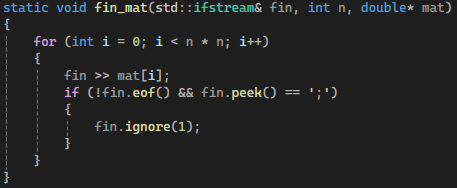


Рисунок 3 – Функция для ввода матрицы из файла

### 2.3. Функция для ввода элементов матрицы с клавиатуры

Была написана функция cin\_mat, которая предназначена для ввода элементов квадратной матрицы с клавиатуры.

Функция полностью аналогична fin\_mat за исключением использования потока ввода cin вместо потока ввода файла fin, а также функция не удаляет разделители CSV, потому что при вводе с клавиатуры это не нужно.

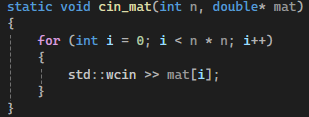


Рисунок 3 – Функция для ввода матрицы из файла

### 2.4. Функция для вывода матрицы в файл:

Была написана функция fout\_mat, которая предназначена для вывода элементов квадратной матрицы в файл.

Используется поток вывода файла fout, который функция получает в качестве параметра. Создан двойной цикл для вывода элементов матрицы. Внешний цикл (for (int i = 0; i < n; i++)) проходит по всем строкам матрицы. Переменная i представляет индекс текущей строки. Внутренний цикл (for (int j = 0; j < n-1; j++)) проходит по всем столбцам текущей строки, кроме последнего. Переменная j представляет индекс текущего столбца. fout << matrix[i\*n+j] << ";" выводит значение ячейки матрицы, находящейся в строке i и столбце j, в файл. После каждого значения добавляется CSV разделитель. fout << matrix[(i+1)\*n-1] << “\n” После завершения внутреннего цикла (т.е. после вывода всех элементов текущей строки) происходит вывод последнего элемента без разделителя и переход на новую строку.

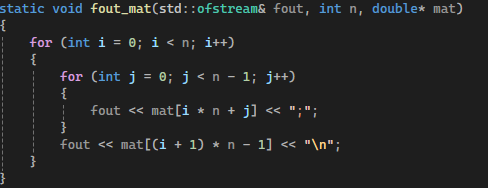


Рисунок 4 – Функция для вывода матрицы на экран

### 2.5. Функция вывода матрицы на экран:

Была написана функция cout\_mat, которая предназначена для вывода элементов квадратной матрицы на экран.

Функция полностью аналогична fout\_mat за исключением использования потока вывода cout вместо потока вывода файла fout, а также функция выводит значения через пробелы вместо разделителей CSV для лучшей читаемости.

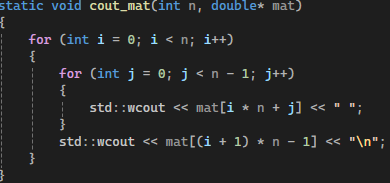


Рисунок 5 – Функция для редактирования элементов матрицы

### 2.6. Алгоритм замены столбцов матрицы С на основе количества нулей в столбцах матриц А и В

Был написан алгоритм замены столбцов матрицы С в соответствии с условием задачи. Сначала идет подсчет нулей в столбце матрицы А и в столбце матрицы В. После этого столбец с большим количеством нулей копируется в матрицу С. Если нулей в столбцах поровну, то будет скопирован столбец матрицы А. Такая операция выполняется для всех столбцов матрицы С.

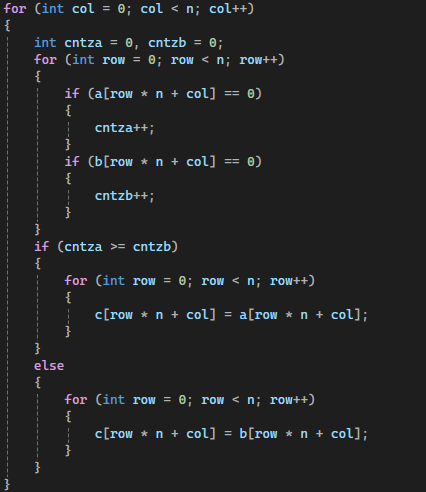


Рисунок 6 – Функция для замены в матрице С на основе сумм столбцов матриц А и В

### 2.11. Основная функция

Создано тело функции. Добавлена поддержка UNICODE. Было выполнено объявление переменной n, которая будет использоваться для хранения размерности квадратных матриц (количество строк и столбцов) и переменной cmd, хранящей номер последней введенной команды. Были созданы указатели a, b, c для хранения данных матриц A, B, C. Далее был написан цикл интерактивного меню, имеющего 8 возможных команд:

1 - ввод матриц A, B и C из консоли;

2 - ввод матриц А, В и С из файла;

3 - редактирование элементов матриц A, B и C;

4 - выполнение алгоритма из условия задачи;

5 - вывод матриц в консоль;

6 - вывод матриц в файл;

7 - вывод списка команд;

0 - выход из программы.

Далее идёт считывание введенной пользователем команды и сохранение его в переменной cmd. Было написано динамическое выделение памяти для трех матриц: A, B и C. После этого идёт выделение памяти для строк матриц.

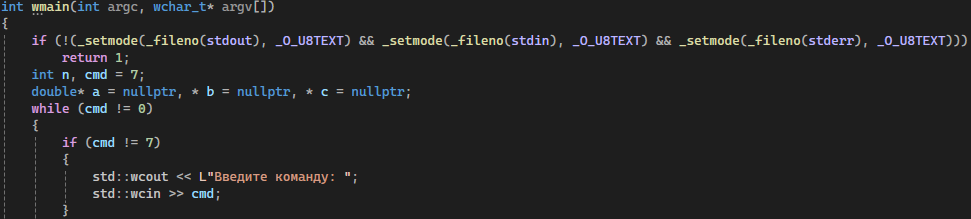


Рисунок 7 – Основная функция

### 2.12. Основное меню программы

Было написано меню программы, позволяющего пользователю выбирать различные действия, связанные с матрицами. Переменная cmd, объявленная ранее, используется для хранения текущей команды. Цикл while гарантирует, что содержимое блока будет выполнено хотя бы один раз, потому что на момент начала цикла cmd равен 7. Цикл будет продолжаться до тех пор, пока значение cmd не станет равным 0, что соответствует пункту "Выход". Это позволяет пользователю многократно выбирать действия из меню, пока он не решит завершить программу.

Далее был написан вывод меню. Каждая строка представляет собой отдельный пункт меню, который описывает доступные действия, которые пользователь может выбрать. Далее выводится сообщение, предлагающее пользователю ввести свой выбор. Затем идёт считывание введенного пользователем значения и сохранение его в переменной cmd. Это значение будет использоваться для определения того, какое действие выполнить в соответствии с пунктом меню.



Рисунок 8 – Основное меню программы

## **3. ТЕСТИРОВАНИЕ**

### 3.1. Ввод данных и успешное тестирование

Программа вывела меню, был выбран консольный ввод и после ввода размерности матриц было произведено заполнение числами.

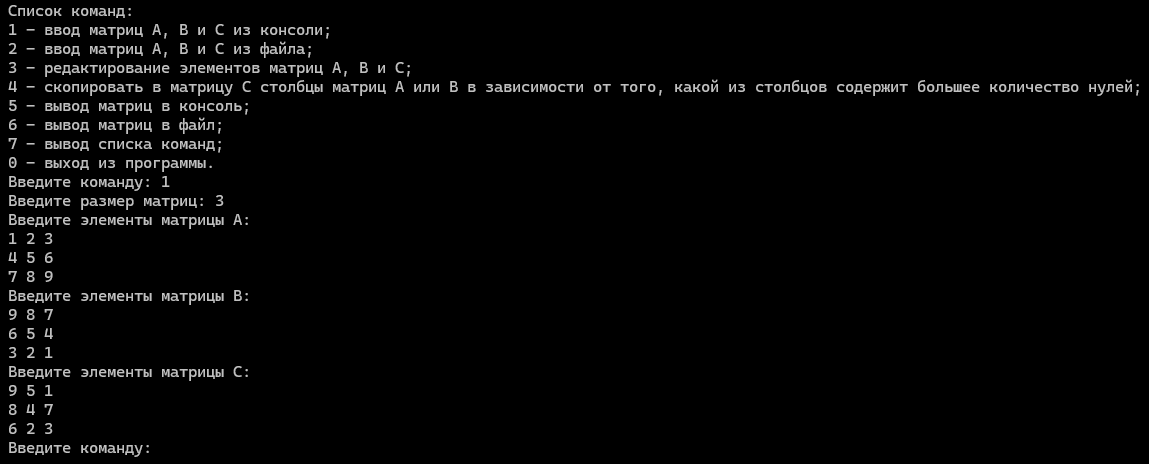


Рисунок 9 – Ввод данных

Было произведено редактирование элементов матриц:

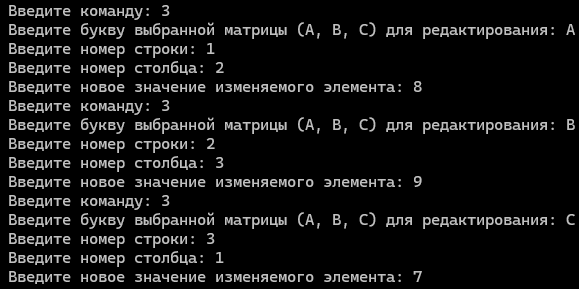


Рисунок 10 – Изменение элемента в матрице

Была выполнена замена столбцов матрицы C аналогичными столбцами из матриц A или B в зависимости от количества нулей в соответствующих столбцах. Программа выдала правильный результат.

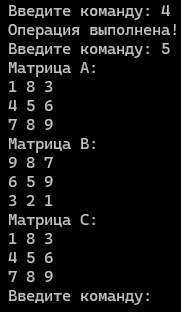


Рисунок 16 – Заполнение матриц и замена столбцов в матрице С

### 3.2. Отображение сохранённых матриц в файле

Матрица С была сохранена в файл. Программа успешно сохранила матрицу.

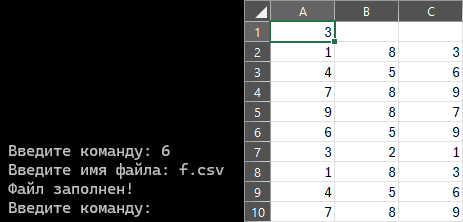


Рисунок 17 – Матрица С сохранена в файл

# **ЗАКЛЮЧЕНИЕ**

При выполнении курсовой работы была разработана программа для работы с квадратными матрицами, включающей создание, редактирование, сохранение и загрузку данных, а также анализ элементов. Цель работы была достигнута: посредством выполнения задач была разработана программа, полностью удовлетворяющая условию работы.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. https://metanit.com/cpp/tutorial/

2. http://cppstudio.com/

3. Бьярне Страуструп. Язык программирования С++. Краткий курс, второе издание.

4. Брайан Керниган. Деннис Ритчи. Язык программирования. Второе издание.

5.Васильев А.Н. Программирование на С++ в примерах и задачах.

# **ПРИЛОЖЕНИЕ А. ПОЛНЫЙ ТЕКСТ ПРОГРАММЫ**

#include <iostream>

#include <fstream>

#include <string>

#include <io.h>

#include <fcntl.h>

static void fin\_mat(std::ifstream& fin, int n, double\* mat)

{

for (int i = 0; i < n \* n; i++)

{

fin >> mat[i];

if (!fin.eof() && fin.peek() == ';')

{

fin.ignore(1);

}

}

}

static void cin\_mat(int n, double\* mat)

{

for (int i = 0; i < n \* n; i++)

{

std::wcin >> mat[i];

}

}

static void fout\_mat(std::ofstream& fout, int n, double\* mat)

{

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

{

for (int j = 0; j < n - 1; j++)

{

fout << mat[i \* n + j] << ";";

}

fout << mat[(i + 1) \* n - 1] << "\n";

}

}

static void cout\_mat(int n, double\* mat)

{

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

{

for (int j = 0; j < n - 1; j++)

{

std::wcout << mat[i \* n + j] << " ";

}

std::wcout << mat[(i + 1) \* n - 1] << "\n";

}

}

int wmain(int argc, wchar\_t\* argv[])

{

if (!(\_setmode(\_fileno(stdout), \_O\_U8TEXT) && \_setmode(\_fileno(stdin), \_O\_U8TEXT) && \_setmode(\_fileno(stderr), \_O\_U8TEXT)))

return 1;

int n, cmd = 7;

double\* a = nullptr, \* b = nullptr, \* c = nullptr;

while (cmd != 0)

{

if (cmd != 7)

{

std::wcout << L"Введите команду: ";

std::wcin >> cmd;

}

switch (cmd)

{

case 1:

{

std::wcout << L"Введите размер матриц: ";

std::wcin >> n;

delete[] a;

a = new double[n \* n];

delete[] b;

b = new double[n \* n];

delete[] c;

c = new double[n \* n];

std::wcout << L"Введите элементы матрицы A:\n";

cin\_mat(n, a);

std::wcout << L"Введите элементы матрицы B:\n";

cin\_mat(n, b);

std::wcout << L"Введите элементы матрицы C:\n";

cin\_mat(n, c);

break;

}

case 2:

{

std::wstring name;

std::wcout << L"Введите имя файла: ";

std::wcin >> name;

std::ifstream fin;

fin.open(name);

if (!fin)

{

std::wcout << L"Некорректная команда!";

break;

}

fin >> n;

delete[] a;

a = new double[n \* n];

delete[] b;

b = new double[n \* n];

delete[] c;

c = new double[n \* n];

fin\_mat(fin, n, a);

fin\_mat(fin, n, b);

fin\_mat(fin, n, c);

fin.close();

std::wcout << L"Файл считан!\n";

break;

}

case 3:

{

int row, col;

double val;

std::wcout << L"Введите букву выбранной матрицы (A, B, C) для редактирования: ";

wchar\_t name;

std::wcin >> name;

row = 1;

std::wcout << L"Введите номер строки: ";

std::wcin >> row;

std::wcout << L"Введите номер столбца: ";

std::wcin >> col;

std::wcout << L"Введите новое значение изменяемого элемента: ";

std::wcin >> val;

switch (name)

{

case L'A':

{

a[(row - 1) \* n + col - 1] = val;

break;

}

case L'B':

{

b[(row - 1) \* n + col - 1] = val;

break;

}

case L'C':

{

c[(row - 1) \* n + col - 1] = val;

break;

}

default: {

std::wcout << L"Некорректная команда!";

break;

}

}

break;

}

case 4:

{

for (int col = 0; col < n; col++)

{

int cntza = 0, cntzb = 0;

for (int row = 0; row < n; row++)

{

if (a[row \* n + col] == 0)

{

cntza++;

}

if (b[row \* n + col] == 0)

{

cntzb++;

}

}

if (cntza >= cntzb)

{

for (int row = 0; row < n; row++)

{

c[row \* n + col] = a[row \* n + col];

}

}

else

{

for (int row = 0; row < n; row++)

{

c[row \* n + col] = b[row \* n + col];

}

}

}

std::wcout << L"Операция выполнена!\n";

break;

}

case 5:

{

std::wcout << L"Матрица A:\n";

cout\_mat(n, a);

std::wcout << L"Матрица B:\n";

cout\_mat(n, b);

std::wcout << L"Матрица C:\n";

cout\_mat(n, c);

break;

}

case 6:

{

std::wstring name;

std::wcout << L"Введите имя файла: ";

std::wcin >> name;

std::ofstream fout;

fout.open(name, std::ios::app);

if (!fout)

{

std::wcout << L"Некорректная команда!";

break;

}

fout << n << '\n';

fout\_mat(fout, n, a);

fout\_mat(fout, n, b);

fout\_mat(fout, n, c);

fout.close();

std::wcout << L"Файл заполнен!\n";

break;

}

case 7:

{

std::wcout << L"Список команд:\n1 - ввод матриц A, B и C из консоли;\n2 - ввод матриц А, В и С из файла;\n3 - редактирование элементов матриц A, B и C;\n4 - скопировать в матрицу С столбцы матриц А или В в зависимости от того, какой из столбцов содержит большее количество нулей;\n5 - вывод матриц в консоль;\n6 - вывод матриц в файл;\n7 - вывод списка команд;\n0 - выход из программы.\n";

cmd = 1;

break;

}

case 0:

{

std::wcout << L"Программа завершена.\n";

break;

}

default:

{

std::wcout << L"Некорректная команда!\n";

break;

}

}

}

delete[] a, b, c;

}