

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ
СТРОИТЕЛЬНЫЙ УНИВЕРСИТЕТ»**

Институт Экономики, управления и информационных систем в строительстве

Кафедра Информационных систем, технологий и автоматизации в строительстве

КУРСОВОЙ ПРОЕКТ

по дисциплине

«Алгоритмизация. Технология разработки программного обеспечения»

Тема: **«Программное приложение "Приведение структуры матрицы к
ленточной форме (метод 2)"»**

Выполнил студент Михалин Дмитрий Сергеевич, ЭУИСмаг 1-3
(Ф. И. О.)

Руководитель проекта к.т.н., доц. Китайцева Елена Халиловна
(ученое звание, степень, должность, Ф.И.О.)

К защите _____
(дата, подпись руководителя)

Проект защищен с оценкой _____
(оценка)

г. Москва
2020 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ
СТРОИТЕЛЬНЫЙ УНИВЕРСИТЕТ»**

Институт Экономики, управления и информационных систем в строительстве

Кафедра Информационных систем, технологий и автоматизации в строительстве

РАСЧЁТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Тема: **«Программное приложение "Приведение структуры матрицы к
ленточной форме (метод 2)"»**

Выполнил студент Михалин Дмитрий Сергеевич, ЭУИСмаг 1-3
(Ф. И. О.)

Руководитель проекта к.т.н., доц. Китайцева Елена Халиловна
(ученое звание, степень, должность, Ф.И.О.)

г. Москва
2020 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ
СТРОИТЕЛЬНЫЙ УНИВЕРСИТЕТ»**

Институт Экономики, управления и информационных систем в строительстве
Кафедра Информационных систем, технологий и автоматизации в строительстве

«Утверждаю»:
заведующий кафедрой
д. т. н., профессор Гинзбург А. В.
(ученое звание, степень, Ф.И.О.)

(подпись)

ЗАДАНИЕ ПО КУРСОВОМУ ПРОЕКТИРОВАНИЮ

Студенту группы ИЭУИСм 1-3 Михалину Дмитрию Сергеевичу

1. Тема курсового проекта: «Программное приложение "Приведение структуры матрицы к ленточной форме (метод 2)»
2. Срок сдачи проекта: 17 декабря 2020 г.
3. Исходные данные к курсовому проекту: Алгоритм алгоритм уменьшения ширины ленты разреженных симметричных матриц Катхилла – МакКи
4. Содержание расчетно-пояснительной записки (перечень подлежащих разработке вопросов):
 - Описание алгоритма;
 - Структура приложения;
 - Руководство пользователя
5. Перечень графического материала (с точным указанием обязательных чертежей): _____
6. Консультанты по проекту (работе): _____
7. Задание выдал 01 октября 2020 г. руководитель проекта Китайцева Е. Х.
8. Задание получил 01 октября 2020 г. студент Михалин Д. С.
9. Дополнительные указания *(по усмотрению руководителя)*
10. График индивидуальных консультаций:

№ п/п	Дата консультации	Продолжительность консультации	Подпись консультанта
1	08 октября 2020 г.	10 минут	
2	22 октября 2020 г.	10 минут	
3	05 ноября 2020 г.	20 минут	
4	19 ноября 2020 г.	10 минут	
5	03 декабря 2020 г.	10 минут	

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	5
Глава 1. Теоретическая часть	6
1.1. Математическая формулировка алгоритма	6
1.2. Пример выполнения алгоритма	7
Глава 2. Практическая часть	11
2.1. Проектирование программного решения	11
2.1.1. Требования к функционалу	11
2.1.2. Структура программы	12
2.2. Реализация поставленной задачи	13
ЗАКЛЮЧЕНИЕ	22
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	23
Приложение А. Диаграмма классов	24

ВВЕДЕНИЕ

Разрежённой матрицей называется матрица с преимущественно нулевыми элементами. При хранении и преобразовании таких матриц в компьютере бывает полезно, а часто и необходимо, использовать специальные алгоритмы и структуры данных, которые учитывают разрежённую структуру матрицы. Операции и алгоритмы, применяемые для работы с обычными, плотными матрицами, применительно к большим разрежённым матрицам работают относительно медленно и требуют значительных объёмов памяти. Однако разрежённые матрицы могут быть легко сжаты путём записи только своих ненулевых элементов, что снижает требования к компьютерной памяти.

В данном курсовом проекте будет рассмотрен алгоритм уменьшения ширины ленты разреженных симметричных матриц, который был назван по именам разработчиков – Элизабет Катхилл и Джеймса МакКи.

Глава 1. Теоретическая часть

1.1. Математическая формулировка алгоритма

Исходная симметричная матрица $n \times n$ рассматривается как матрица смежности графа (V, E) . Алгоритм Катхилла-МакКи перенумеровывает вершины графа таким образом, чтобы в результате соответствующей перестановки столбцов и строк исходной матрицы уменьшить ширину её ленты (т. е. область матрицы, удаленная от главной диагонали не более, чем на N позиций).

Алгоритм строит упорядоченный набор вершин R , представляющий новую нумерацию вершин. Для связного графа алгоритм выглядит следующим образом:

1. Выбрать периферийную вершину (или псевдопериферийную вершину) v для начального значения кортежа $R := (v)$;
2. Для $i = 1, 2, \dots$, пока выполнено условие $|R| < n$, выполнять шаги 3-5:
3. Построить множество смежности $Adj(R_i)$ для R_i , где R_i — i -ая компонента R , и исключить вершины, которые уже содержатся в R , то есть: $A_i := Adj(R_i) \setminus R$;
4. Отсортировать A_i по возрастанию степеней вершин;
5. Добавить A_i в кортеж результата R .

Другими словами, алгоритм нумерует вершины в ходе поиска в ширину, при котором смежные вершины обходятся в порядке увеличения их степеней.

Для несвязного графа алгоритм можно применить отдельно к каждой компоненте связности.

1.2. Пример выполнения алгоритма

1. Пусть дан случайный неориентированный связный граф (рис. 1.1).

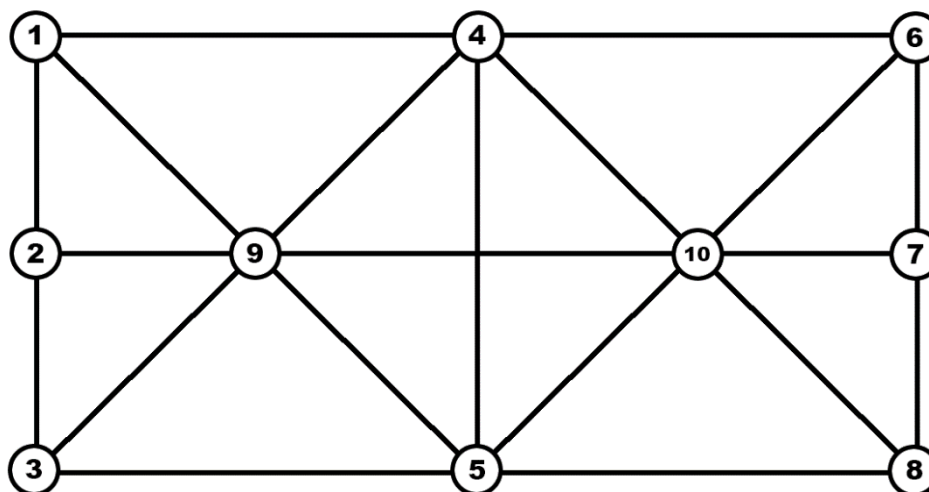


Рисунок 1.1 – граф

2. Матрица смежности данного графа будет выглядеть следующим образом (рис. 1.2).

	1	2	3	4	5	6	7	8	9	10
1	x	x		x					x	
2	x	x	x						x	
3		x	x		x				x	
4	x			x	x	x			x	x
5			x	x	x			x	x	x
6				x		x	x			x
7						x	x	x		x
8					x		x	x		x
9	x	x	x	x	x				x	x
10				x	x	x	x	x	x	x

Рисунок 1.2 – матрица смежности A

2. Для каждой вершины графа, соответствующего матрице, вычислить ее степень p_i , равную общему числу недиагональных единиц i -й строки матрицы

В. Таким образом, получаем следующие значения (табл. 1).

Таблица 1. Значения связей p_i каждой вершины i

i	1	2	3	4	5	6	7	8	9	10
p_i	3	3	3	5	5	3	3	3	6	6

3. Выбрать какую-либо вершину i_1 , для которой $p_i = \min_i p_i$ (в случае, если таких вершин несколько, то выбирается любая из них), и пометить эту вершину первой. В данном примере это вершина под номером 1.

Условно отнесем ее к **нулевому** уровню.

4. Присвоить вершинам, смежным с выбранной на предыдущем шаге вершиной 1, новые номера, начиная со 2, в порядке возрастания их степеней (если степени таких вершин совпадают, выбирать также любую из них). В данном примере вершинам 2, 4 и 9 присвоены новые номера 2, 3 и 4 соответственно (рис. 1.3).

Эти вершины будут отнесены к **первому** уровню.

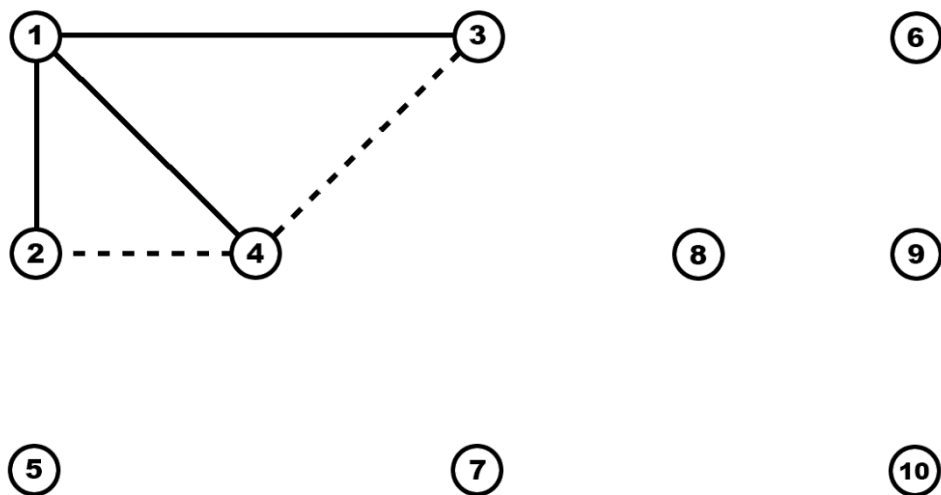


Рисунок 1.3 – перенумерация вершин, 1 уровень

5. Повторить пункт 4 для каждой вершины первого уровня по возрастанию их нового номера

Так, для вершины 2 (изначально 2), смежной с ней и еще не

перенумерованной вершиной будет вершина 3 - она станет вершиной 5.

Смежной с вершиной 3 (изначально 4) являются вершины 6, 5 и 10 (по возрастанию степени). Присвоим им номера 6, 7 и 8 соответственно.

Вершина 4 не имеет смежных не перенумерованных вершин, следовательно, будет пропущена.

Вершины 5, 6, 7 и 8 связаны с вершиной 1 путем длины 2 и относятся ко **второму** уровню (рис. 1.4).

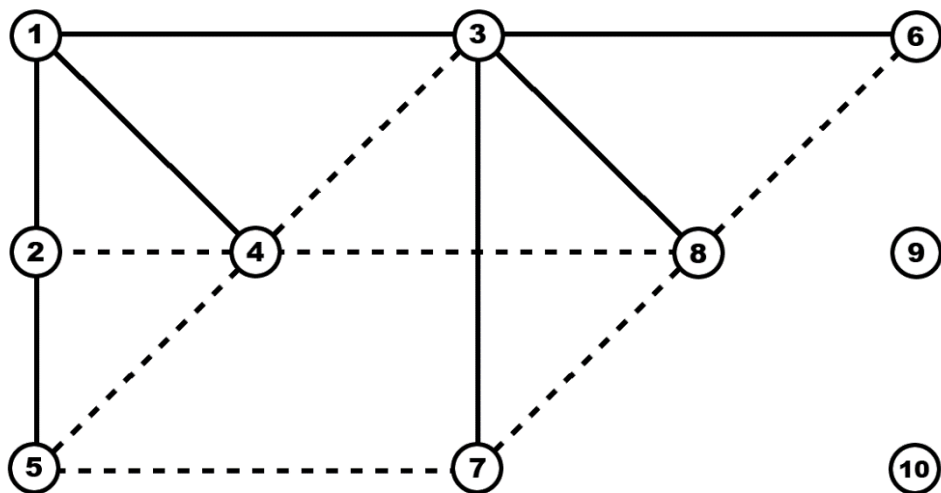


Рисунок 1.4 – перенумерация вершин, уровень 2

6. Повторить вышеизложенную процедуру для вершин каждого следующего уровня, пока все n вершин графа не будут перенумерованы (рис. 1.5).

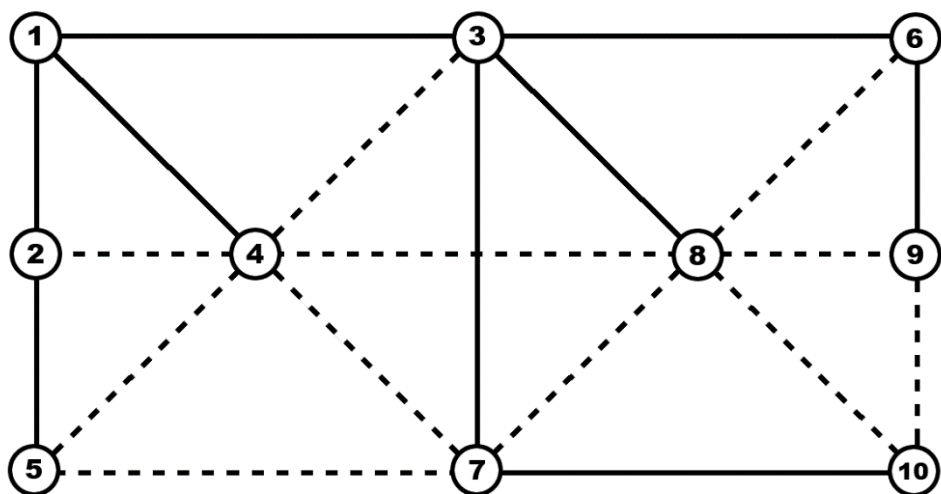


Рисунок 1.5 – перенумерация вершин, уровень 3

7. Последним этапом требуется переставить строки и столбцы матрицы в соответствии с новыми номерами вершин для получения ее преобразованной матрицы В (рис. 1.6). Заметим, что ширина ленты в матрице А равна 17, а в матрице В – 11.

	1	2	3	4	5	6	7	8	9	10
1	x	x	x	x						
2	x	x		x	x					
3	x		x	x		x	x	x		
4	x	x	x	x	x		x	x		
5		x		x	x		x			
6			x			x		x	x	
7			x	x	x		x	x		x
8			x	x		x	x	x	x	x
9						x		x	x	x
10							x	x	x	x

Рисунок 1.6 – матрица связности В

Глава 2. Практическая часть

В этой главе будет рассмотрен процесс проектирования и разработки программного решения.

2.1. Проектирование программного решения

На данном этапе необходимо сформулировать требования к разрабатываемому продукту, предусмотрев возможные исключения, исходя из которых составить ориентировочную структуру программы.

2.1.1. Требования к функционалу

Ввод данных

1. Считывание из файла в виде матрицы
2. Считывание из файла в виде пар вершин
3. Генерация случайной матрицы смежности
4. Ввод матрицы смежности пользователем
5. Ввод пар вершин пользователем

Ввод данных

1. Сохранение результата в файл
2. Запрет на сохранение файла до выполнения алгоритма
3. Корректный и наглядный вывод на экран матриц любой размерности

Алгоритм

1. Корректное выполнение алгоритма вне зависимости от размерности входной матрицы смежности
2. Запрет на запуск алгоритма до загрузки/ввода матрицы

Исключения

1. Прямоугольная матрица смежности
2. Различное количество символов в строке (не матрица)
3. Пустой или поврежденный файл
4. Граф не связный
5. Наличие посторонних символов в матрице

6. Наличие цифр в матрице, отличных от 0 или 1
7. Не симметричная матрица

2.1.2. Структура программы

Для поддержания модульности структуры программы функционал функционал был разбит на несколько классов, каждый из которых отвечает за набор действий в одной области (табл. 2).

Таблица 2. Разделение функционала на классы

Класс	Описание
App	Создает экземпляр приложения, производит начальную инициализацию и определяет точку входа
MainWindow	Логика взаимодействия пользователя с главным окном
AboutWindow	Логика взаимодействия пользователя с окном «О программе»
InputWindow	Представляет собой окно для ввода графа в виде матрицы смежности или списка пар вершин
LogWindow	Представляет собой окно для вывода протокола работы алгоритма и действий с ним
RankWindow	Представляет собой окно для ввода размерности генерируемой матрицы смежности
CuthillMethod	Содержит логику алгоритма Катхилла-МакКи
Vertex	Моделирует вершину графа и ее параметры
Graph	Моделирует неориентированный граф
Matrix	Абстрактный класс, предоставляющий методы по обработке, проверке и считыванию матриц

Графически вышеуказанную таблицу можно представить в виде диаграммы классов (Приложение А)

2.2. Реализация поставленной задачи

Алгоритм был реализован с помощью фреймворка .NET Framework 4.7.2 в среде разработки программного обеспечения Microsoft Visual Studio 2019 Community. Язык программирования – C# 7.3. Платформа – WPF.

Интерфейс главного окна изображен на рисунке ниже (рис. 2.1).

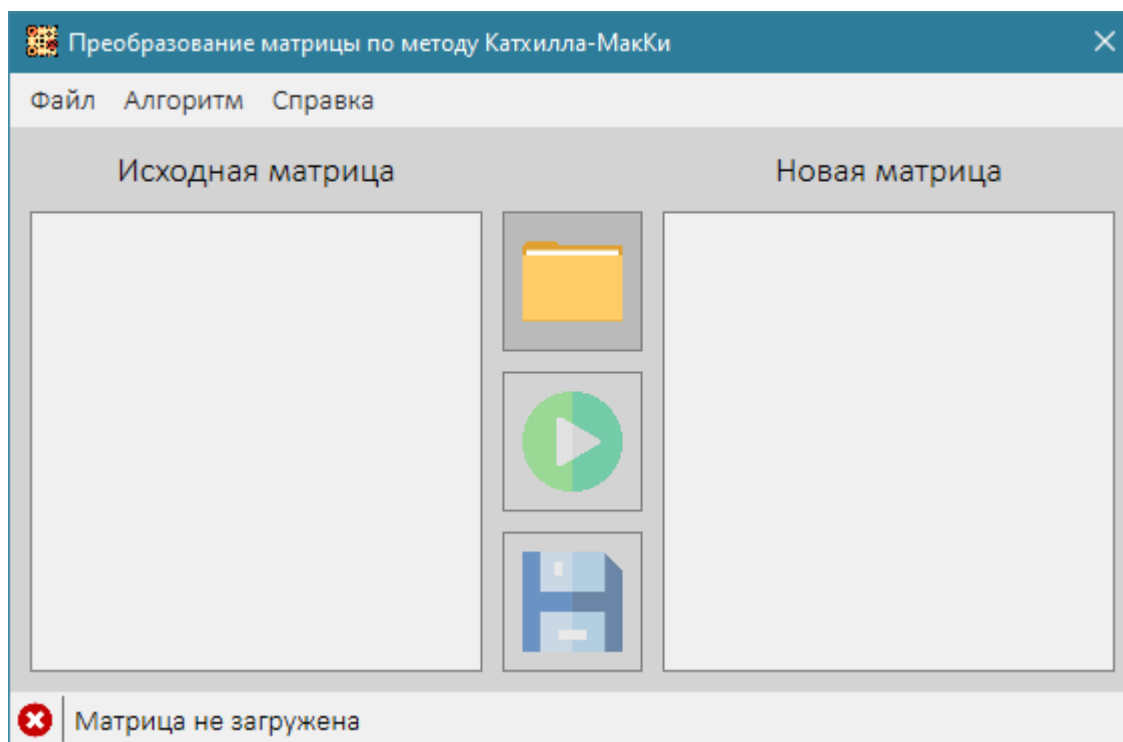


Рисунок 2.1 –Главное окно программы

Сверху расположен заголовок окна программы, содержащий ее название и элементы управления окном (свернуть, закрыть)

Ниже размещена панель меню. Пункт «Файл» отвечает за действия с входными и выходными данными, пункт «Алгоритм» содержит сопутствующий функционал по работе алгоритма, пункт «Справка» - информацию о программе.

В основной области окна находятся (слева направо, сверху вниз): поле вывода входной матрицы смежности, кнопка «Открыть файл», кнопка «Запустить алгоритм», кнопка «Сохранить результат в файл», поле вывода новой преобразованной матрицы.

Внизу окна находится строка состояния, т. н. статус-бар, отображающий информацию о открытом на данный момент файле.

Пункт меню «Файл» содержит следующие действия (рис. 2.2).

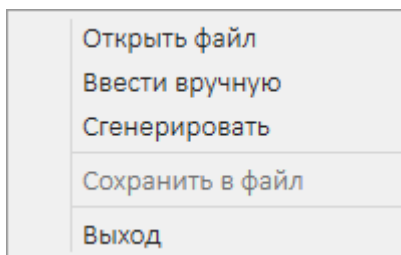


Рисунок 2.2 – Меню «Файл»

Если выбрать пункт «Открыть файл», появится стандартное окно выбора файла Windows (рис. 2.3).

Фильтр: только текстовые файлы (расширение .txt.).

Директория по умолчанию определена как «Рабочий стол».

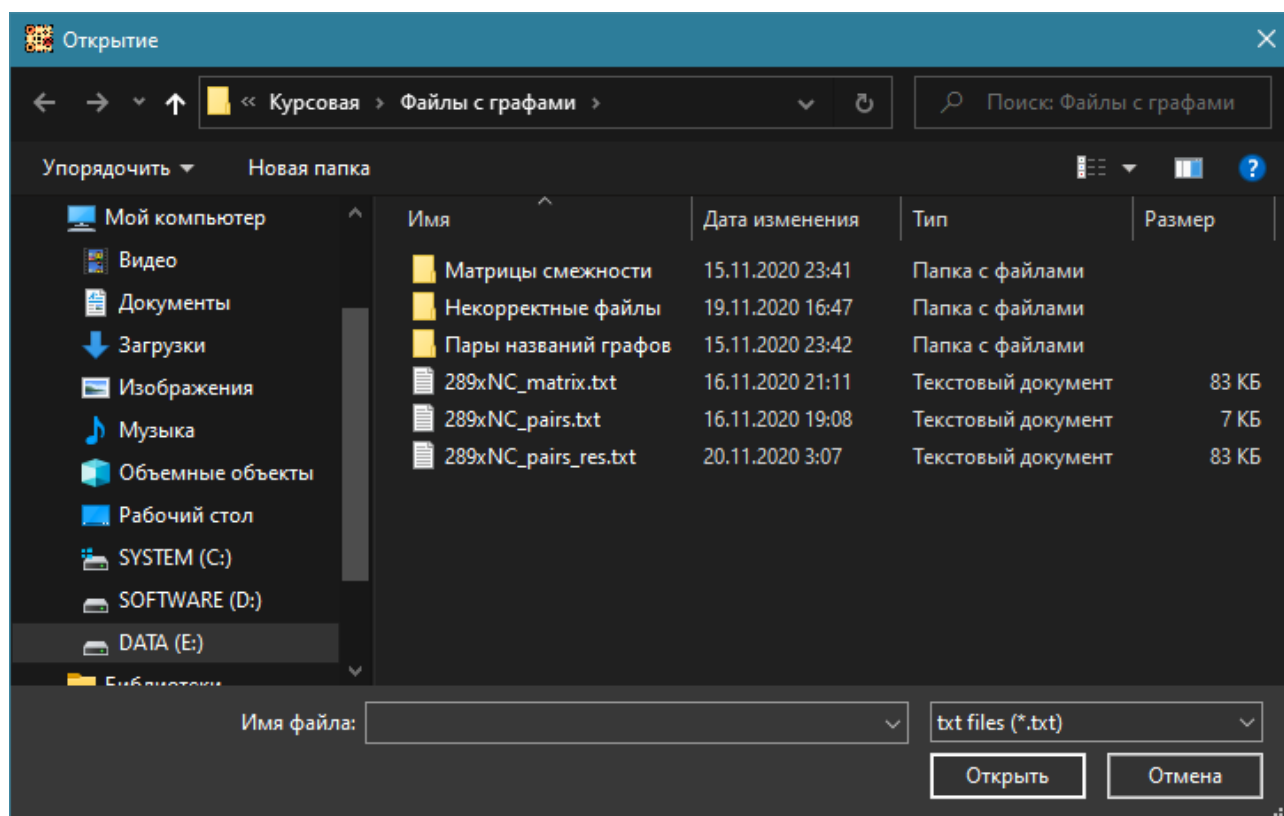


Рисунок 2.3 – Окно выбора файла для открытия

Если выбрать пункт «Ввести вручную», появится окно для ввода графа. Вводить в поле в левой части окна можно как матрицу смежности (рис. 2.4), так и пары вершин, при этом разделитель – символ двоеточия «:» (рис. 2.5). Получившаяся матрица будет динамически отрисовываться. В самой правой части окна – кнопки: «Подтверждение» (доступна при безошибочном вводе данных), «Отмена» и «Помощь». Это окно также содержит строку состояния, в

которую выводит ошибки, если таковые появляются по ходу ввода данных.

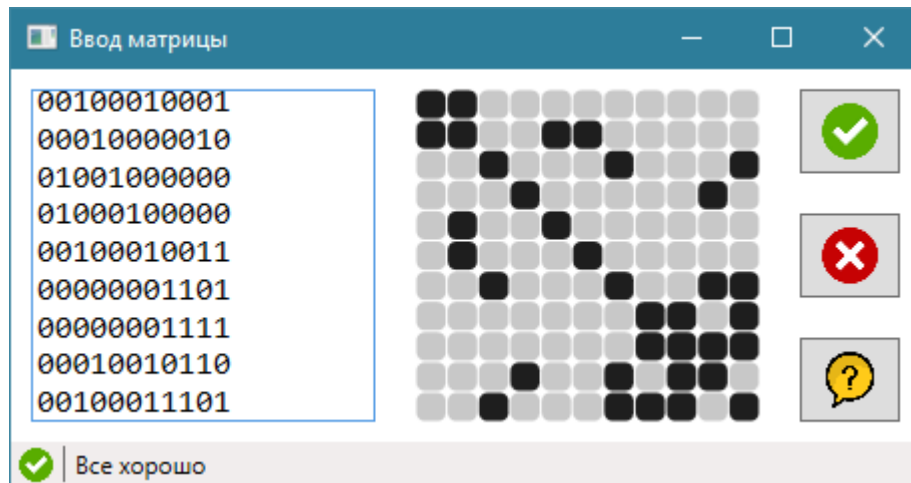


Рисунок 2.4 – Ввод матрицы смежности

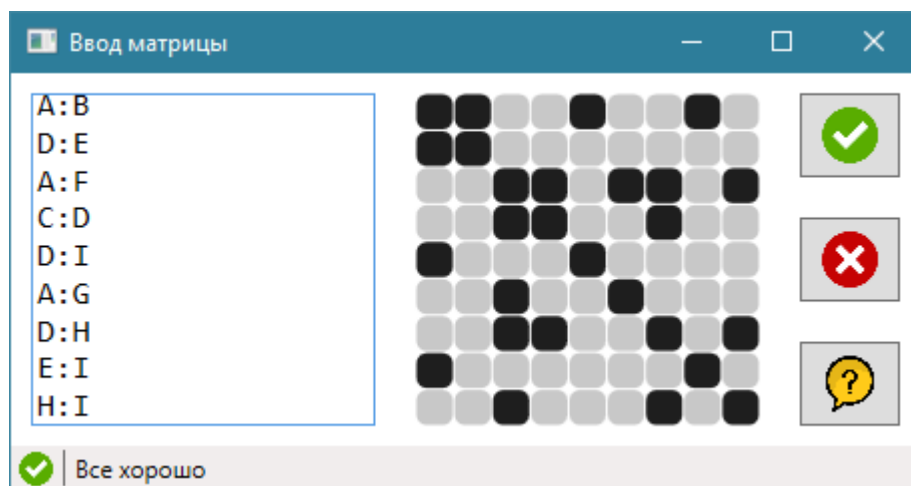


Рисунок 2.5 – Ввод пар вершин графа

Если выбрать «Сгенерировать», появится окно для ввода размерности генерируемой матрицы. Фильтр ввода работает так, что пользователь не сможет ввести букву/символ, ноль или 4-х значное число (рис. 2.6).

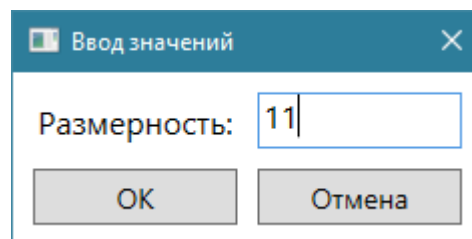


Рисунок 2.6 – Выбор размерности генерируемой матрицы

После ввода графа любым из вышеперечисленных способов в левой части окна появится его матрица смежности в виде , а в заголовке в скобках – ширина ленты матрицы. Также станет доступна кнопка «Запустить алгоритм» (рис. 2.7).



Рисунок 2.7 – Отображение матрицы

После выполнения алгоритма в правой части появится новая матрица смежности, с меньшей шириной ленты, указанной в скобках (рис. 2.8).

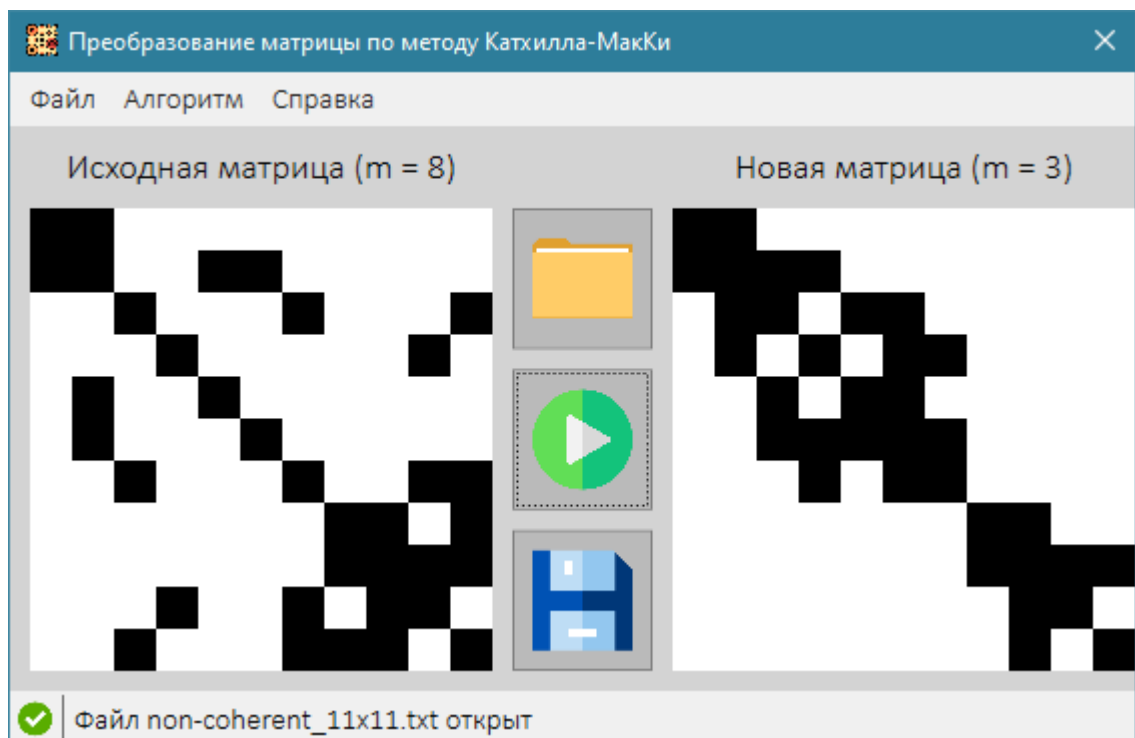
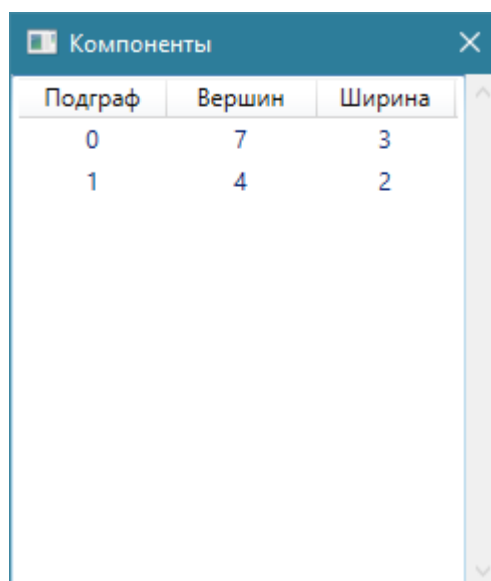


Рисунок 2.8 – Результат работы алгоритма

Программа также работает с несвязными графами, обрабатывая алгоритмом каждую компоненту связности как отдельный связный граф.

Список таких компонент можно посмотреть в меню «Алгоритм» - «Показать список компонент» (рис. 2.9).



Подграф	Вершин	Ширина
0	7	3
1	4	2

Рисунок 2.9 – Окно отображения компонент связности

Результат можно сохранить в файл (текстовый .txt или графический .bmp) через одноименную кнопку или через меню «Файл». Имя файла хранит текущую дату и время (рис. 2.10).

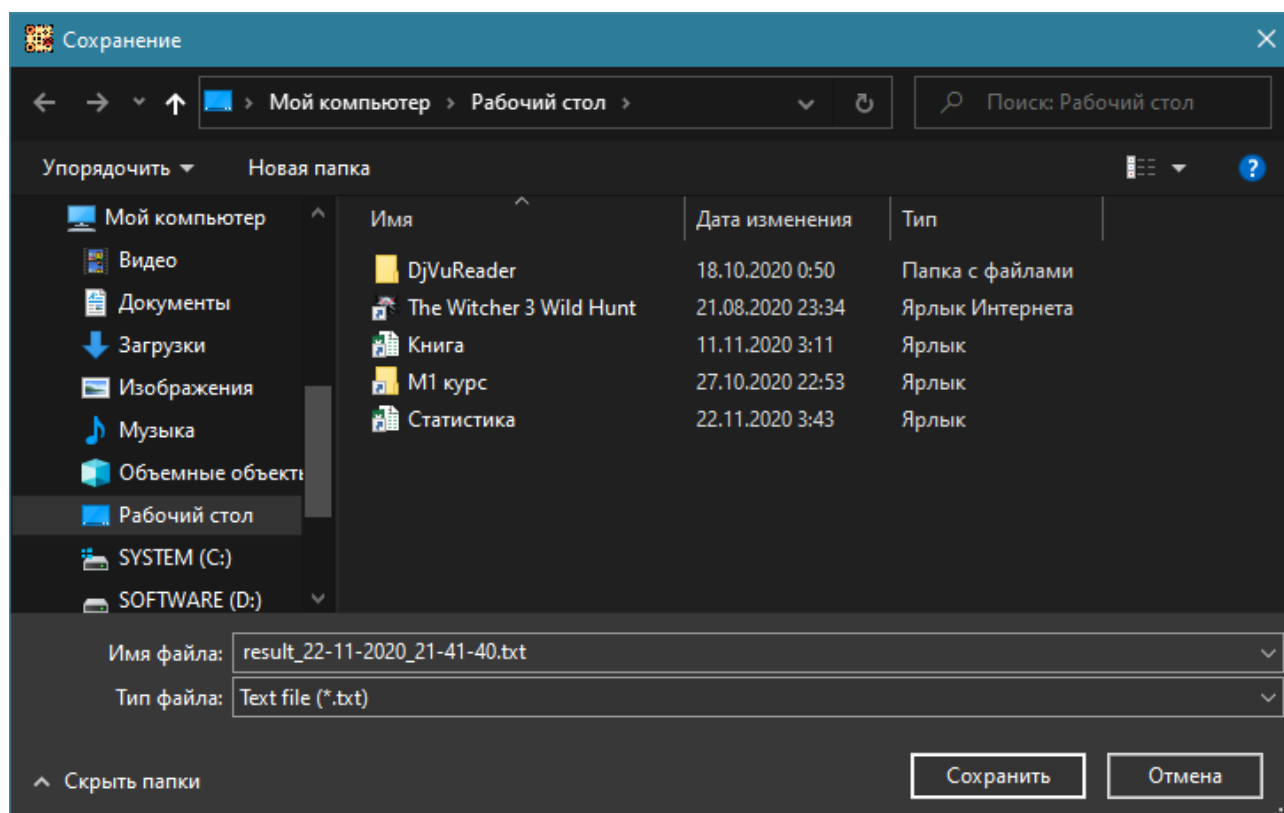


Рисунок 2.10 – Окно выбора пути для сохранения результата

После выполнения алгоритма становится доступной функция показа

протокола (лога) выполнения алгоритма (в меню «Файл»). Если ее выбрать, то появится окно (рис. 2.11). Последовательность действий выводится в текстовое поле. Кнопки справа закрывают окно, копируют лог и сохраняют его в файл соответственно.

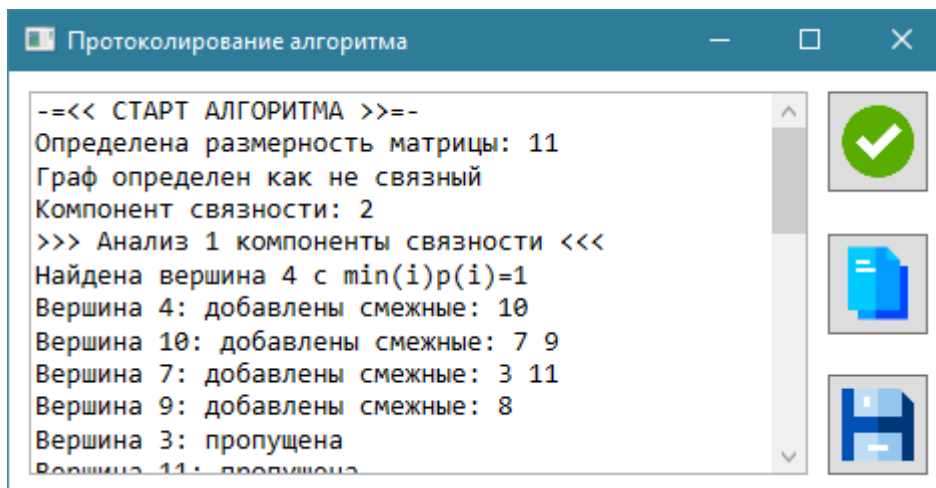


Рисунок 2.11 – Лог работы алгоритма

Программа подразумевает ввод некорректных данных и обрабатывает данные исключения. Например, файл может быть пустым (рис. 2.12).

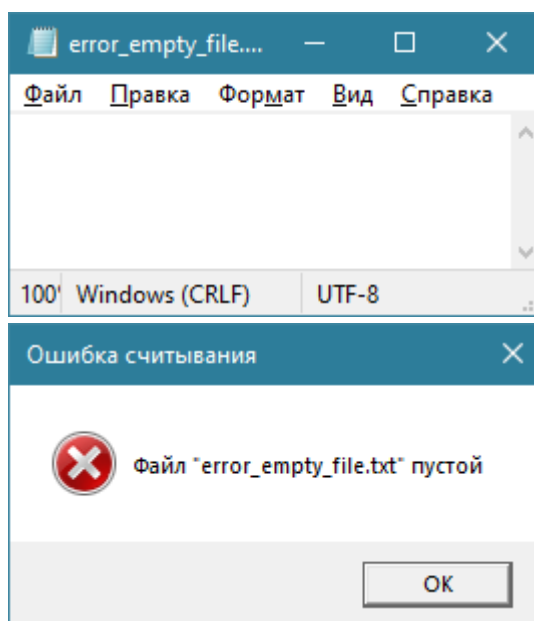


Рисунок 2.12 – Пустой файл

В считываемом программой файле-матрице может быть найден посторонний символ (например, буква или знак). Программа предусматривает эту ситуацию и выводит ошибку об этом, уточняя ряд и столбец, в котором был найден неправильный символ (рис. 2.13).

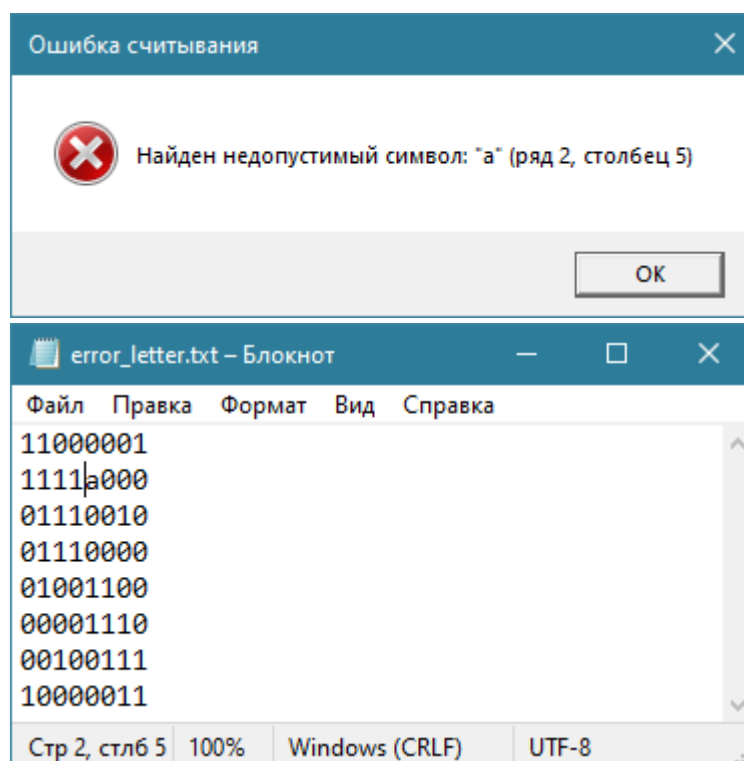


Рисунок 2.13 – Символ в строке

Если считываемые строки разной длины, такая форма записи не может считаться матрицей вовсе, и на это также выдается ошибка (рис. 2.14).

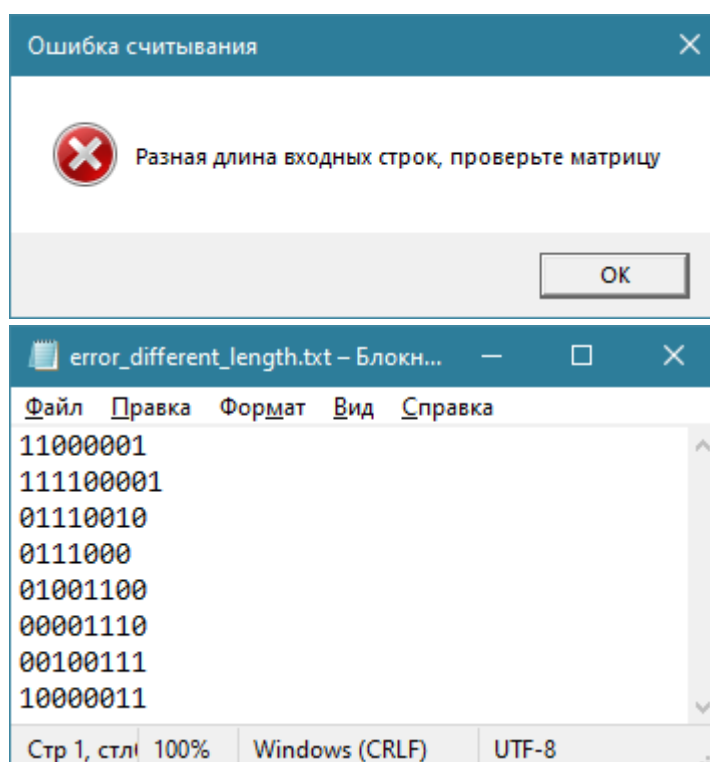


Рисунок 2.14 – Разная длина строк

В случае, если матрица прямоугольная, т. е. количество строк и столбцов не равно друг другу, данный файл тоже не подлежит считыванию (рис. 2.15).

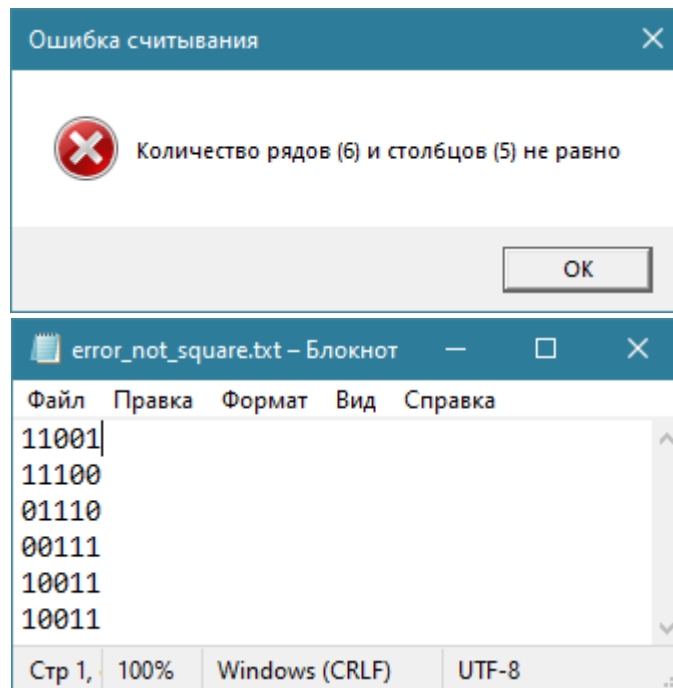


Рисунок 2.15– Прямоугольная матрица

Пункт меню «Справка» (рис. 2.16) содержит подпункты: «Открыть справку» открывает справку о программе, «О программе» открывает окно с информацией об авторе, программе и ее версии (рис. 2.17).

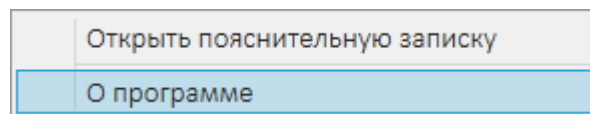


Рисунок 2.16 – Меню «Справка»

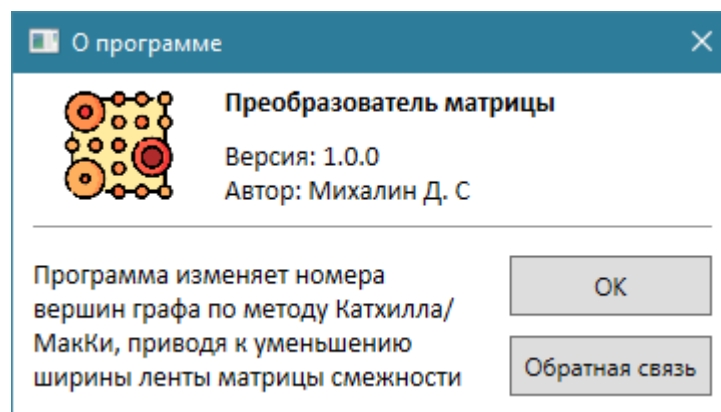


Рисунок 2.17 – Окно «О программе»

Во избежание случайного выхода из программы запрашивается простое подтверждение (рис. 2.18).

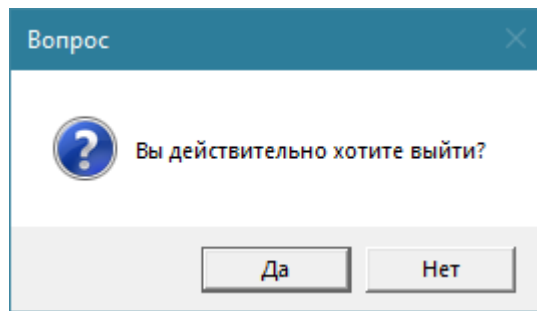


Рисунок 2.18 – Запрос подтверждения выхода

ЗАКЛЮЧЕНИЕ

В данной работе я усовершенствовал свои навыки работы с .NET Framework, а также получил практически применимые знания по алгоритмизации и работе с графами и матрицами смежности.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Тьюарсон Р. Разреженные матрицы. М.: Мир, 1977, с. 93-96. Пер изд.: Reginald P. Tewarson Sparse Matrices. Academic Press: New York and London 1973.
2. Алгоритм Катхилла — Макки [Электронный ресурс] // Википедия. URL: https://ru.wikipedia.org/wiki/Алгоритм_Катхилла_—_Макки (дата обращения: 13.10.2020)
3. Разреженная матрица [Электронный ресурс] // Википедия. URL: https://ru.wikipedia.org/wiki/Разреженная_матрица (дата обращения: 13.10.2020)

Приложение А. Диаграмма классов

