|  |  |
| --- | --- |
|  | **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» (ИУ) |

|  |  |
| --- | --- |
| КАФЕДРА | «Информационная безопасность» (ИУ8) |

Домашняя работа № 1

ПО КУРСУ

«Алгоритмические языки»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Студент | ИУ8-23 |  |  |  | М. Ю. Григорьева,  В. С. Ажгирей, |
|  | (Группа) |  |  |  | (И. О. Фамилия) |
|  |  |  |  |  |  |
| Преподаватель: |  |  |  |  | М. В. Малахов |
|  |  |  |  |  | (И.О. Фамилия) |

2024

**Введение**

Цели и задачи работы

Цель работы состоит в овладении навыками разработки графического ПО, его архитектуры и сборки. Для достижения цели необходимо выполнить следующие задачи:

1. Разработать макет графического интерфейса для приложения, разрабатываемого во втором домашнем задании.
2. Согласовать формат входных/выходных данных для приложения, разрабатываемого во втором ДЗ
3. Разработать тестовые примеры для приложения, разрабатываемого во втором домашнем задании.
4. Разработать архитектуру приложения для третьего домашнего задания (согласовать набор классов и частично методов)
5. Согласовать набор исключительных ситуаций для обработки в приложении, разрабатываемом в третьем домашнем задании.
6. Разработать тестовые примеры для приложения, разрабатываемого в третьем домашнем задании.

**Условия домашнего задания №2 для 6 варианта**

Разработайте приложение – кликер. В приложении вводится идентификатор запущенной сторонней программы (рекомендуется запускать блокнот) и текстовая строка. При запуске приложения каждый символ из заданной строки должен с паузой в 1 секунду отображаться в запущенной сторонней программе (имитация ввода с клавиатуры). По окончании ввода заданной текстовой строки необходимо вывести всплывающее окно с надписью, что операция успешно выполнена.

**Условия домашнего задания №3 для 2 варианта**

Разработайте приложение для поиска кратчайшего пути из одной вершины в другую в ориентированном взвешенном и невзвешенном (в рамках данной задачи будем считать, что вес всех ребер в невзвешенном графе равен единице) графах. Графы читаются из файла, в приложении должен быть функционал по изменению считанных графов (удалению/добавлению вершин и рёбер, изменению веса ребер для взвешенного графа).

**Графический интерфейс окон приложения**

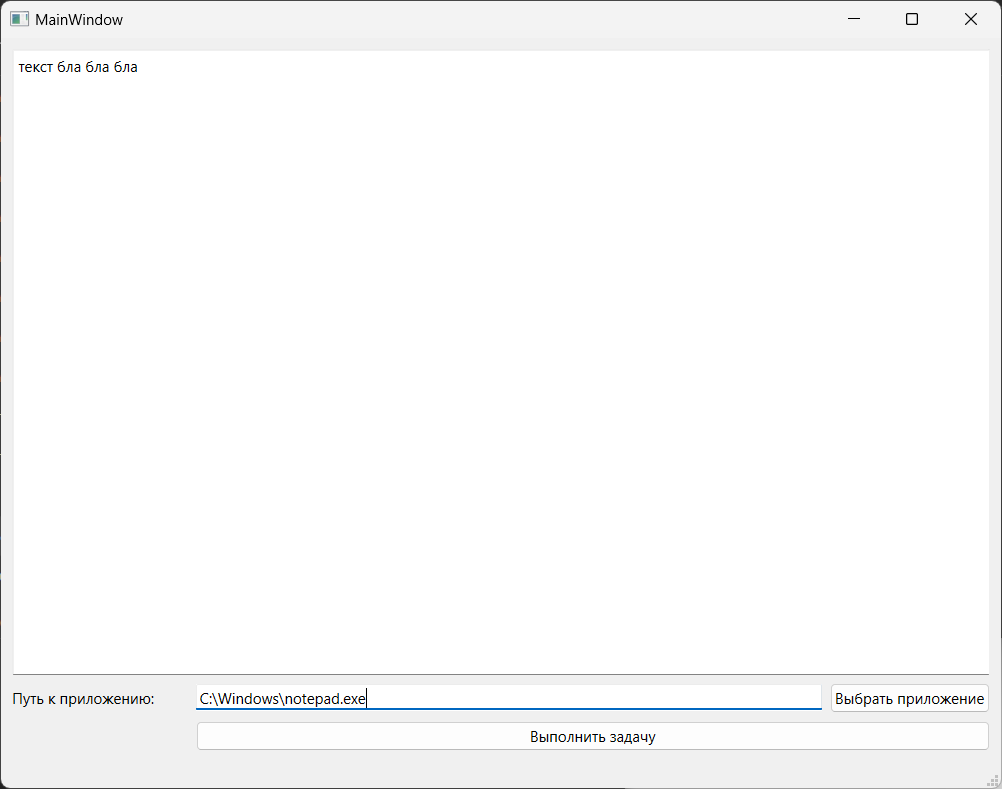
**домашнего задания №2**

Рисунок 1 – окно MainWindow (основное окно программы)

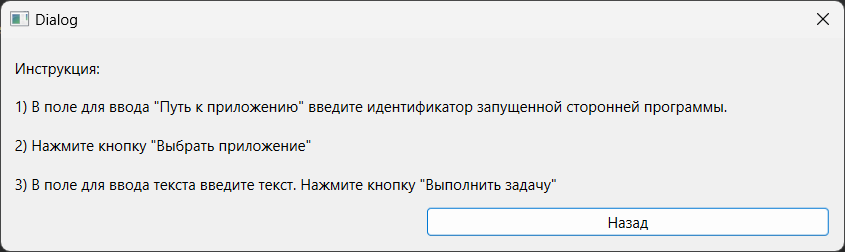


Рисунок 2 – окно Dialog (окно-справка)

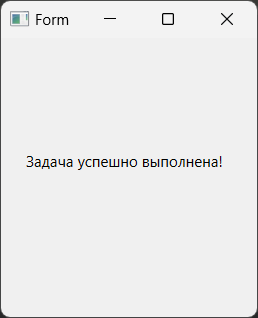


Рисунок 3 – окно Form (окно, подтверждающее успешное выполнение задачи)

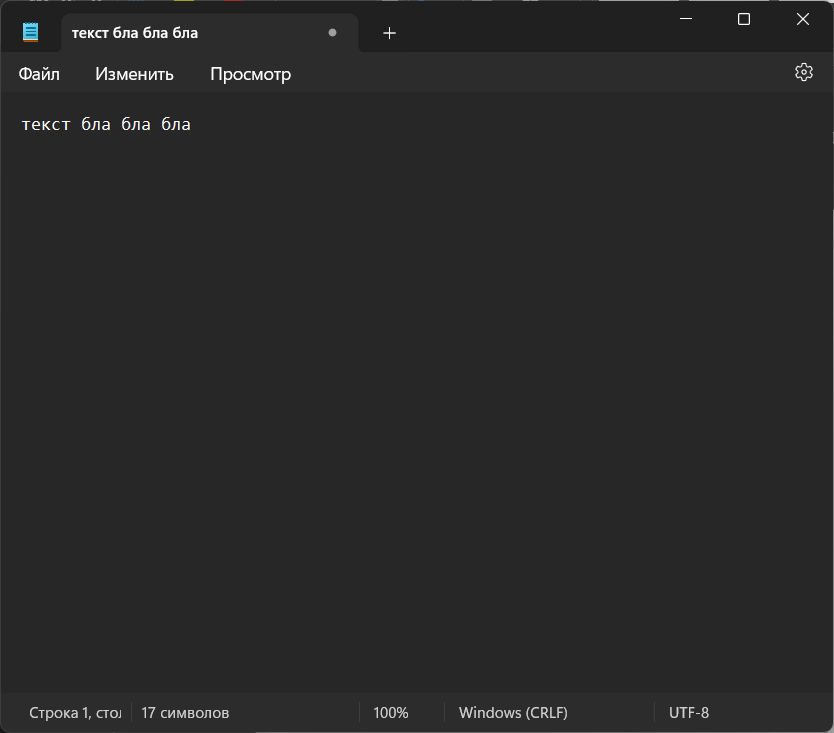


Рисунок 4 – результат работы программы

**Архитектура приложения домашнего задания №3**

В процессе разработки программы будет создано 2 класса: Graph и Vertex.

Graph – класс графа

Поля:

* vertices – массив вершин графа (объектов класса Vertex)

Методы:

* print – метод печати графа
* Dijkstra\_distance – метод нахождения кратчайшего расстояния между двумя вершинами (алгоритм Дейкстры)
* addVertex – метод добавления вершины в граф
* removeVertex- метод удаления вершины графа
* setDistanceVertices – метод добавления связи между вершинами/изменения расстояния
* removeConnectionVertices – метод удаления связи между вершинами

Vertex – класс вершины графа.

Поля:

* adjacentVertices - массив вершин (объектов класса Vertex), связных с данной

Методы:

* addConnection – добавить связь с вершиной
* removeConnection – удалить связь с вершиной
* setDistance - метод добавления связи между вершинами/ изменения расстояния

**Набор исключительных ситуаций приложения домашнего задания №3**

“Incorrect input data format” - Некорректный или непредвиденный формат входных данных (например, неправильно указанные вершины или ребра).

“Absence of a vertex or edge” - Приложение может пытаться обратиться к вершине или ребру, которых фактически не существует в графе, что может привести к ошибкам во время обработки данных.

“Impossible to find the way” - Если алгоритм поиска пути не может найти путь между двумя вершинами (например, из-за отсутствия связи).

**Тестовый пример приложения домашнего задания №3**

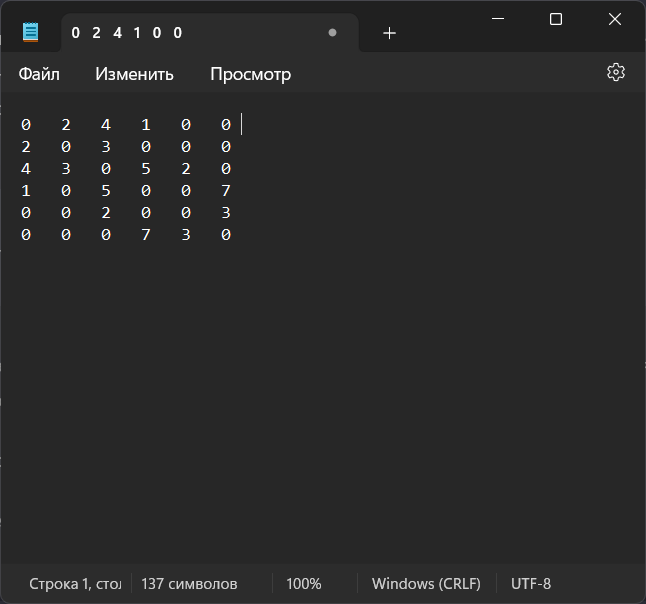


Рисунок 5 – тестовый пример 1. Ожидаемый результат: расстояние между вершиной 1 и вершиной 6 будет равно 8

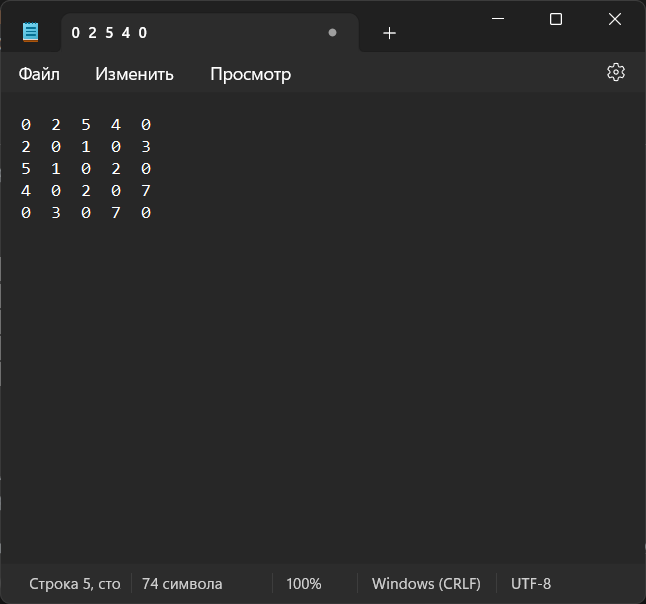


Рисунок 6 – тестовый пример 2. Ожидаемый результат: расстояние между вершиной 1 и вершиной 5 будет равно 9