**Описание проекта**

***Назначение***

*Проект представляет собой приложение для визуализации и работы с графами, разработанное в среде Microsoft Visual Studio на языке C# с использованием технологии WPF. Приложение предоставляет пользователю интерфейс для создания, редактирования и управления графами, а также различные методы взаимодействия с графами, включая сохранение и загрузку данных, работу с базой данных.*

***Основные возможности***

*1. Добавление и удаление графов: Пользователь может добавлять новые графы в визуальную область и удалять их по необходимости.*

*2. Выделение и выбор графов:*

*2.1 Одинарный клик для выделения графа.*

*2.2 Двойной клик для установки графа в качестве текущего.*

*3. Операции над графами:*

*3.1 Сложение графов: Объединение вершин и ребер двух графов.*

*3.2 Вычитание графов: Удаление совпадающих ребер и добавление обратных ребер из одного графа в другой.*

*3.3 Создание дерева кратчайших путей: Создание дерева из кратчайших путей, с выделением результирующего графа красным цветом.*

*3.4 Сохранение и загрузка графов: Сохранение графов в формате XML и загрузка их обратно в приложение.*

*4. Создание случайного графа с заданным количеством вершин*

*\*Шанс на появления ребра 65% если хочется по другому задается в поле Chance в MainWindow\**

***Дополнительные возможности***

*1. Индивидуальный выбор ребер: Возможность выбирать и задавать вес каждого ребра. Отсутствие ребра обозначается весом 0.*

*2. Ограничение на количество вершин: Максимальное количество вершин в графе - 10.*

*3. Поддержка базы данных: Состояние приложения сохраняется в базе данных при закрытии и восстанавливается при запуске.*

*4. Интерактивный интерфейс: Удобный интерфейс для работы с графами, включающий контекстные меню и подсказки.*

***Архитектура***

*1.Проект включает в себя сборку в виде файла DLL, содержащую основные классы и методы для работы с графами.*

*2.1 Graph: Основной класс для представления графа. Включает методы и свойства для работы с вершинами и ребрами.*

*2.2 EditableGraph: Класс, наследующий от Graph, добавляющий возможность редактирования графов и обработки событий.*

*3. Коллекции: Использование типизированных коллекций для хранения вершин и ребер графа.*

*4. Использование свойств для хранения данных.*

*5. Индексаторы для доступа к вершинам и ребрам.*

*6. Методы для добавления, удаления и редактирования графов.*

*7. Исключения*

*8. Обработка исключений, включая проверку на количество вершин (не более 10).*

*9. Работа с файлами*

*10. Сохранение и загрузка данных графов в формате XML.*

*11. Работа с базой данных*

*12. Сохранение состояния приложения в базу данных при закрытии и восстановление при запуске.*

*13. Использование LINQ*

*14. Использование LINQ для работы с данными графов.*

***Реализованные требования***

*1. Сборка в виде DLL.*

*2. Многокомпонентная архитектура.*

*3. Хранилище данных в виде коллекций.*

*4. Методы, свойства и индексаторы для работы с данными.*

*5. Использование исключений.*

*6. Работа с файлами и стандартными диалогами.*

*7. Работа с базой данных.*

*8. Использование LINQ для работы с данными.*

***Дополнительное описание***

*1. Выделение идет так, что синий граф является первым а красный вторым, что важно для вычитания ведь там вычитается из первого второй граф.*

*2. Вычитание и сложение мои личные операции, которые я сам придумал и возможно они не соотвествуют общепризнаным вычитаниям и сложениям, но я захотел так*

*3. В папке DataBase лежит база данных*

*4. В папке Windows лежат два окна которые я создавал:*

*4.1 DfsDialog для выбора вершины из которой идет поиск в глубину*

*5.1 ActionsMenuDialog для выбора действия над графом*

*6. Класс EditableGraph создан как наследник Graph и только они идут в поле графов, потому что у них есть события при нажатии которые я обрабатываю, класс Graph используется в Текущем графе, там эти событиия не нужны и его использовать предпочтительнее*

*7. Dll Graph содержит сам класс граф*

*7.1 А также дополнительный класс ArrowLine для направленной стрелки от вершины к вершине*

*7.2 А также дополнительный класс GraphUtils для работы с графами, в самом классе описаны методы которые можно применять для графов и как они работают*

***В программе я старался описать всё через <summary> надеюсь всё будет понятно***