ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ

БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ –

ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

НИЖЕГОРОДСКИЙ ФИЛИАЛ

ФАКУЛЬТЕТ БИЗНЕС-ИНФОРМАТИКИ И

ПРИКЛАДНОЙ МАТЕМАТИКИ

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**к курсовой работе:**

**«**Визуальное редактирование графов**»**

**Выполнил:** студент группы 10ПМИ2

Савиных А.Л.

**Руководитель:** Логинов О.В.

Нижний Новгород

2013

Оглавление

[Введение 3](#_Toc348975762)

[История 4](#_Toc348975763)

[Аналоги 5](#_Toc348975764)

[Задача 6](#_Toc348975765)

[Требования 6](#_Toc348975766)

[Разработка 6](#_Toc348975767)

[Средства разработки 6](#_Toc348975768)

[Архитектура 7](#_Toc348975769)

[Реализация 8](#_Toc348975770)

[Эксплуатация 11](#_Toc348975771)

[Добавление элемента: 11](#_Toc348975772)

# Введение

Темой данной курсовой работы является визуальное редактирование графов.

Графы как структуры данных распространены и используются для представления и моделирования различных объектов. Практически любая задача, в которой важны объекты и связи между ними, может быть представлена в виде графа. Вот несколько примеров:

* Идеи, приходящие сотрудникам во время совещания, зарисовываются в виде графов, идеи – вершины графа, связанные по смыслу идеи соединяют линиями – рёбрами графа
* Веб-сайты с различными страницами и ссылками между ними могут быть представлены в виде графов. Страницы – вершины графа, ссылки на другие страницы – рёбра графа.
* Компьютерные сети могут быть представлены в виде графов
* Карты земной поверхности с проложенными по ней линиями коммуникации могут быть представлены в виде графов

Графы воспринимаются проще и эффективнее когда представлены визуально. В то же время, это не единственный способ их представления. Например:

Таблица 1

|  |  |
| --- | --- |
| Визуальное представление | Текстовое представление |
|  | Вершины: 2, 3, 1, 4, 5  Связи: 3 соединено с 1, 1 соединено с 2, 1 соединено с 4, 1 соединено с 5 |

Для более эффективной и удобной с точки зрения программиста обработки графы как структуры данных хранятся часто в виде наборов вершин и связей между ними, имеющем много общего с тем, как представлено в колонке «текстовое представление» в таблице Таблица 1 выше. В тех случаях, когда, например, человеку необходимо ввести подготовить исходные данные в виде являющиеся графом для последующей обработки программой или же просто для хранения в цифровом виде, требуется визуальное представление графа и его последующее преобразование в пригодный для программной обработки формат.

Программа, разработанная в рамках курсовой работы, позволяет более эффективно и удобно хранить графы в виде файлов, а так же изменять их в дальнейшем и обрабатывать с помощью иных программ.

# История

Родоначальником теории графов считается Леонард Эйлер. В 1736 году в одном из своих писем он формулирует и предлагает решение задачи о семи кёнигсбергских мостах, ставшей впоследствии одной из классических задач теории графов.

# Аналоги

Существует множество программ выполняющих аналогичные задачи. Вот описание одной из них в качестве примера: приложение XMind имеет платную и бесплатную версии, предоставляет возможности для визуального редактирования графов. На рисунке Рисунок 1 показана схема-граф, созданная с помощью приложения XMind.

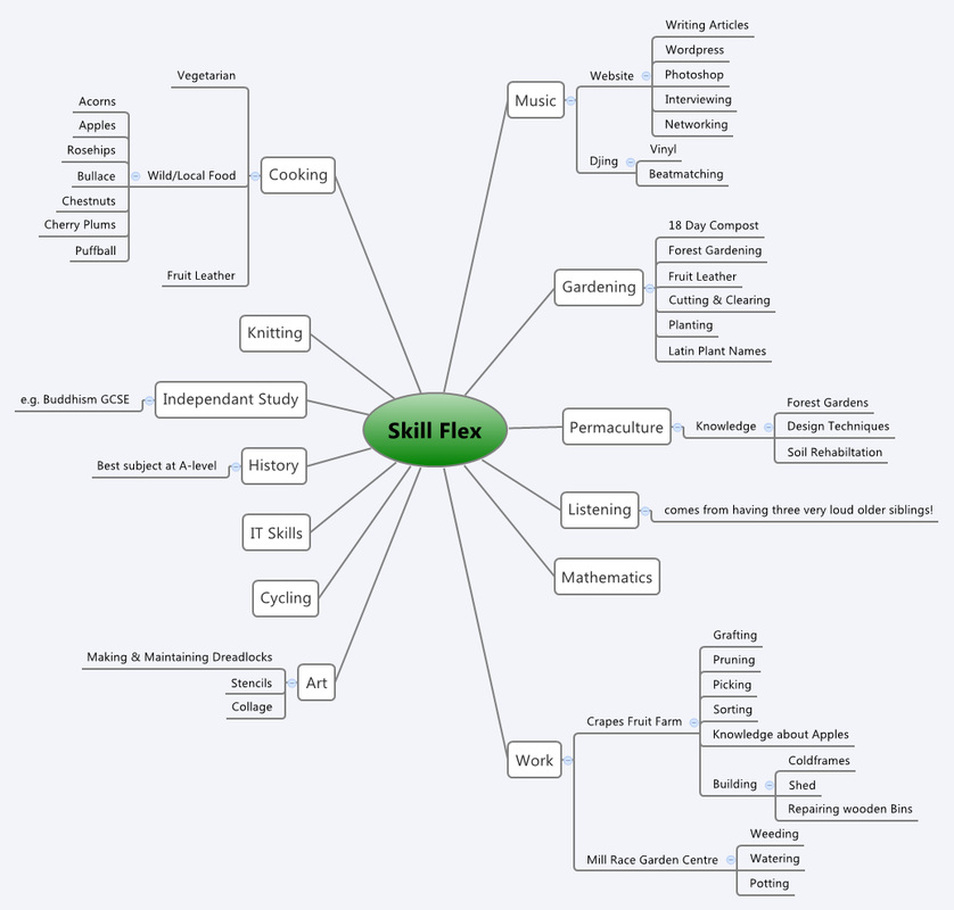


Рисунок 1

XMind может подбирать позиции узлов таким образом, чтобы они выглядели естественно. Так, если пользователь явно не указывает расположение дочерних узлов, они располагаются по кругу возле центрального узла в случае узлов, непосредственно связанных с центральным узлом («Skell Flex» на рисунке Рисунок 1) либо на свободном от других узлов и связей месте в случае более удалённых от центрального узла узлов. XMind сохраняет документы в формате XML, так что они могут быть использованы другими приложениями. При создании документа в XMind возможно изменять цвет, размер и начертание шрифтов подписей узлов, добавлять изображения к узлам, изменять положение узлов, изменять фигуры обрамляющие узлы, добавлять дополнительный текст к узлам, а также цвет и формат прочих элементов.

# Задача

В ходе подготовки к курсовой работе была сформулирована следующая задача: создать приложение для визуального редактирования графов, обладающее некоторыми особенностями. Одной из особенностей должна быть возможность сохранения графов в формат, пригодный для использования другими приложениями.

# Требования

Все требования разделены на две группы: обязательные требования, отмеченные в данном документе закрашенными кружками, и необязательные требования, отмеченные в данном документе незакрашенными кружками.

* Возможность добавления новых узлов в редактируемый граф
* Возможность присоединения подписей к узлам графа
* Возможность редактирования подписи к определённому узлу графа
* Возможность просмотра графов с помощью приложения
* Возможность просмотра графа по частям, прокручивая документ, в случае если граф имеет большую площадь
* Возможность удаления существующих узлов из редактируемого графа
* Возможность соединения узлов связями
* Возможность удаления связей между узлами
* Возможность сохранения редактируемого графа в файл
* Возможность загрузки графа из файла, в который он ранее был сохранён тем же приложением
* Возможность изменения положения узлов в документе
* Возможность работы с несколькими графами одновременно так, чтобы открытые графы были представлены в интерфейсе пользователя вкладками
* Возможность присоединения меток к узлам графа
* Возможность управления приложением с помощью жестов

# Разработка

## Средства разработки

Для написания приложения была выбрана среда разработки Visual Studio Express 2012 для Windows, язык программирования C#. Упомянутая среда разработки является бесплатной для некоммерческого использования. Для работы с отладочными сообщениями использована библиотека NLog. Для хранения исходного кода проекта и документации выбрана система контроля версий Git. Для работы с системой контроля версий использовано приложение SmartGit, представляющее собой графический интерфейс для Git и некоторых других систем контроля версий. Исходный код проекта хранится также на сайте GitHub, предоставляющем возможность хранить репозитории Git с исходным кодом онлайн. Благодаря этому в случае отказа жёсткого диска компьютера, на котором разрабатывался проект, можно было бы скачать репозиторий с сайта GitHub и продолжить работу. Также использованы различные средства и библиотеки, входящие в состав Visual Studio 2012: Windows Presentation Foundation для разработки пользовательского интерфейса, система управления пакетами NuGet, средства для работы с XML и прочее.

## Архитектура

Так как проект разрабатывался на языке программирования C#, использован объектно-ориентированный подход, основными строительными блоками проекта являются классы. Для удобства все классы разбиты на несколько категорий. Ниже слово проект используется в смысле проекта VisualStudio как объединения классов, слово решение используется в смысле решения VisualStudio как объединения нескольких проектов.

На рисунке «Рисунок 2: Структура решения» представлена структура решения.

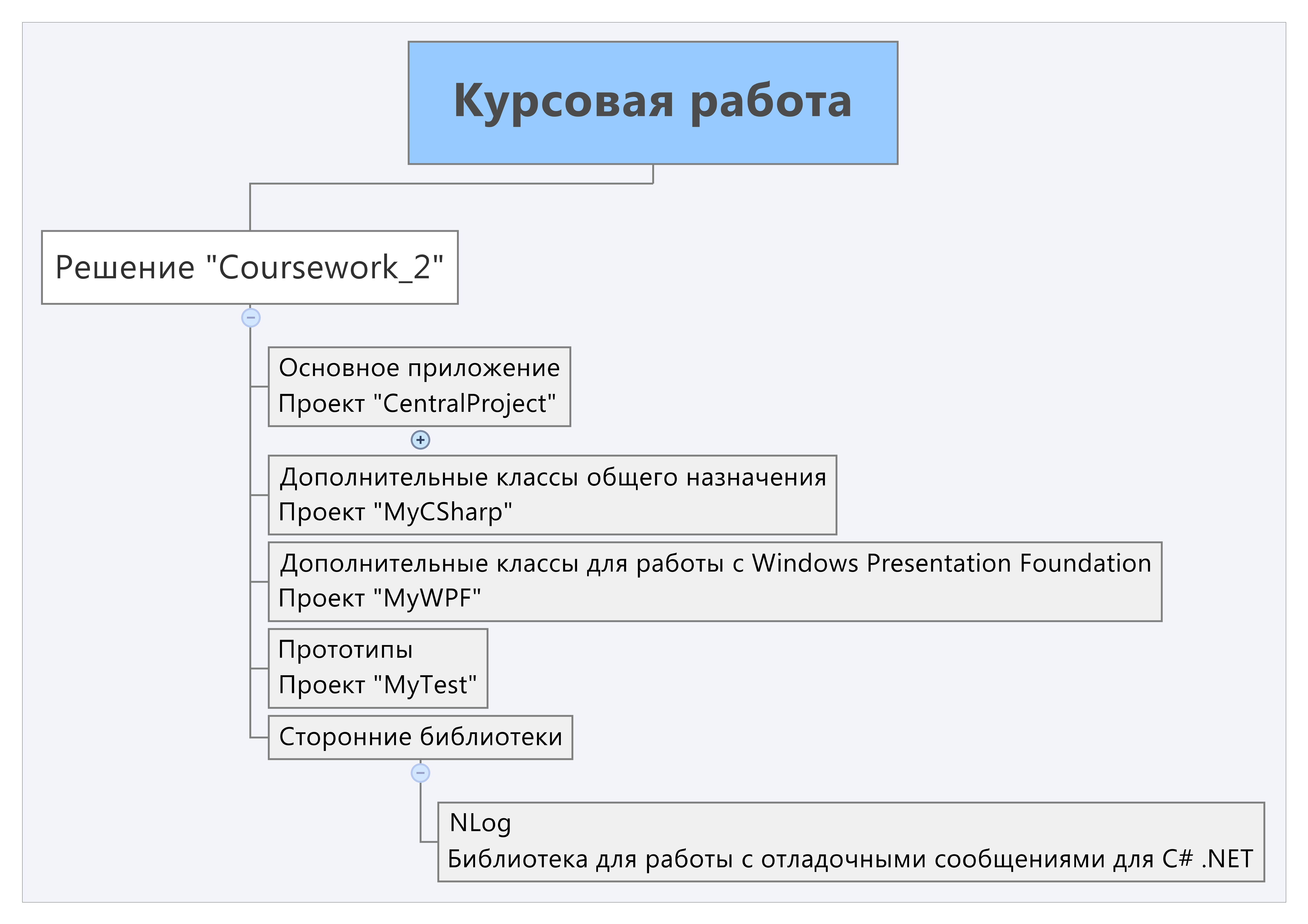


Рисунок 2: Структура решения

В проекте «CentralProject» собраны классы, непосредственно относящиеся по смыслу к приложению для визуального редактирования графов. Все классы в основном проекте тесно связаны между собой и не обладают большой самостоятельностью. В остальных проектах содержатся различные вспомогательные классы. В приложении явно не создаются никакие дополнительные потоки, большую роль играет Windows Presentation Foundation, который и определил архитектуру приложения, код приложения привязан к событиям Windows Presentation Foundation и выполняется полностью в том же потоке, что и обработчик этих событий, отвечающий за пользовательский интерфейс.

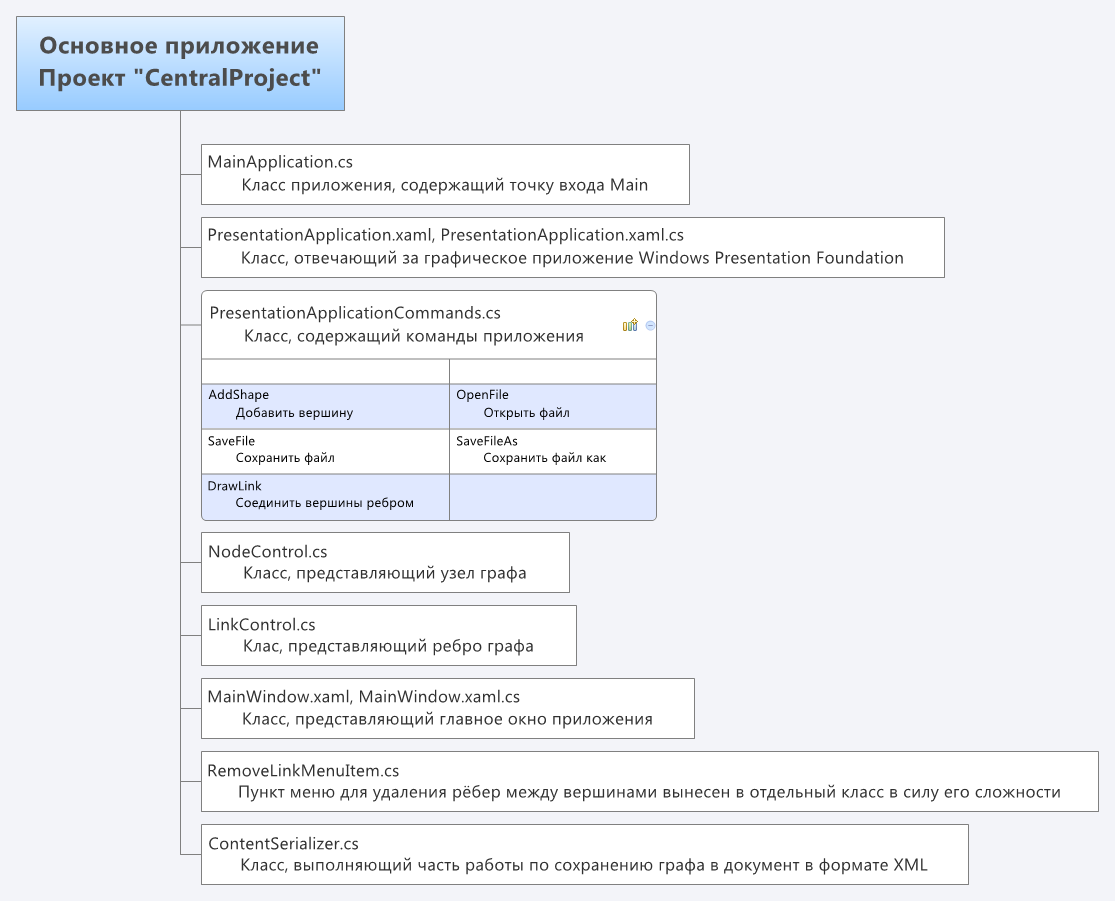


Рисунок 3: CentralProject

Некоторые классы определены как с помощью исходного кода на языке C# в файлах с расширением .cs, так и с помощью исходного кода на языке XAML в файлах с расширением .xaml, которые используются главным образом при разработке приложений Windows Presentation Foundation для определения пользовательского интерфейса.

## Реализация

В самом начале своей работы программа передаёт управление приложению Windows Presentation Foundation:

var presentation = new PresentationApplication();

presentation.Run();

Метод Run вызывает обработку событий пользовательского интерфейса и работает, пока главное приложения окно не будет закрыто. Затем графическое приложение будет уничтожено.

В файле PresentationApplication.xaml указан класс главного окна приложения:

<Application x:Class="Coursework\_2.PresentationApplication"

… … …

**StartupUri="MainWindow.xaml"**

>

На снимке экрана «Рисунок 4: Окно приложения» представлено главное окно приложения с открытым документом.

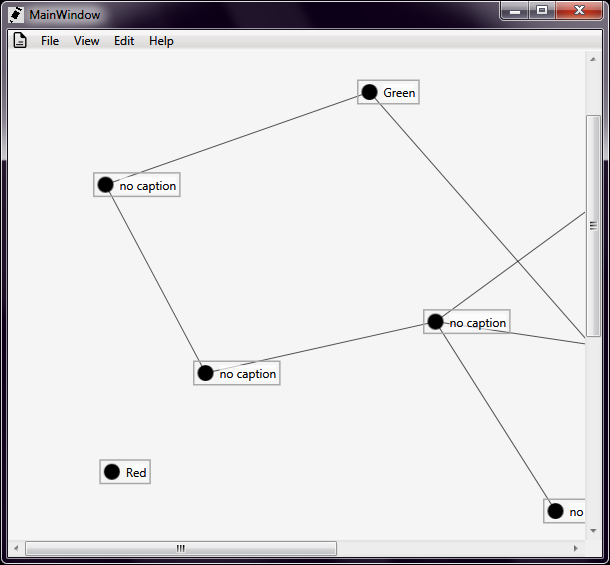


Рисунок 4: Окно приложения

В файле разметки MainWindow.xaml определены пункты главного меню, пиктограммы для них, а также область документа.

Ниже представлен фрагмент кода из файла MainWindow.xaml:

<MenuItem Header="File" Icon="{StaticResource Image\_File}">

<MenuItem

Header="Open file..."

x:Name="OpenFileMenuItem"

Icon="{StaticResource Image\_OpenFile}"

/>

<MenuItem

Header="Save file"

x:Name="SaveFileMenuItem"

Icon="{StaticResource Image\_SaveFile}"

/>

… … …

Ниже представлен фрагмент кода, привязывающий команды и их обработчики к пунктам главного меню окна приложения.

protected void BindCommands()

{

BindCommand(OpenFileMenuItem, Commands.OpenFile, UserOpenFile);

BindCommand(SaveFileMenuItem, Commands.SaveFile, UserSaveFile);

BindCommand(SaveAsFileMenuItem, Commands.SaveAsFile, UserSaveAsFile);

BindCommand(AddItemMenuItem, Commands.AddShape, UserAddItem);

BindCommand(AddLinkMenuItem, Commands.DrawLink, UserAddLink);

}

protected void BindCommand(MenuItem item, RoutedUICommand command,

ExecutedRoutedEventHandler handler)

{

CommandManager.RegisterClassCommandBinding(

this.GetType(),

new CommandBinding(command, handler)

);

item.Command = command;

}

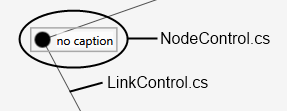


Рисунок 5

На рисунке «Рисунок 5» показано, каким видимым элементам управления в пользовательском интерфейсе соответствуют классы NodeControl и LinkControl.

Ниже представлен фрагмент документа в формате XML, полученного с помощью программы визуального редактирования графов.

<?xml version="1.0" encoding="utf-8"?>

<content>

<NodeControl Text="no caption" Left="629" Top="731" id="1" />

<NodeControl Text="no caption" Left="374" Top="617" id="2" />

<NodeControl Text="no caption" Left="703" Top="536" id="3" />

<NodeControl Text="no caption" Left="535" Top="551" id="4" />

<NodeControl Text="no caption" Left="140" Top="737" id="5" />

<NodeControl Text="Some Other Node" Left="185" Top="413" id="6" />

<NodeControl Text="no caption" Left="875" Top="430" id="7" />

<NodeControl Text="no caption" Left="415" Top="361.60526315789468" id="8" />

<NodeControl Text="SomeNode" Left="85" Top="224.60526315789471" id="9" />

<NodeControl Text="no caption" Left="677" Top="169.60526315789471" id="10" />

<NodeControl Text="Green" Left="349" Top="132" id="11" />

<NodeControl Text="Red" Left="91.331466965285514" Top="511.6973684210526" id="12" />

<LinkControl Node1="11" Node2="9" id="13" />

<LinkControl Node1="11" Node2="3" id="14" />

<LinkControl Node1="10" Node2="8" id="15" />

<LinkControl Node1="8" Node2="6" id="16" />

<LinkControl Node1="9" Node2="6" id="17" />

<LinkControl Node1="8" Node2="4" id="18" />

<LinkControl Node1="7" Node2="8" id="19" />

<LinkControl Node1="2" Node2="5" id="20" />

</content>

Содержимое документа заключено в корневом элементе XML-документа “content”. В нём содержатся элементы “NodeControl”, соответствующие вершинам графа, и элементы “LinkControl”, соответствующие рёбрам графа. Каждый элемент “NodeControl” содержит следующие атрибуты:

* “Text”: подпись вершины
* “Left”, “Top”: расстояния от левой и верхней границ документа, выраженные в единицах измерения расстояний Windows Presentation Foundation, соответствующих одному пикселю на экране с разрешением 96 пикселей на дюйм.
* “id”: уникальный в рамках одного документа номер элемента

Каждый элемент LinkConrol соответствует ребру графа и содержит следующие атрибуты:

* “Node1” и “Node2”: уникальные номера вершин, связываемых данным ребром и содержащиеся в их атрибутах “id”.
* “id”: уникальный в рамках одного документа номер элемента

Таким образом, из документа, генерируемого приложением визуального редактирования графов можно полностью восстановить структуру графа и, при необходимости, его внешний вид при редактировании.

В процессе реализации приложения были удовлетворены все обязательные требования, описанные выше в секции «Требования», но не было удовлетворено ни одно из необязательных требований.

# Эксплуатация

Ниже приведены инструкции для выполнения различных действий в приложении в том порядке, в котором приведены соответствующие этим действиям требования.

## Добавление вершины

* Выбрать пункт меню Edit → Add Item, как показано на рисунке «Рисунок 6»

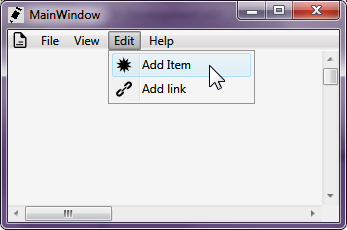


Рисунок 6

При этом если навести мышь на область документа, указатель изменится, как показано на рисунке «Рисунок 7».

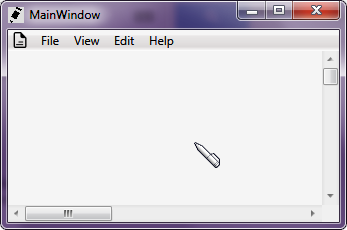


Рисунок 7

* Нажать левой кнопкой мыши в той области документа, в которой необходима новая вершина графа. При этом появится окно, в котором будет предложено ввести имя для вершины, как показано на рисунке «Рисунок 8»
* Для отмены добавления вершины на этом шаге следует нажать правой кнопкой мыши в тот момент, когда указатель принял форму авторучки. При этом указатель примет первоначальную форму.

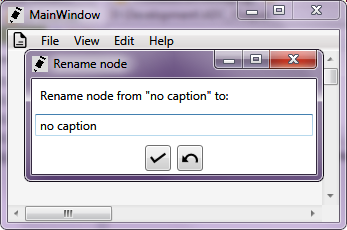


Рисунок 8

* Ввести имя вершины и нажать кнопку C:\Users\hinst\Docs\Pro\MyCSharp\MyWPF\Images\SubmitBlack16.png для подтверждения, как показано на рисунке «Рисунок 9», либо кнопку C:\Users\hinst\Docs\Pro\MyCSharp\MyWPF\Images\UndoBlack16.png для отмены.

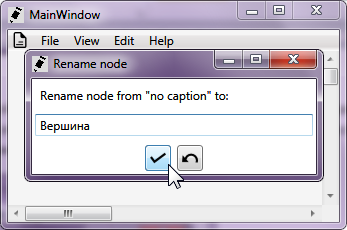


Рисунок 9

* В случае подтверждения в документе появляется новая вершина, как показано на рисунке «Рисунок 10».

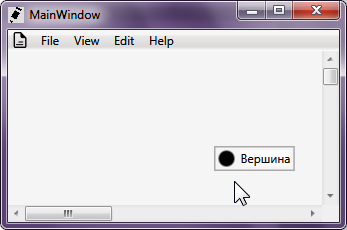


Рисунок 10

## Редактирование подписи

* Для того чтобы отредактировать подпись вершины графа, следует вызвать её контекстное меню путём нажатия левой кнопкой мыши по пиктограмме вершины графа C:\Users\hinst\Docs\Pro\Coursework_2\CentralProject\Images\BulletShapeBlack16.png и выбрать пункт меню Rename item, как показано на рисунке «Рисунок 11».

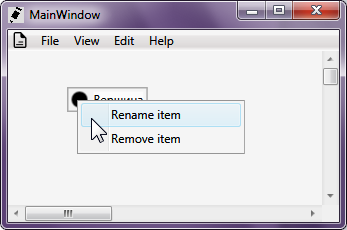


Рисунок 11

* При этом отобразится окно, в котором можно изменить имя вершины и затем подтвердить изменение [C:\Users\hinst\Docs\Pro\MyCSharp\MyWPF\Images\SubmitBlack16.png] либо отменить изменение [C:\Users\hinst\Docs\Pro\MyCSharp\MyWPF\Images\UndoBlack16.png], при этом в последнем случае, что бы ни находилось в поле ввода для подписи вершины в окне редактирования имени вершины, окно редактирования подписи вершины скроется, и подпись вершины останется прежней.

## Просмотр графа

* Чтобы просматривать различные части документа с графом, следует воспользоваться полосами прокрутки

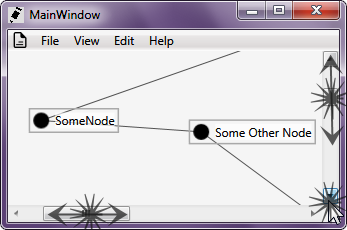


Рисунок 12