**Министерство науки и высшего образования Российской Федерации Пермский институт (филиал)**

**Федерального государственного бюджетного образовательного учреждения высшего образования**

**«Российский экономический университет имени Г.В. Плеханова»**

Факультет учетно-финансовый

Кафедра экономического анализа и статистики

**КОНТРОЛЬНАЯ РАБОТА №1**

**по дисциплине «Программирование в операционных системах с графическим интерфейсом»**

**на тему** «**Приложение для просмотра содержимого фалов \*.jpg, используя технологию WPF»**»

Выполнил студент группы ПР-31

очной формы обучения

учетно-финансового факультета

Давыдов Олег Александрович

Подпись \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г.

Научный руководитель

Берестов Дмитрий Борисович,

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подпись \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2023г.

Пермь 2023

**Содержание**

[ВВЕДЕНИЕ 3](#_Toc128536866)

[Реализация 4](#_Toc128536867)

[MainWindow.xaml 4](#_Toc128536868)

[GetImage.cs 6](#_Toc128536869)

[Поиск файлов 8](#_Toc128536870)

[Вставка изображения и очистка 9](#_Toc128536871)

[Загрузка стилей 11](#_Toc128536872)

[Дополнительный функционал 13](#_Toc128536873)

## ВВЕДЕНИЕ

**Цель исследования**: разработка и реализация на языке C#, используя технологию WPF алгоритмов решения задач, представленных в задании контрольной работы.

**Объект исследования:** контрольной работы являются методы и технологии разработки настольных программ на базе WPF XAML.

**Предмет исследования:** технологии разработки программы для просмотра изображений в формате .jpg на WPF XAML, методы обработки списков, файлов, строк.

**Задача:** разработать и реализовать приложение для просмотра содержимого фалов \*.jpg, используя технологию WPF на базе XAML.

Должен быть дополнительно реализован класс с изображениями, хранящий в себе свойства путь, и имя. А также присутствовать элемент «GridSplitter».

Дополнительные задачи:

1. Реализовать меню с элементами «MenuItem»
2. Использовать стили формата \*.xaml

## Реализация

## MainWindow.xaml

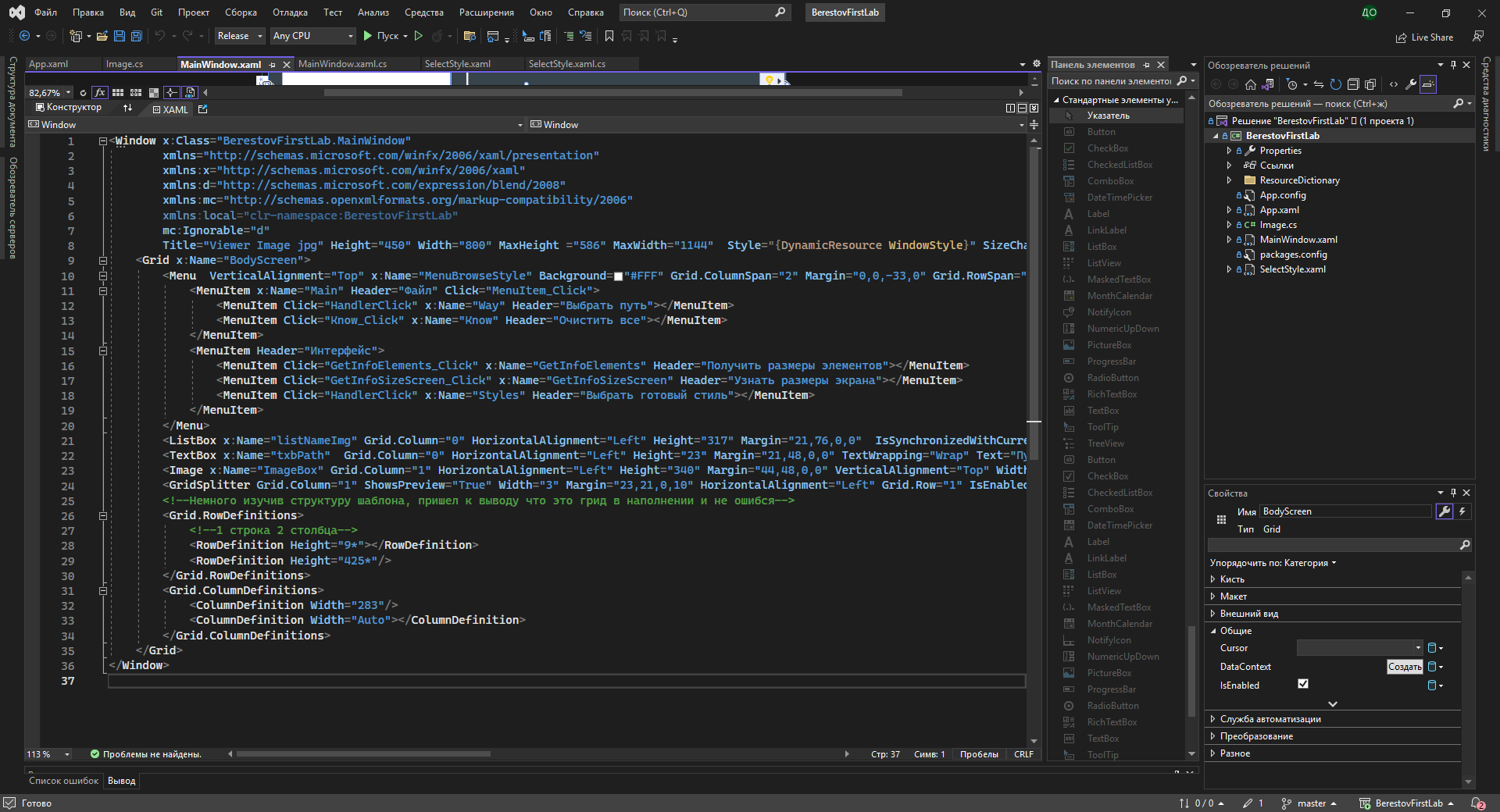
Для создания приложения на базе WPF XAML, был создан проект «[BerestovFirstLab](https://github.com/Oleja2018/BerestovFirstLab)» и по окончанию работ залит на ресурс GITHUB. Первым делом нужно было создать разметку страницы, на основе технологии XAML. Для изучения технологии было принято решение, самостоятельно воссоздать структуру файла.

Рисунок 1.Структура документа MainWindow.xaml.

В структуре используется элемент таблицы «Grid», разделённый на строку и столбцы. В таблице находятся элементы: «Menu», «ListBox», «Image». Элемент «Menu» включает два элемента MenuItem «Файл», «Интерфейс»

MenuItem «Файл» состоит из двух компонентов:

1. «Выбрать путь», открывает диалоговое окно файловой системы, выбирает нужный каталог. Инициализирует вспомогательные методы, и выводит список изображений в список, и выводит выбранный элемент.
2. «Очистить все», очищает коллекцию и выбранное изображение.

MenuItem «Интерфейс» состоит из трех компонентов:

1. «Получить размеры элементов», выводит информацию о размерах элементов: txtpath(Textbox.Text), listNameImg(ListBox), ImageBox(Image). Использует функцию «MessageBox.Show()», и обработчик событий.
2. «Узнать размеры экрана», выводит размер приложения. Использует встроенные методы определения высоты и ширины(\*.AbsoluteHeight, \*AbsoluteWidth).
3. «Выбрать готовый стиль», инициализирует новое окно «SelectStyle», в котором находится элемент «ComboBox», дающий выбор используемого стиля.

Элемент «ListBox» хранит в себе объекты типа «GetImage» (Image), полученные рукописными процедурами. Отображает их имена благодаря свойству синхронизации. Название класса «GetImage» было вынужденной мерой, так как пространство имен стандартных пакетов хранит в себе много разновидностей Image, и схожих названий.

Элемент «Image» хранит в себе изображение, полученное по свойствам объекта «GetImage».

## GetImage.cs

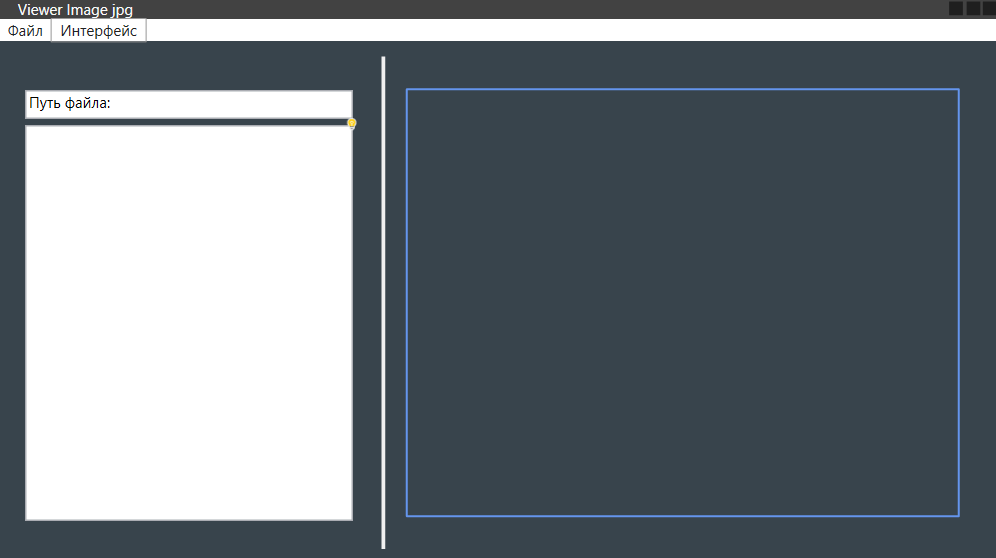
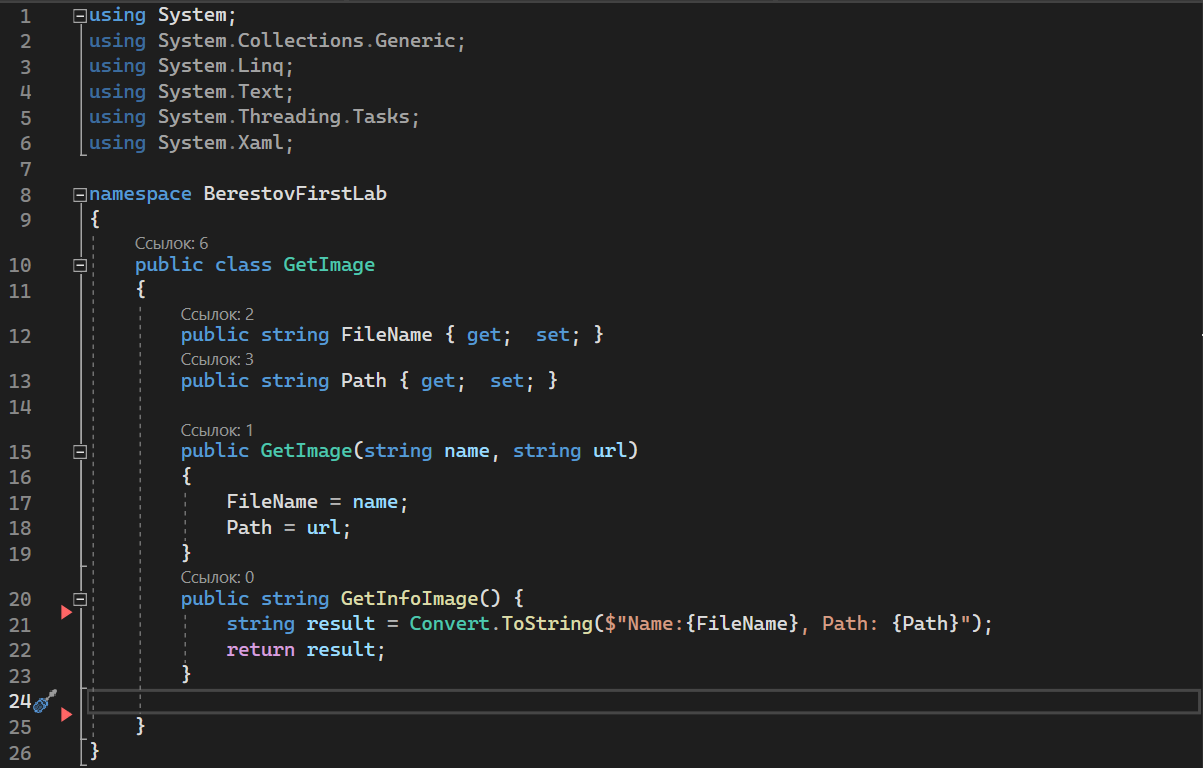
Для работы нам понадобится класс «GetImage», хранит свойства: путь файла (Path), имя файла (FileName) свойства используют {get;set;}.Также существует конструктор класса для определения свойств, и метод возвращающий

Рисунок 3. Класс GetImage

Рисунок 2. Структура части документа графически MainWindow.xaml.

MainWindow.xaml.cs

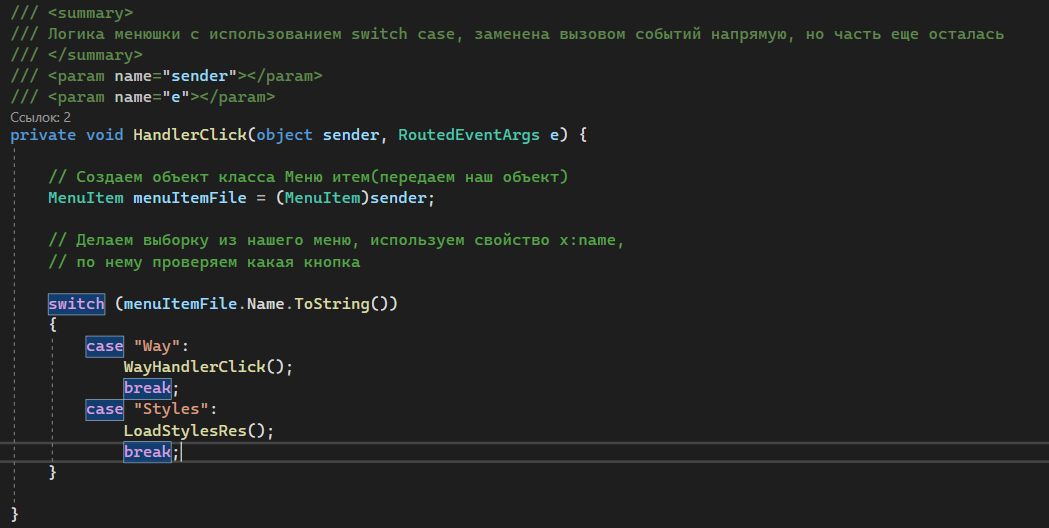
Логика главной страницы, определяется в файле «MainWindow.xaml.cs». Для работы с меню использовались процедуры. «HandlerClick» для работы под элементов. Используется конструкция «Switch case break», что позволяет вызывать разные процедуры.

Рисунок 4. Процедура HandlerClick

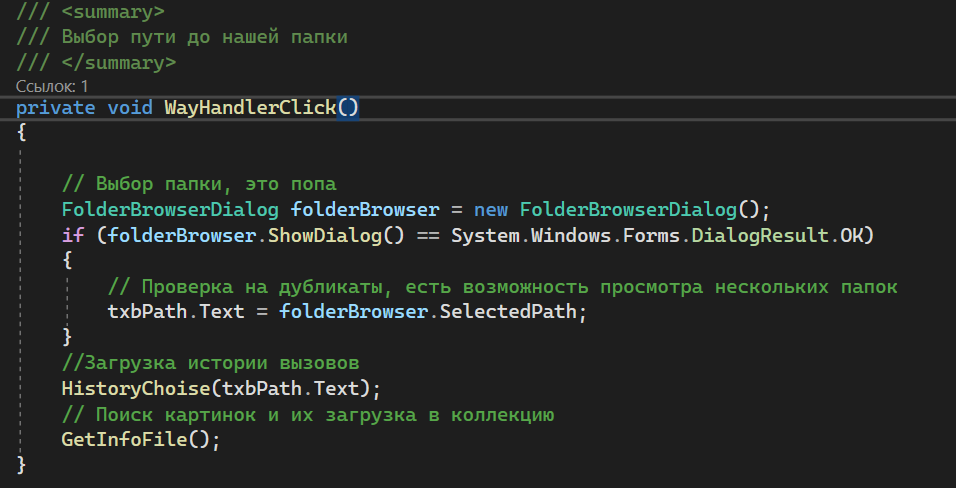
«Выбрать путь» инициализируется процедура WayHandlerClick, с помощью класса DirectInfo, танцев с бубном, и использованием классаWin.Forms. В качестве аргумента для передачи используется «txbPath.Text», что позволяет в будущем произвести модификацию программы. Ручной ввод, но как показала практика, он не актуален, и неудобен. К тому же, он уже реализован в выборе папки.

Рисунок 5. Процедура WayHandlerClick

Далее используется самописная функция записи истории, что не обязательно, но реализовано для последующих модификаций возможностей софта. Потом инициализируется метод «GetInfoFile» суть которой, заполнение коллекции объектов GetImages, объектами GetImage и загрузки данных.

## Поиск файлов

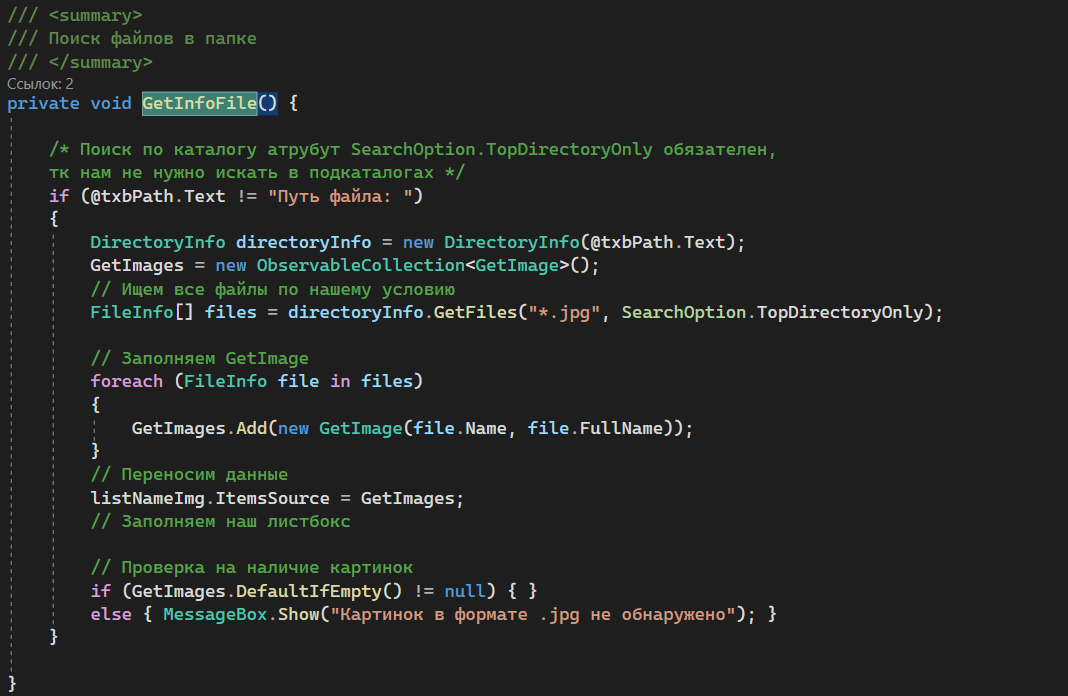
Далее происходит событие «listNameImg\_SelectionChanged» в котором, и происходит загрузка изображения.

Рисунок 6. Процедура GetInfoFile

## Вставка изображения и очистка

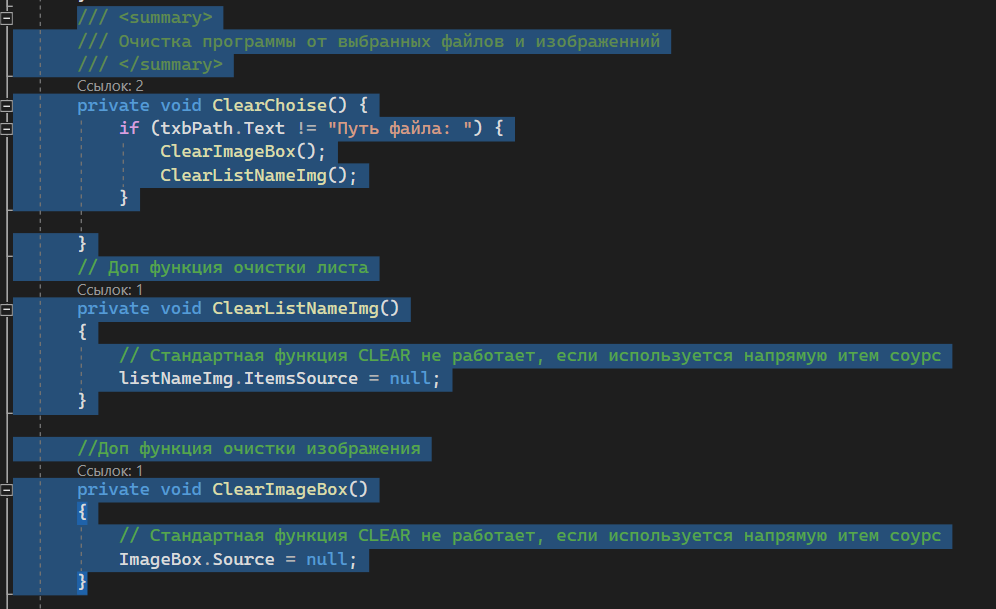
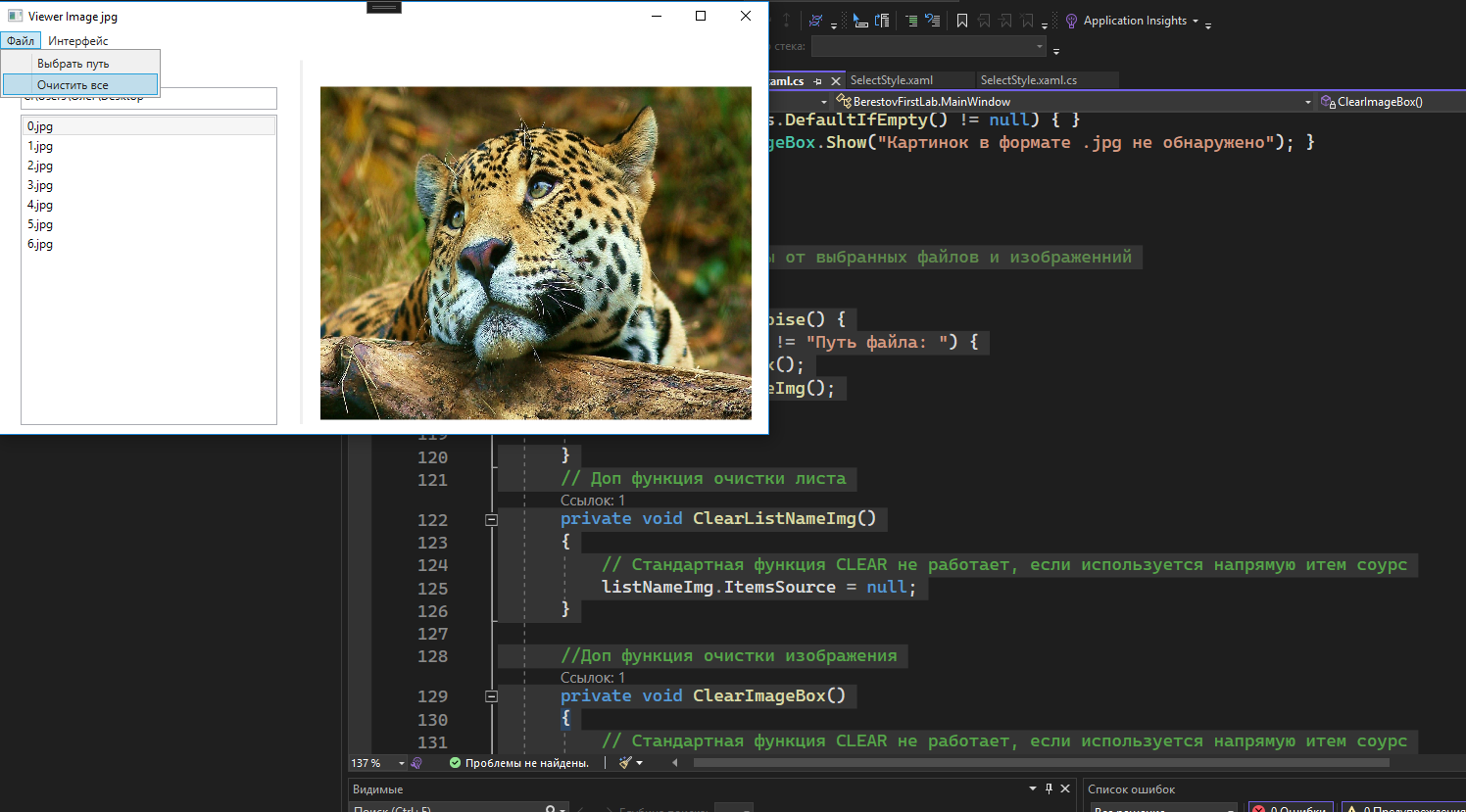
Далее идет вторая часть «Файла», да на описание всего 1 элемента ушло 6 страниц. В данном элементе все намного проще при нажатии происходит событие Know\_Click. В котором вызываются две самописные функции работающие по принципу «listNameImg.ItemsSource = null; ImageBox.Source = null;». Для чего нужно, если есть стандартные Clear, все очень просто, в проекте используется ItemSource, что блокирует возможность использования Clear.

Рисунок 8. Процедуры инициализируемые событием Know\_Click

Рисунок 7. Событие listNameImg\_SelectionChanged

Рисунок 9. Графическое представление программы, а именно MenuItem «Файл»

## Загрузка стилей

Вторая часть элемента Menu «Интерфейс» включает 3 раздела:

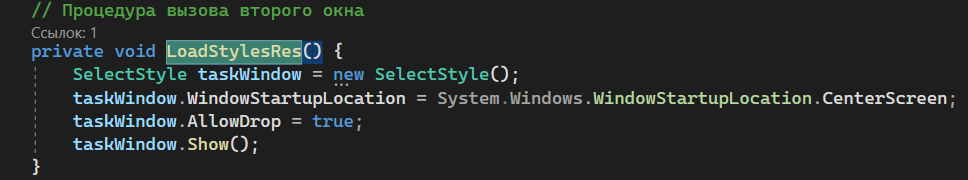
1. «Получить размеры элементов», простой использует процедуру GetInfoElement, сообщает размеры.
2. «Узнать размеры окна», выводит ширину и высоту текущего размера программы.
3. «Выбрать готовый стиль», использует HandlerClick, который уже упоминался в рисунке 4, но в этот раз через case вызывает другую процедуру «LoadStylesRes».

Рисунок 10. Загрузка второго окна «SelectStyle», для выбора стиля

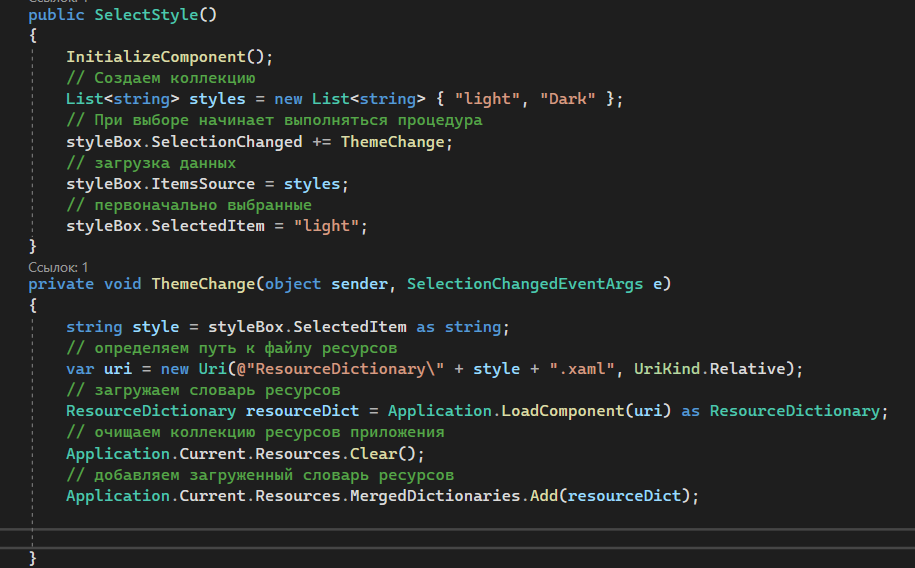
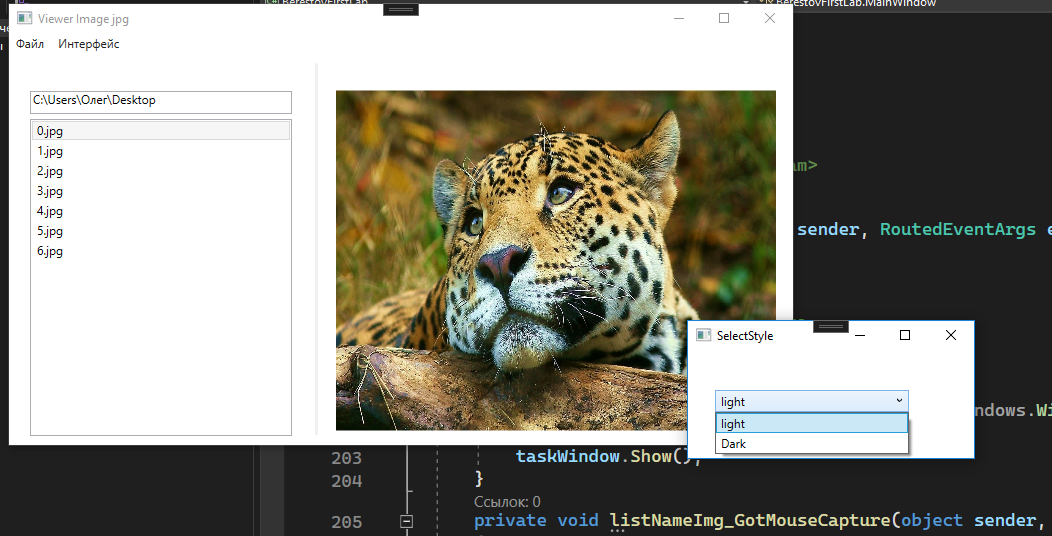
Открывается второе окно, с выбором темы с единственным ComboBox. Используется конструктор и метод, вызываемый в нем. В конструктор создается список названий тем для ComboBox, после чего он просто передает их в него. И потом вызывается метод, который модифицирует название, и ищет его в папке программы. Пришлось доработать, так как

Рисунок 11. Конструктор класса «SelectStyle» и процедура

Рисунок 11. Выбор темы через ComboBox класса «SelectStyle»

## Дополнительный функционал

1. Стили и их реализация
2. Ассеты изображений в проекте, и их стандартная загрузка
3. Проверка разных условий, и ограничения. Файлы не дублируются и многое другое. Было видно в коде, скрины не нужны.
4. Подобие адаптивности, путем создания примитивных медиа запросов.
5. Использование Git, и доведение его до ума [**ССЫЛКА НА ГИТ**](https://github.com/Oleja2018/BerestovFirstLab).
6. Реализация меню. Было видно в коде, скрины не нужны.
7. Сохранение истории, возможность модификации проекта. Было видно в коде, скрины не нужны.

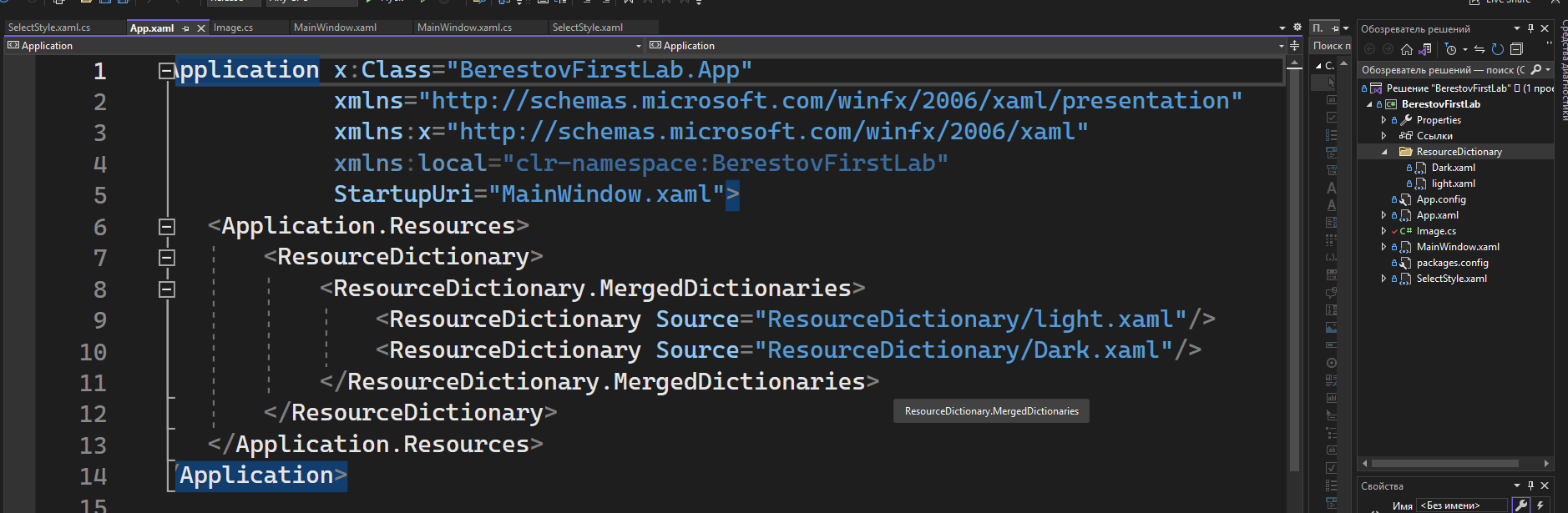
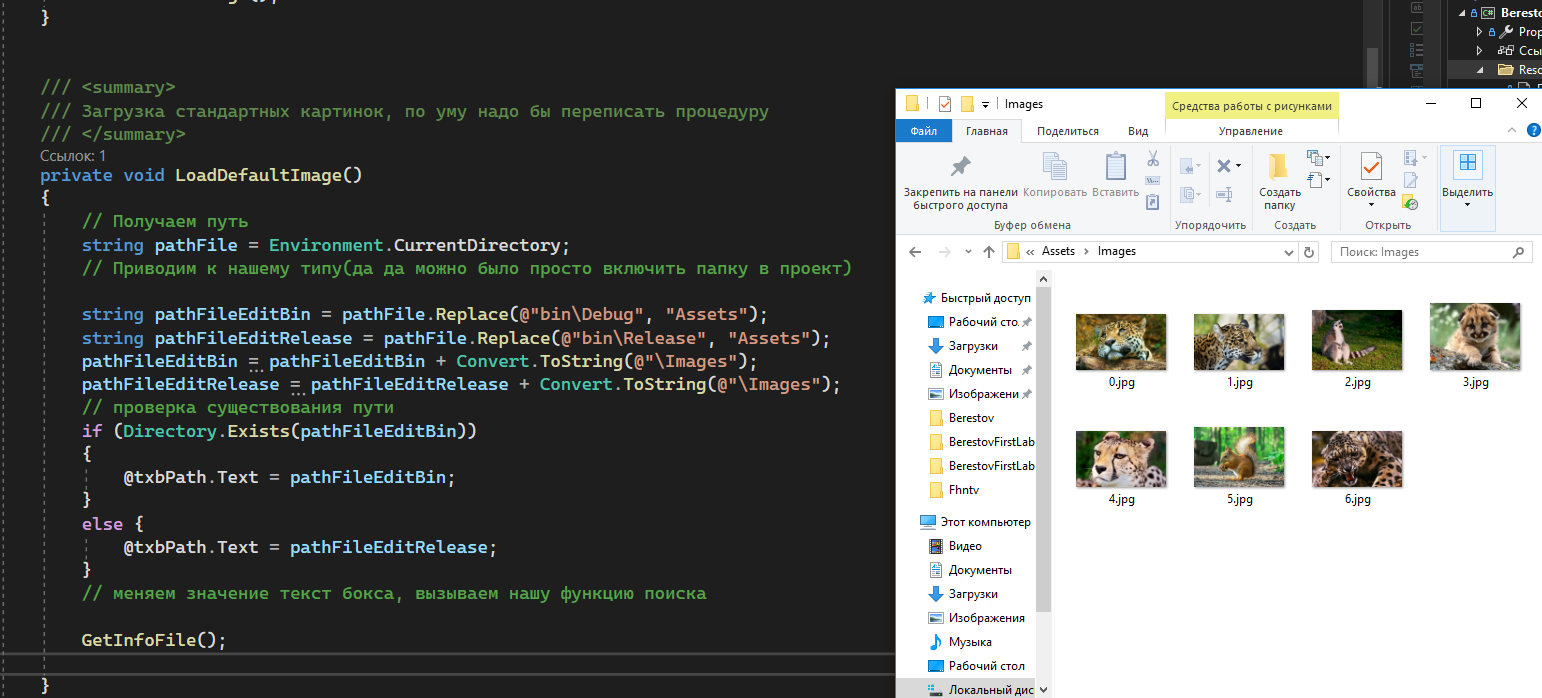
Далее будут представлены скриншоты вышеперечисленных пунк

Рисунок 12. Загрузка стилей из папки, и использования слияния

Рисунок 13. Ассеты изображений в проекте, и их стандартная загрузка

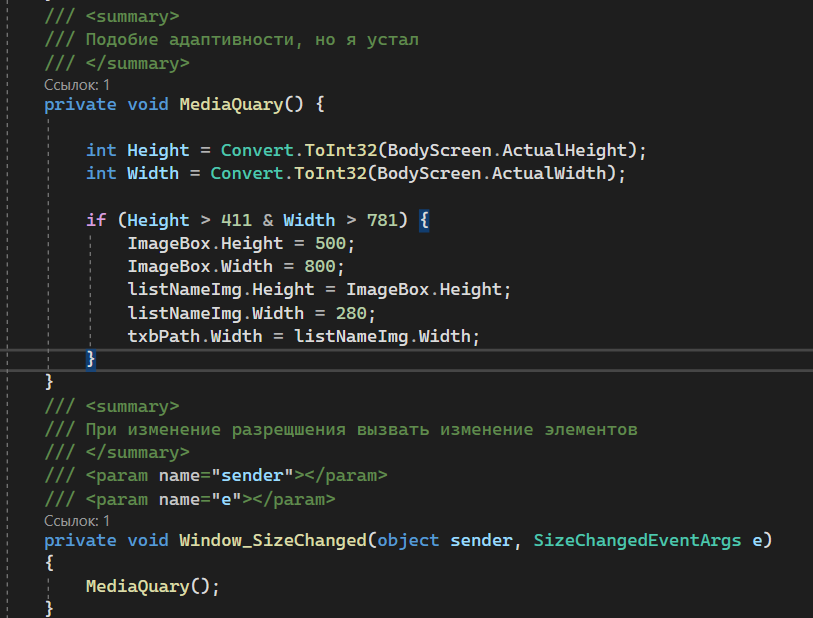


Рисунок 14. Подобие адаптивности, путем создания примитивных медиа запросов.