

Создание страничного интерфейса приложения

1. Запустить среду разработки Visual Studio

2. Выбрать «Create new project»

3. Выбрать «WPF App»

При разработке интерфейсов разработчик может использовать две модели: **оконную** или **страничную** (в настоящее время используют чаще). В рамках практической работы №5 используем страничный вариант.

Контейнеры компоновки.

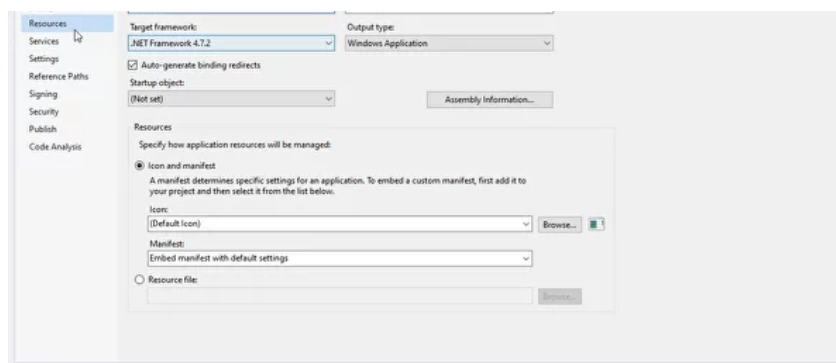
Grid. Наиболее мощный и часто используемый контейнер, похожий на таблицу. Он содержит столбцы и строки, количество которых можно задать. Для определения строк используется свойство RowDefinition, а для столбцов — ColumnDefinition

Для привязки элемента управления к конкретной ячейке необходимо использовать свойства Grid.Column и Grid.Row, причем нумерация строк и столбцов начинается с нуля

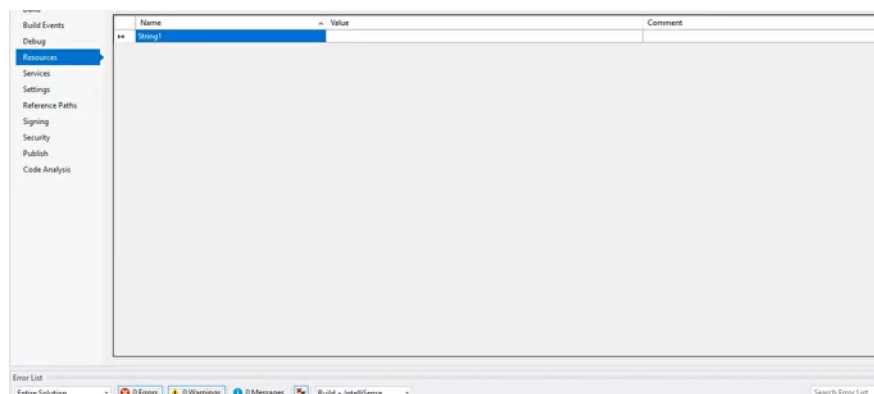
StackPanel. Позволяет размещать элементы управления поочередно друг за другом

4. Добавить логотип приложения в Resources

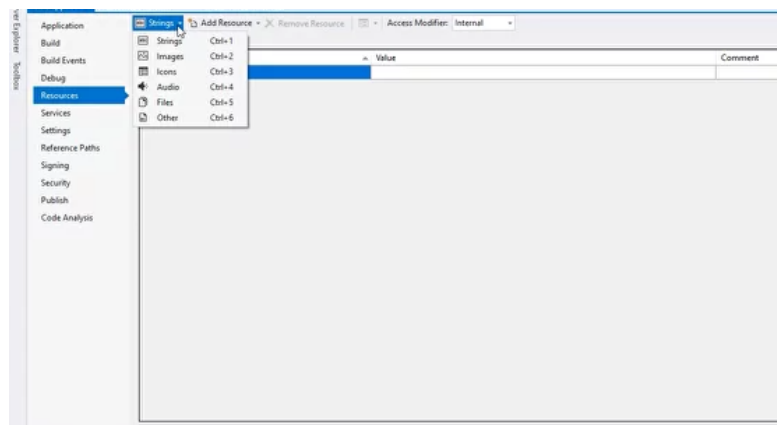
а) Выбрать вкладку TestCleaning



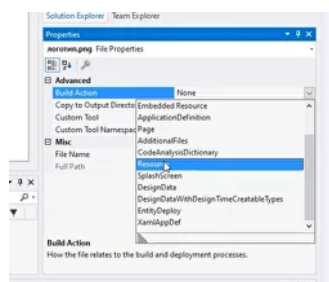
б) Открыть вкладку Resources



в) Выбрать список изображений



г) Перетащить логотип в открывшееся окно



д) Установить значение поля Build Action — Resource

е) Для отображения логотипа используем элемент Image

Далее установите заголовок приложения и цвета для верхней и нижней частей.

5. Создать навигацию

Для оконной навигации основным элементом является страница — Page, которая должна находиться в каком-либо контейнере. Для этого в основном окне приложения разместим элемент Frame, где будут собраны страницы приложения

Далее создадим страницу, которая будет отображаться при первом запуске приложения, и вторую страницу для тестирования навигации между ними

а). Правой кнопкой жмем на название проекта — Add — Page

б). Указываем ее название CleaningPage

в). Размещаем на странице одну кнопку

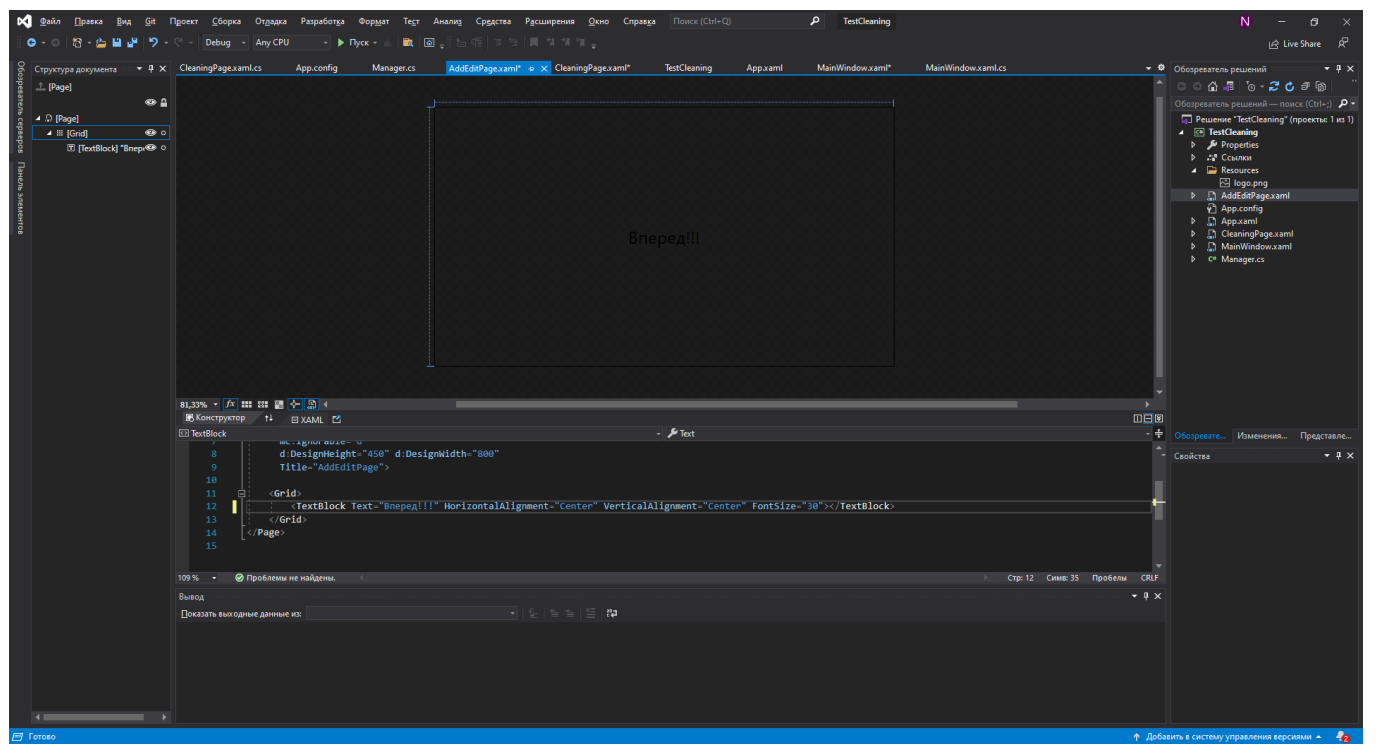
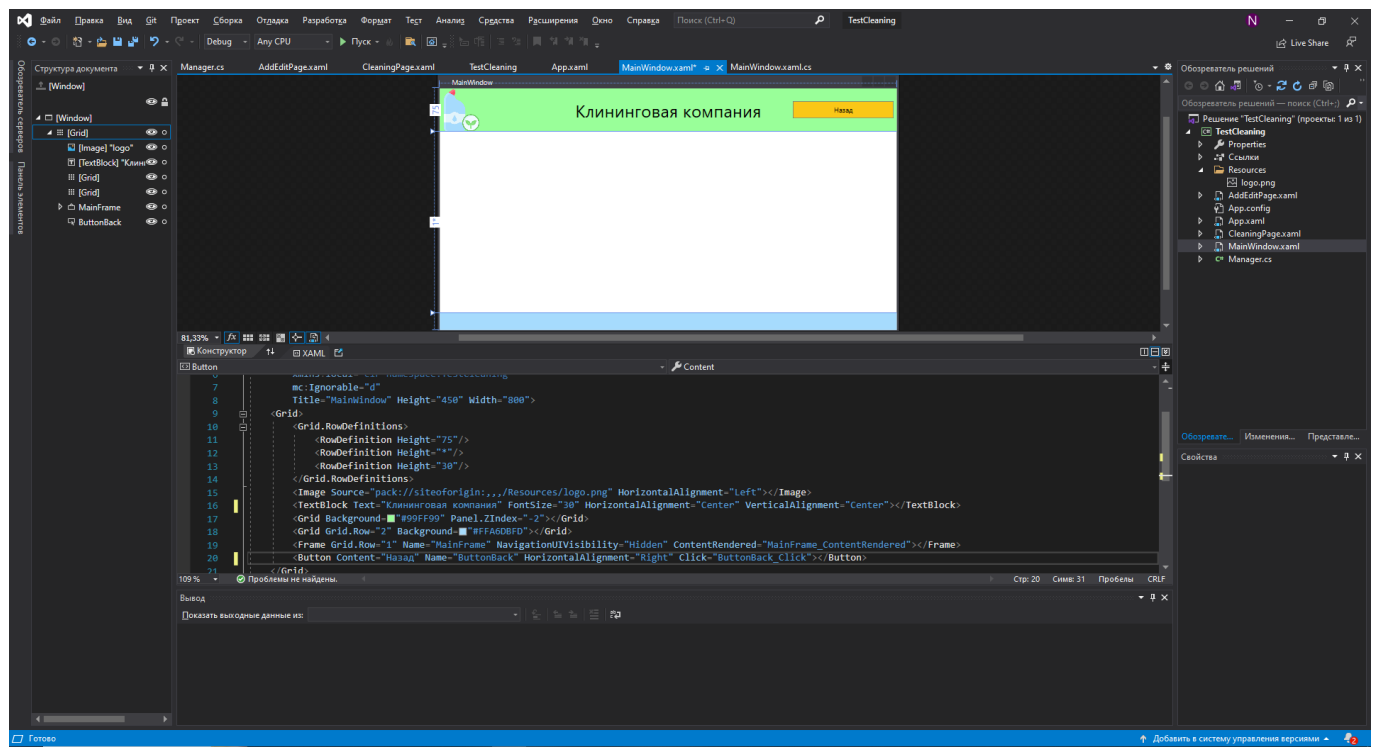
г). Создаем еще одну страницу, повторив пункт 1

д). Даем ей название AddEditPage

е). Размещаем в ней TextBlock

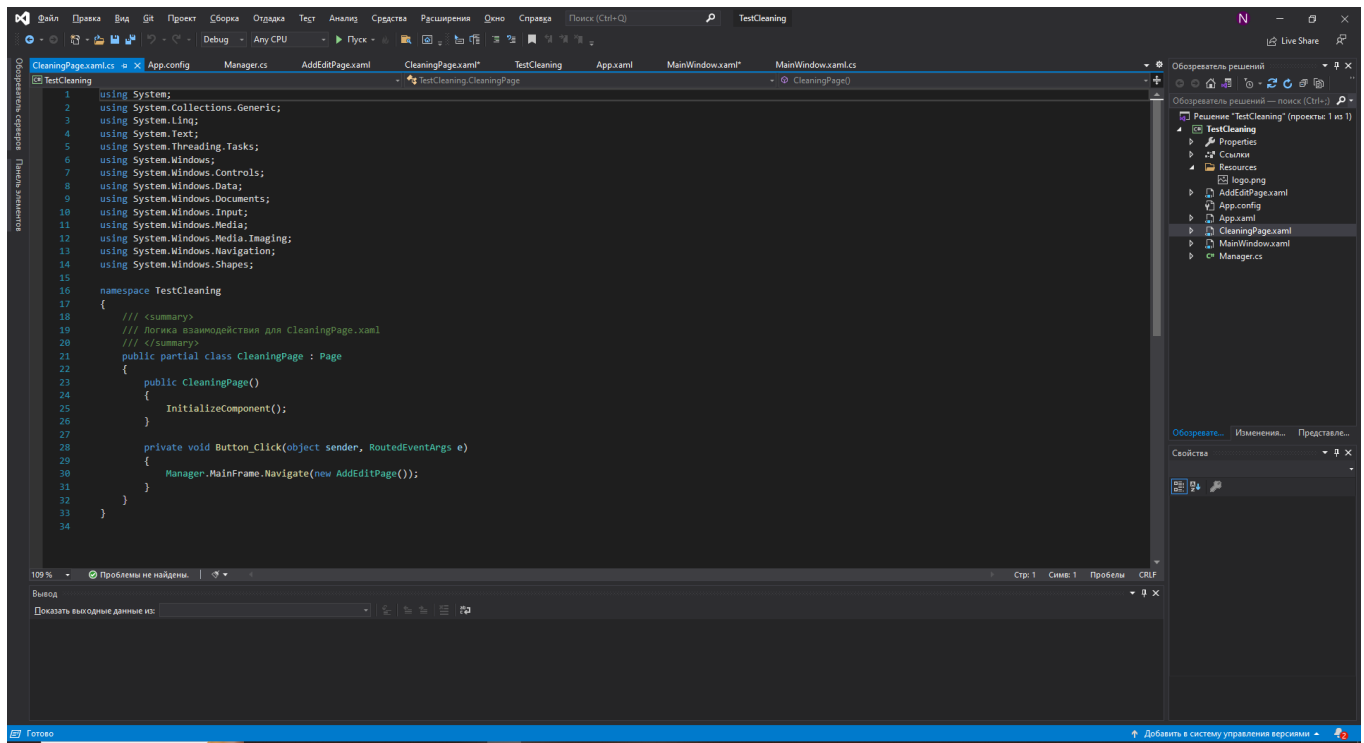
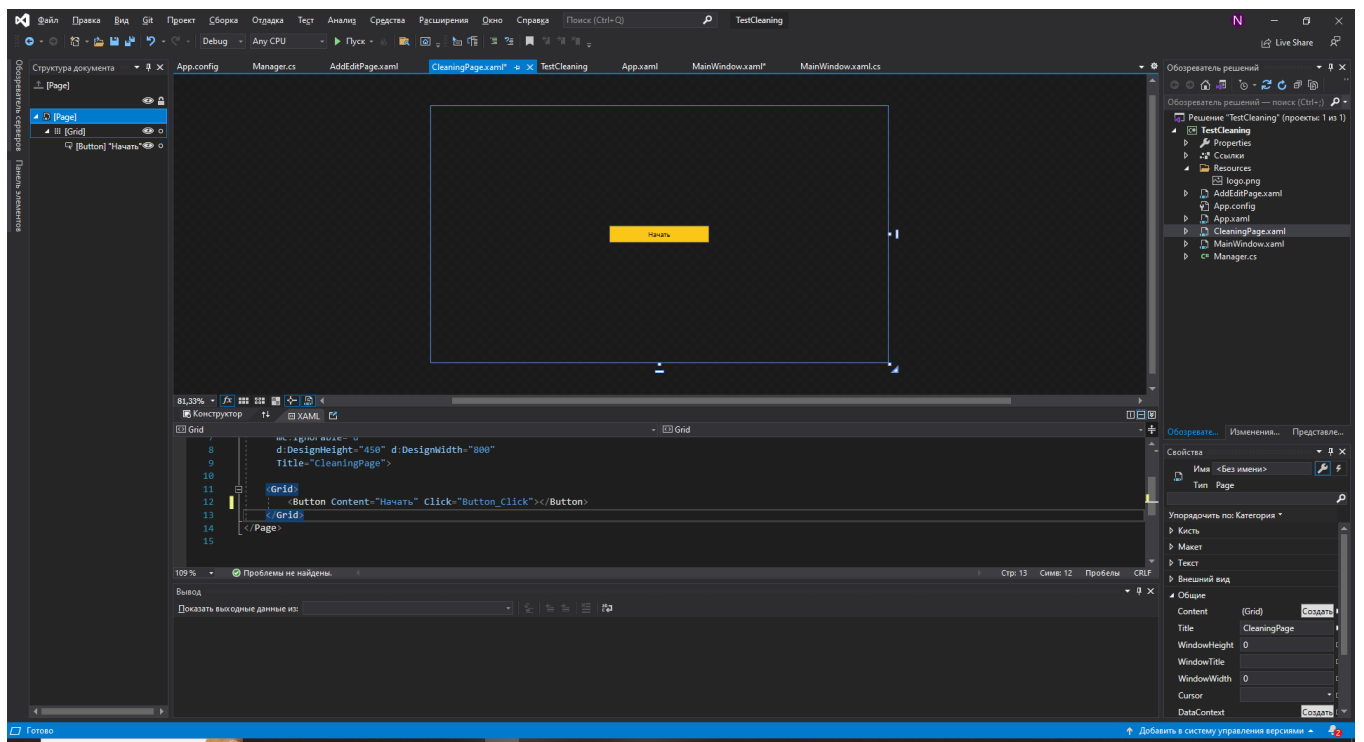
ж) Укажем имя фрейма

з). Для отображения первой страницы необходимо прописать следующий код:
`MainFrame.Navigate(new NamePage());`

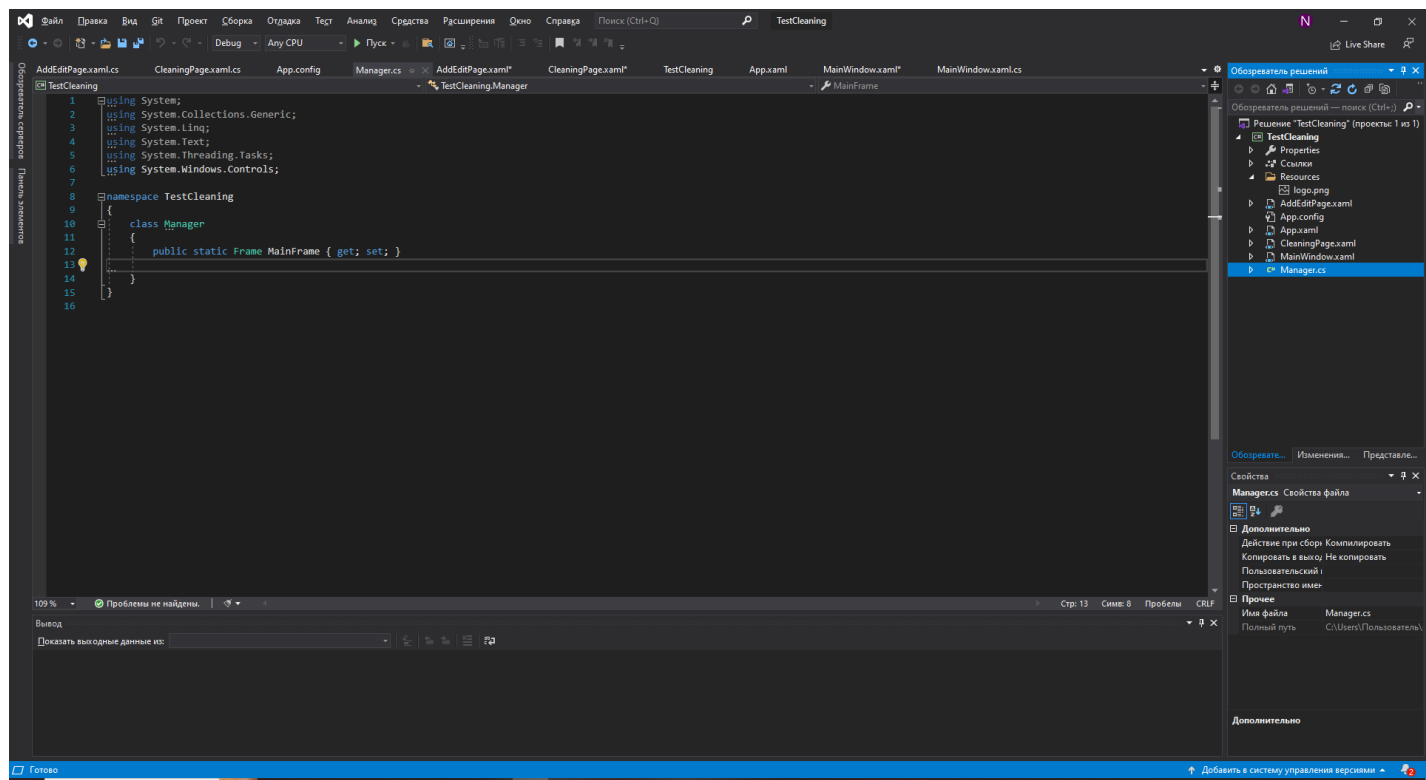


6. Добавить элементы управления и их свойства

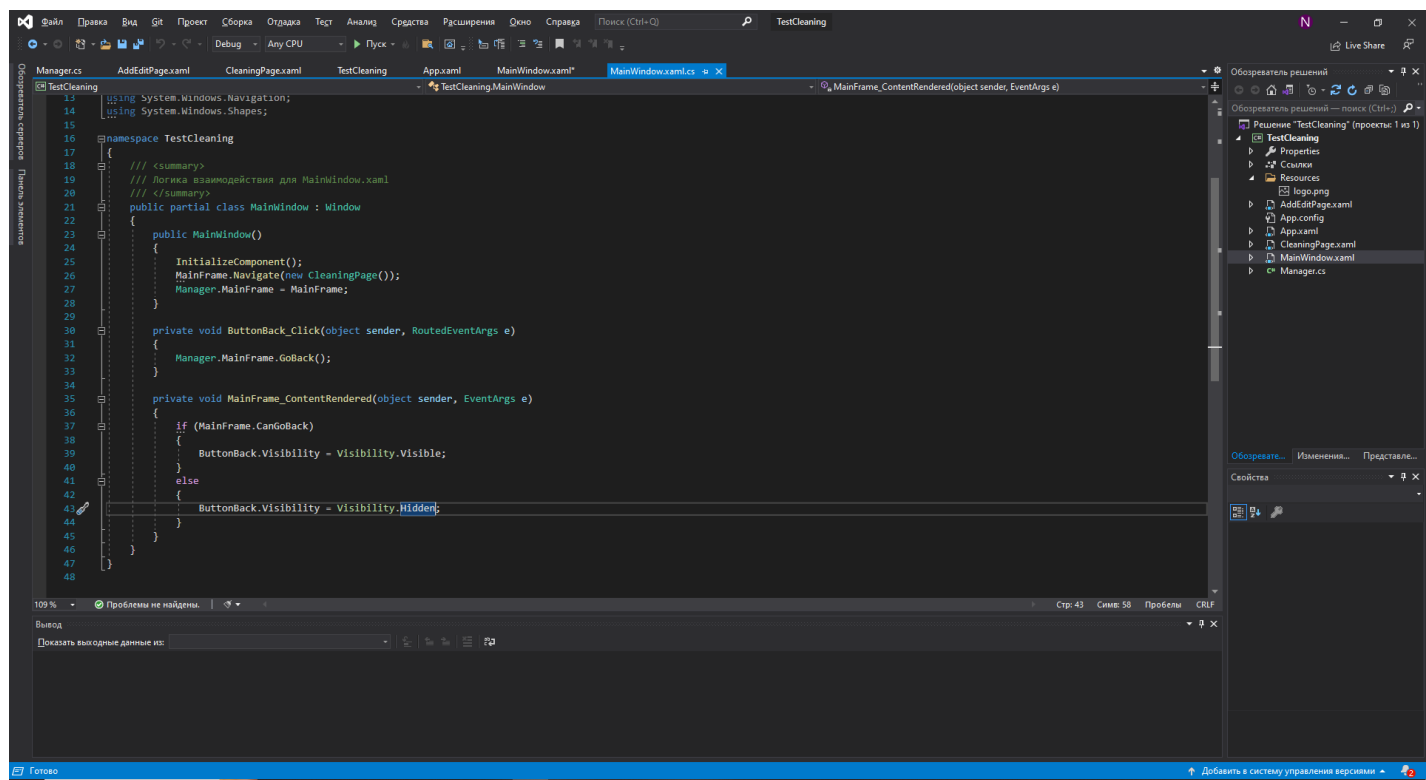
Для взаимодействия с кнопкой будем использовать событие `Click()`. С помощью него мы сможем перейти с первой страницы на вторую. Для создания события пропишем соответствующее значение в верстке:



Чтобы добраться до MainFrame для навигации, необходимо создать новый класс:



В коде для кнопки «Назад» пропишем:

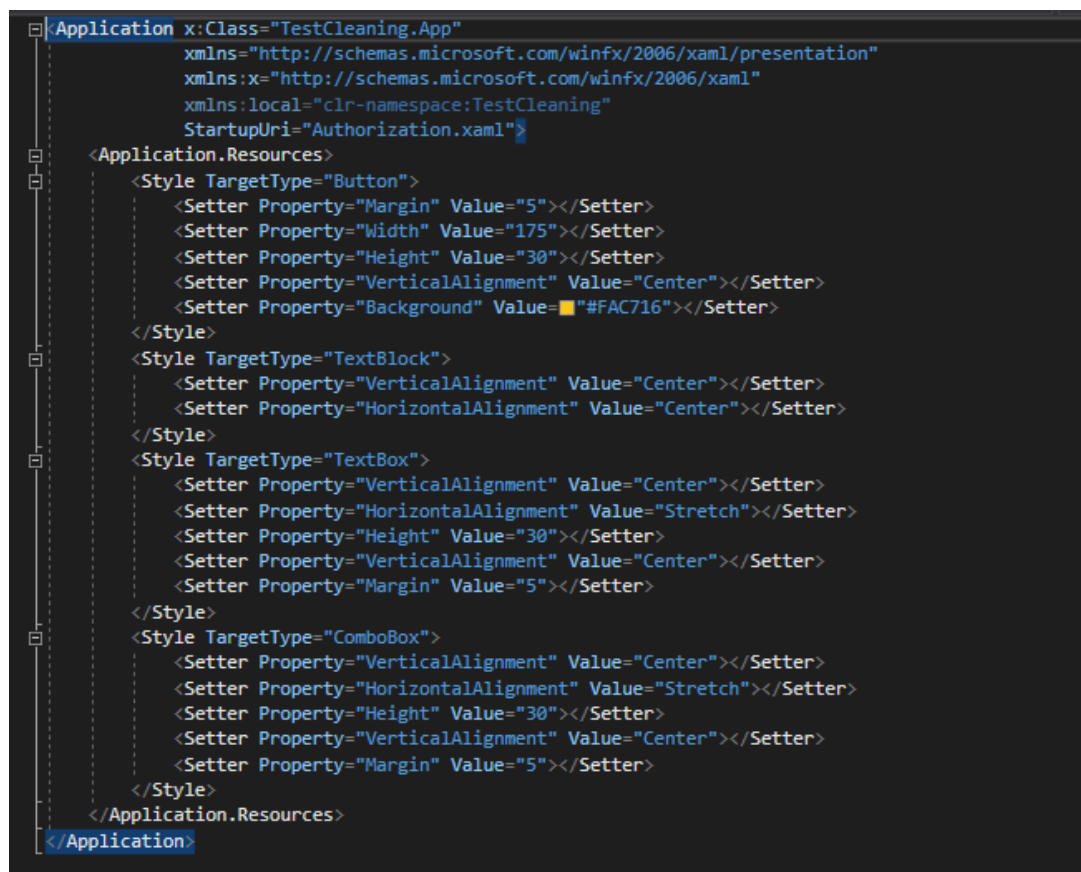
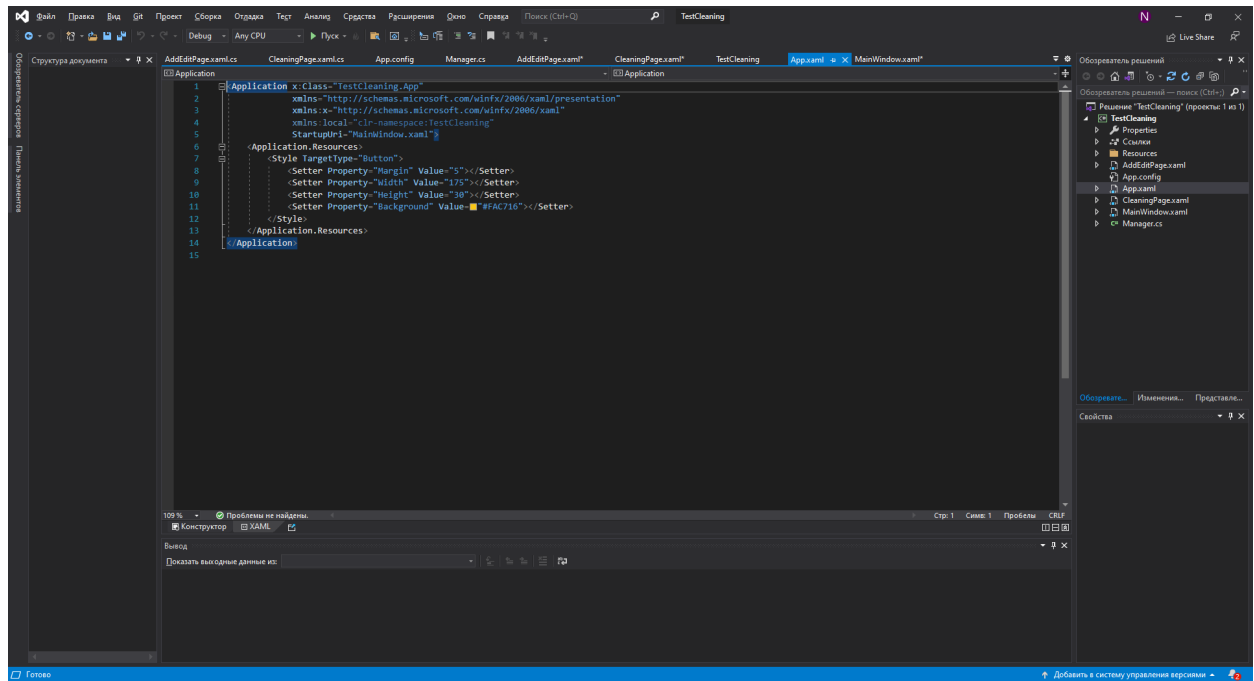


Чтобы скрыть стандартное навигационное меню, воспользуемся свойством фрейма — *NavigationUIVisibility='Hidden'*

7.Создадим глобальные стили приложения

Чтобы их создавать в проекте, есть файл App.xaml. Используем тег Style и свойство TargetType, чтобы указать, для каких элементов предназначен данный стиль.

Внутри тега Style используем тег Setter. В нем установим необходимые свойства.



И теперь удалим написанные ранее свойства элементов, для которых есть стиль

Самостоятельная работа.

Предусмотрите интерфейс для всех необходимых элементов управления в своем проекте, используя глобальные стили.