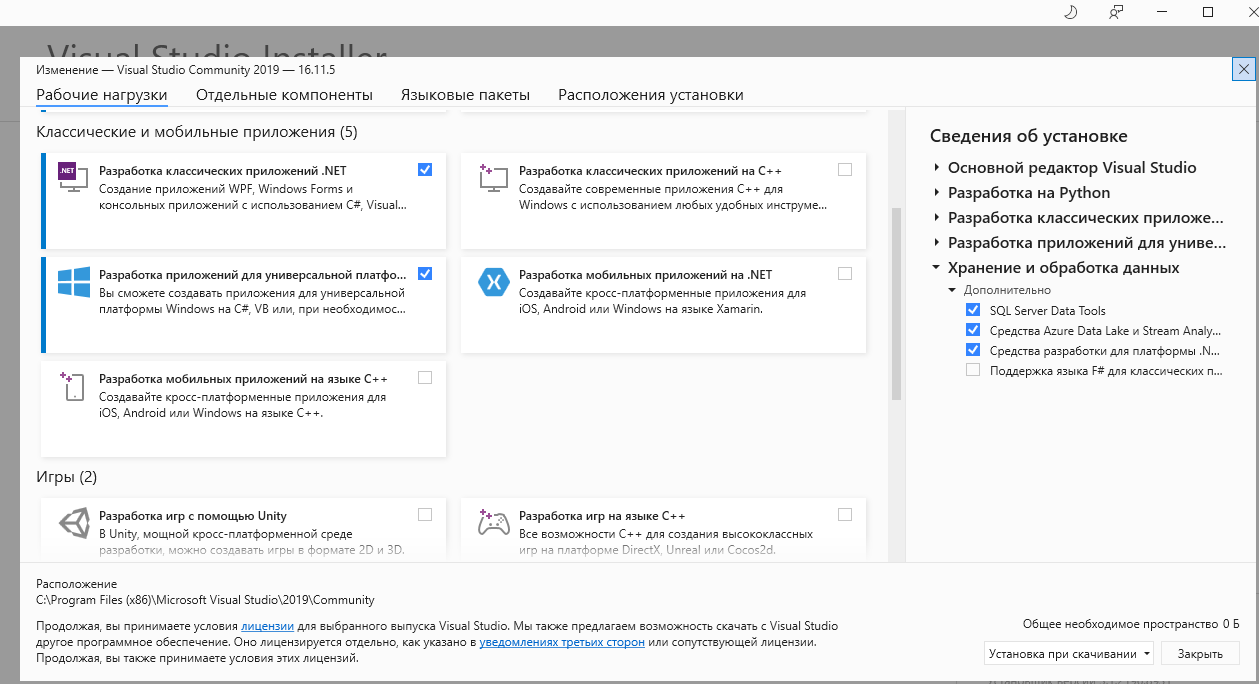
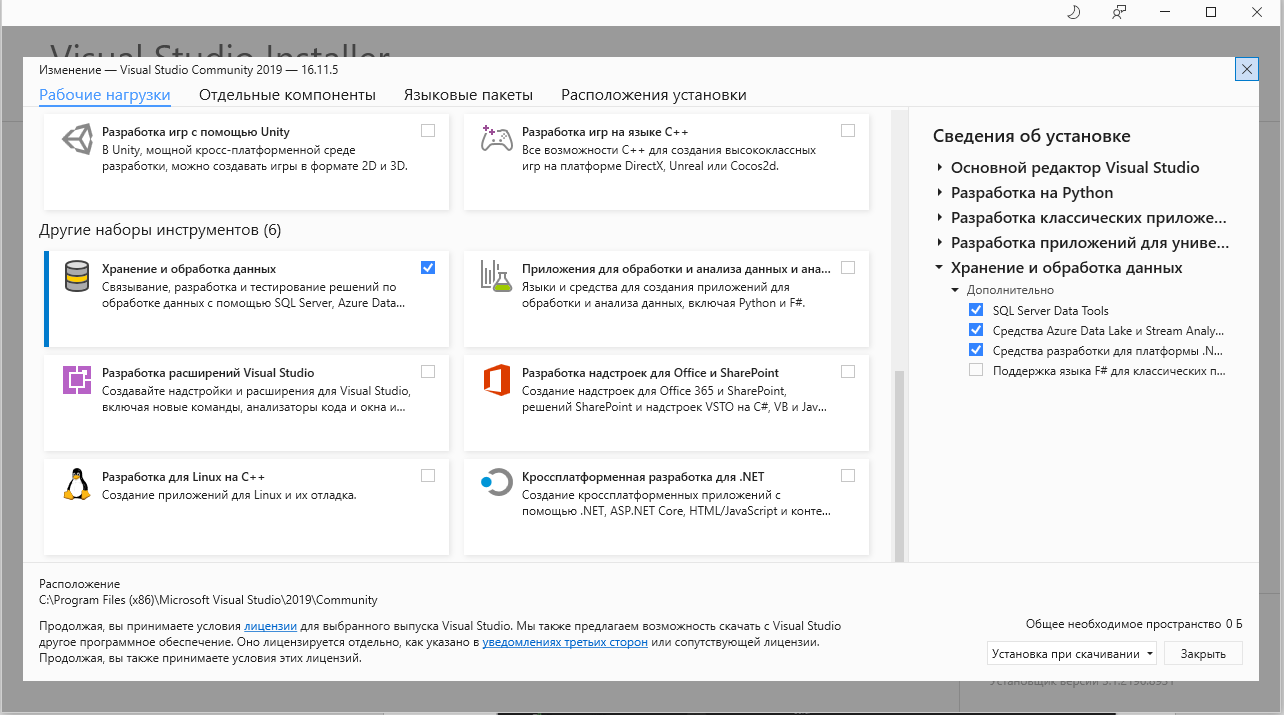
**Разработка интерфейса desktop приложения на языке C#**

Установите Visual Studio 2019 или 2022

При установке выберите компоненты:

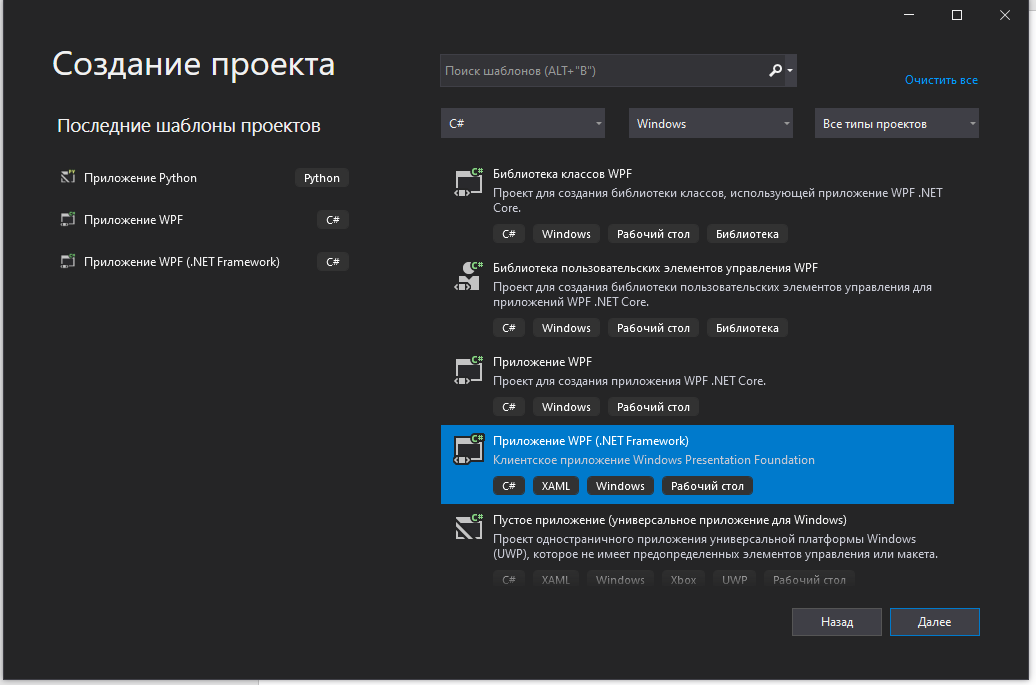




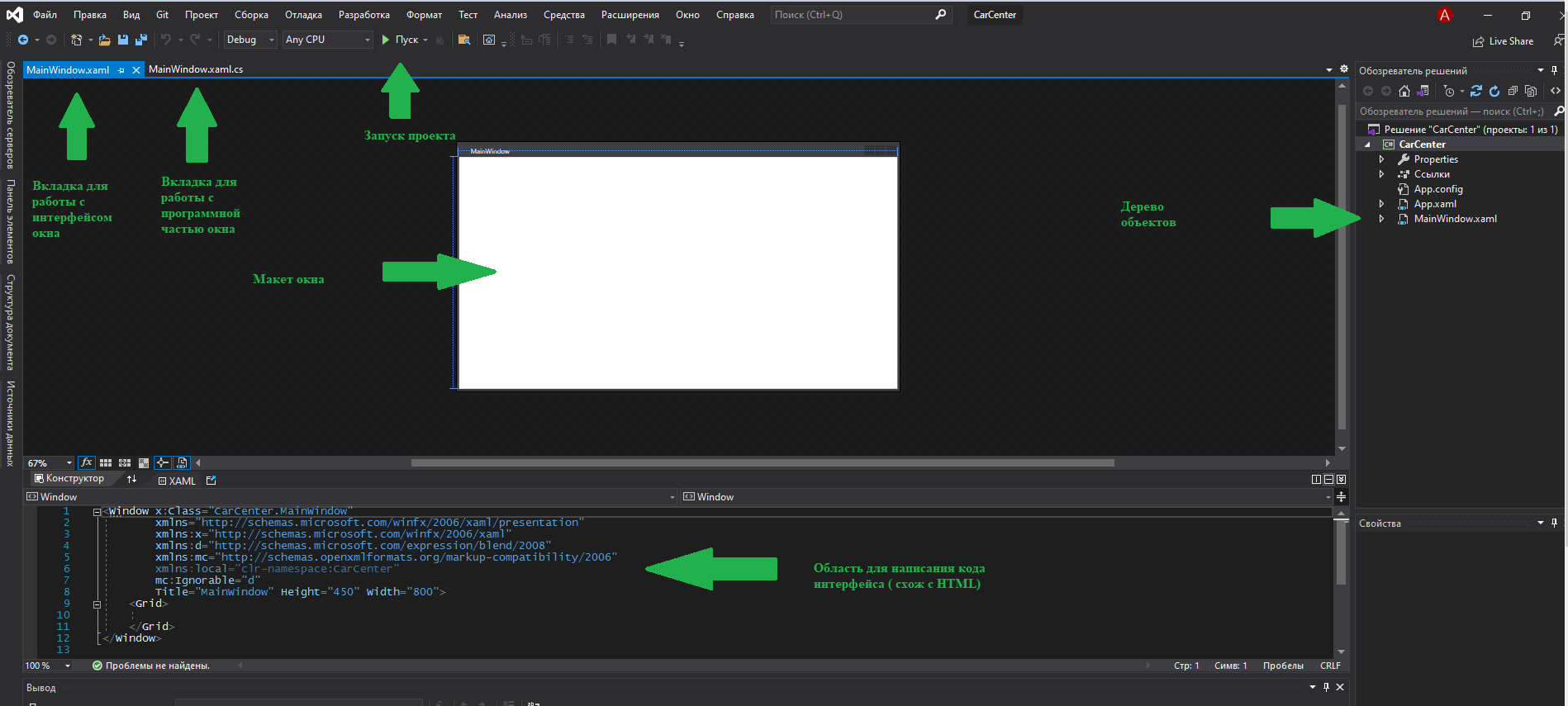
Далее создаем проект, если среда на английском языке, ориентируйтесь по тегам.

Не забудьте затем дать название проекту.

**Важно! Обращайте внимание на название проекта в задании, называйте точно также, за это дают баллы.**

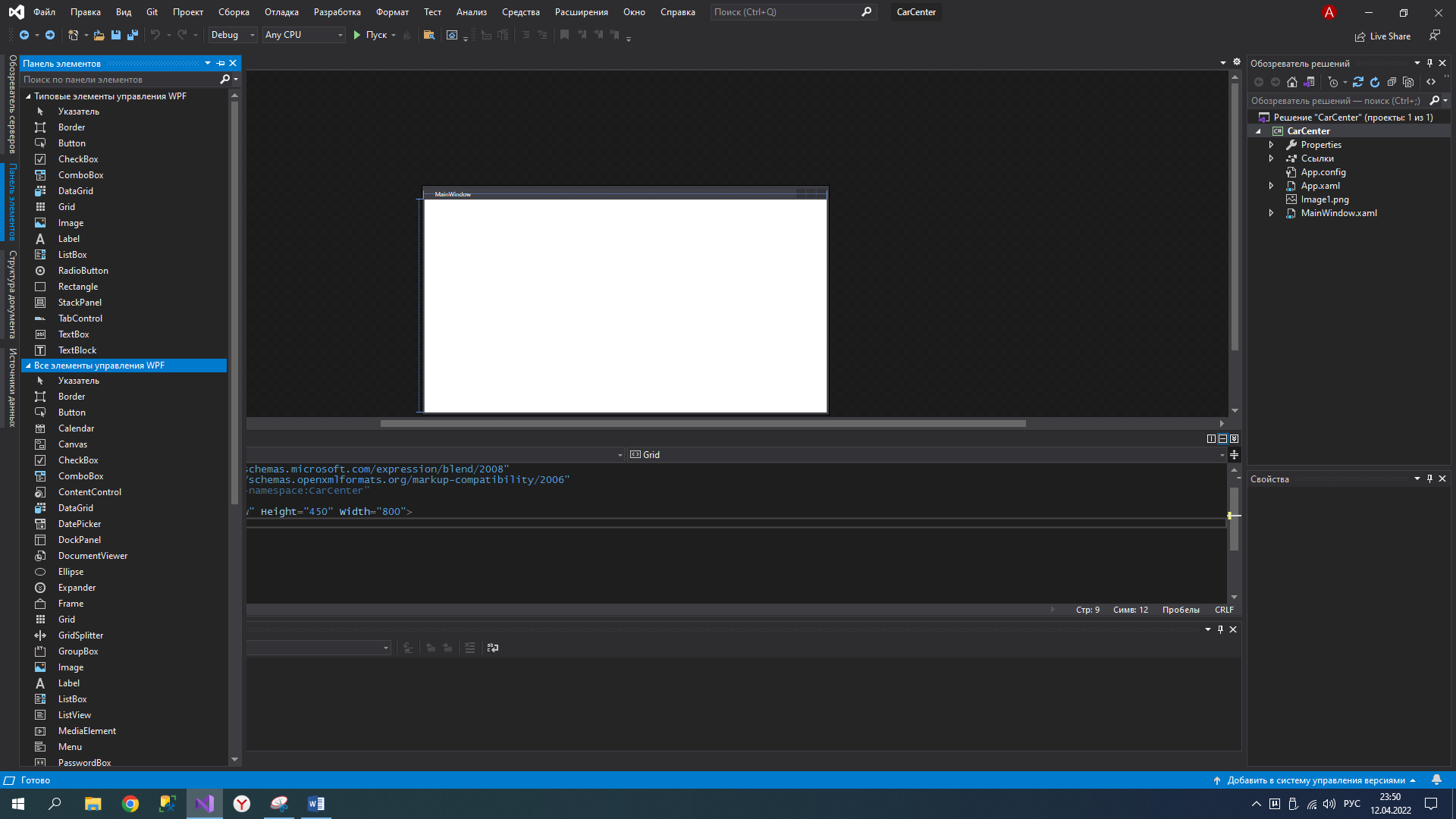


Рассмотрите интерфейс программы. Мы пока будем работать с файлами, имеющими расширение .xaml (отвечают за интерфейс).



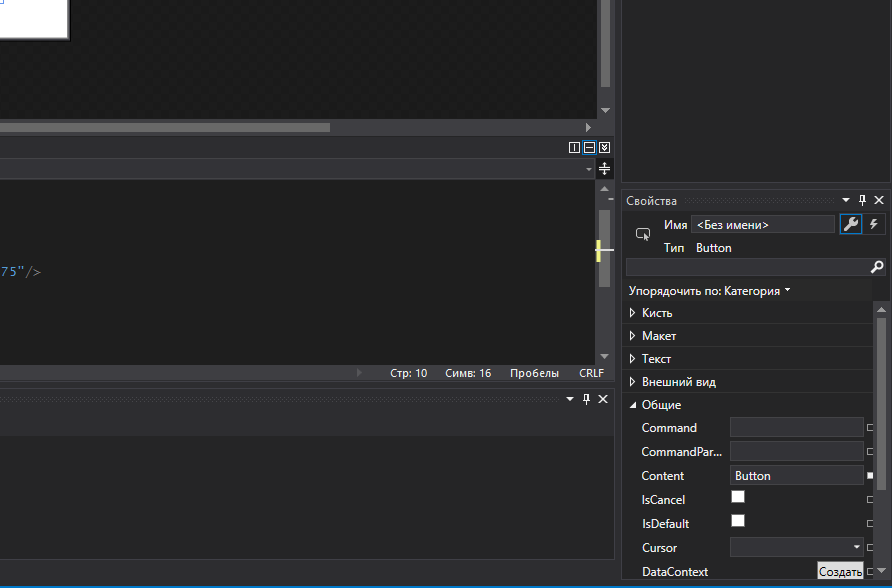
Приложение может быть многооконным, поэтому такие вкладки появляются парно при создании новой формы и имеют названия как у этой формы.

Есть два метода создания компонентов интерфейса. С помощью кода, либо методом перетаскивания из панели элементов.



Настраивать элементы можно также двумя методами, либо программно, либо с помощью панели свойств. Выберите метод, который позволит вам сделать это быстро.

**Важно! Полезно знать и программное добавление, и настройку элементов, без этого практически невозможно правильно разместить и закрепить их на форме, чтобы они не «расползлись» после добавления других элементов.**



Рассмотрим элементы, их назначение и свойства.

**Категории элементов управления**

Элементы управления Windows Presentation Foundation (WPF) можно логически сгруппировать в несколько категорий. Эти категории можно использовать для выбора соответствующего элемента управления для сценария, что поможет увидеть, какие элементы управления имеют схожие шаблоны использования или функциональные возможности.

Основные элементы помечены зеленым цветом, названия являются ссылками, по которым можно перейти и почитать в подробностях про каждый элемент.

**Макет**

Элементы управления макета используются для управления размером, измерениями, размещением и расположением дочерних элементов.

* [Border](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.border)
* [BulletDecorator](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.primitives.bulletdecorator)
* [Canvas](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.canvas)
* [DockPanel](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.dockpanel)
* [Expander](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.expander)
* [Grid](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.grid)
* [GridSplitter](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.gridsplitter)
* [GroupBox](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.groupbox)
* [Panel](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.panel)
* [ResizeGrip](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.primitives.resizegrip)
* [Separator](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.separator)
* [ScrollBar](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.primitives.scrollbar)
* [ScrollViewer](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.scrollviewer)
* [StackPanel](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.stackpanel)
* [Thumb](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.primitives.thumb)
* [Viewbox](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.viewbox)
* [VirtualizingStackPanel](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.virtualizingstackpanel)
* [Window](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.window)
* [WrapPanel](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.wrappanel)

**Кнопки**

Кнопки являются одним из основных элементов управления пользовательского интерфейса. Приложения обычно выполняют некоторую задачу в [Click](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.primitives.buttonbase.click) случае, когда пользователь щелкает их.

* [Button](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.button)
* [RepeatButton](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.primitives.repeatbutton)

**Отображение данных**

Элементы управления отображением данных используются для отображения информации из источника данных.

* [DataGrid](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.datagrid)
* [ListView](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.listview)
* [TreeView](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.treeview)

**Выбор и отображение дат**

Элементы управления даты используются для отображения и выбора данных календаря.

* [Calendar](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.calendar)
* [DatePicker](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.datepicker)

**Меню**

Меню используются для группирования связанных действий или для предоставления контекстной помощи.

* [ContextMenu](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.contextmenu)
* [Menu](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.menu)
* [ToolBar](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.toolbar)

**Выбор**

Элементы управления выбором позволяют выбрать один или несколько вариантов.

* [CheckBox](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.checkbox)
* [ComboBox](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.combobox)
* [ListBox](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.listbox)
* [RadioButton](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.radiobutton)
* [Slider](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.slider)

**Навигация**

Элементы управления навигацией расширяют возможности навигации приложения, создавая внешний вид конечных фреймов или вкладок приложения.

* [Frame](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.frame)
* [Hyperlink](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.documents.hyperlink)
* [Page](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.page)
* [NavigationWindow](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.navigation.navigationwindow)
* [TabControl](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.tabcontrol)

**Диалоговые окна**

Диалоговые окна обеспечивают поддержку распространенных сценариев взаимодействия с пользователем, например печать.

* [OpenFileDialog](https://docs.microsoft.com/ru-ru/dotnet/api/microsoft.win32.openfiledialog)
* [PrintDialog](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.printdialog)
* [SaveFileDialog](https://docs.microsoft.com/ru-ru/dotnet/api/microsoft.win32.savefiledialog)

**Информация о пользователе**

Элементы управления информацией пользователя предоставляют контекстную обратную связь или уточняют пользовательский интерфейс приложения. Пользователь обычно не может взаимодействовать с этими элементами управления.

* [AccessText](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.accesstext)
* [Label](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.label)
* [Popup](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.primitives.popup)
* [ProgressBar](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.progressbar)
* [StatusBar](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.primitives.statusbar)
* [TextBlock](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.textblock)
* [ToolTip](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.tooltip)

**Документы**

WPF включает несколько специализированных элементов управления для просмотра документов. Эти элементы управления оптимизируют процесс чтения, основываясь на целевом пользовательском сценарии.

* [DocumentViewer](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.documentviewer)
* [FlowDocumentPageViewer](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.flowdocumentpageviewer)
* [FlowDocumentReader](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.flowdocumentreader)
* [FlowDocumentScrollViewer](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.flowdocumentscrollviewer)
* [StickyNoteControl](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.stickynotecontrol)

**Входные данные**

Элементы управления вводом позволяют пользователю вводить текст и другое содержимое.

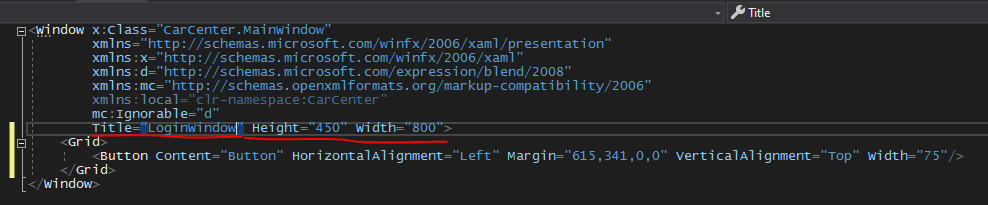
* [TextBox](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.textbox)
* [RichTextBox](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.richtextbox)
* [PasswordBox](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.passwordbox)

**Мультимедиа**

В состав WPF входит интегрированная поддержка размещения аудио и видеосодержимого, а также [кодеки] для большинства популярных форматов изображений.

* [Image](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.image)
* [MediaElement](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.mediaelement)
* [SoundPlayerAction](https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.controls.soundplayeraction)

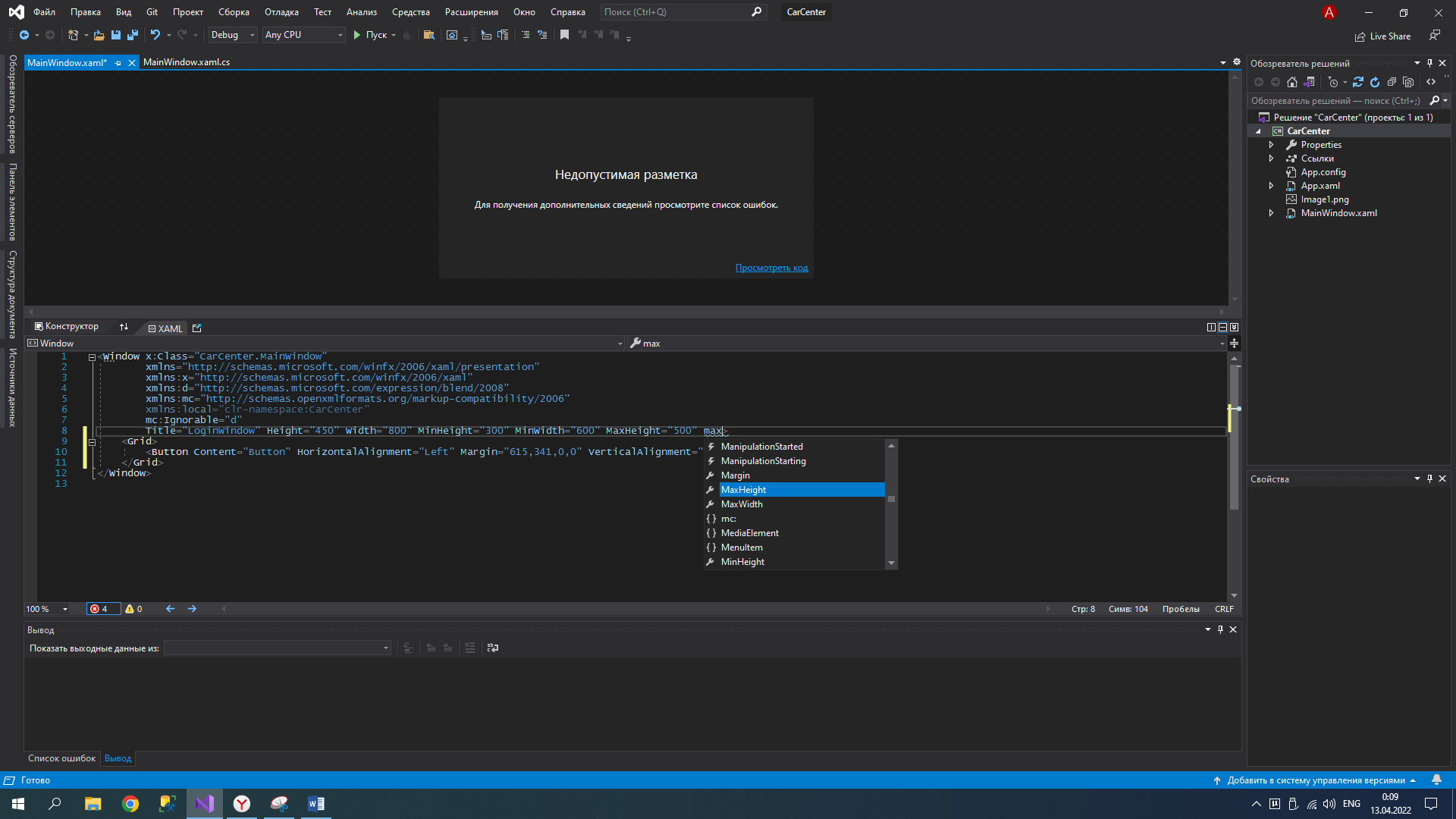
Создадим окно авторизации.



Для начала необходимо задать заголовок окна и его размеры.

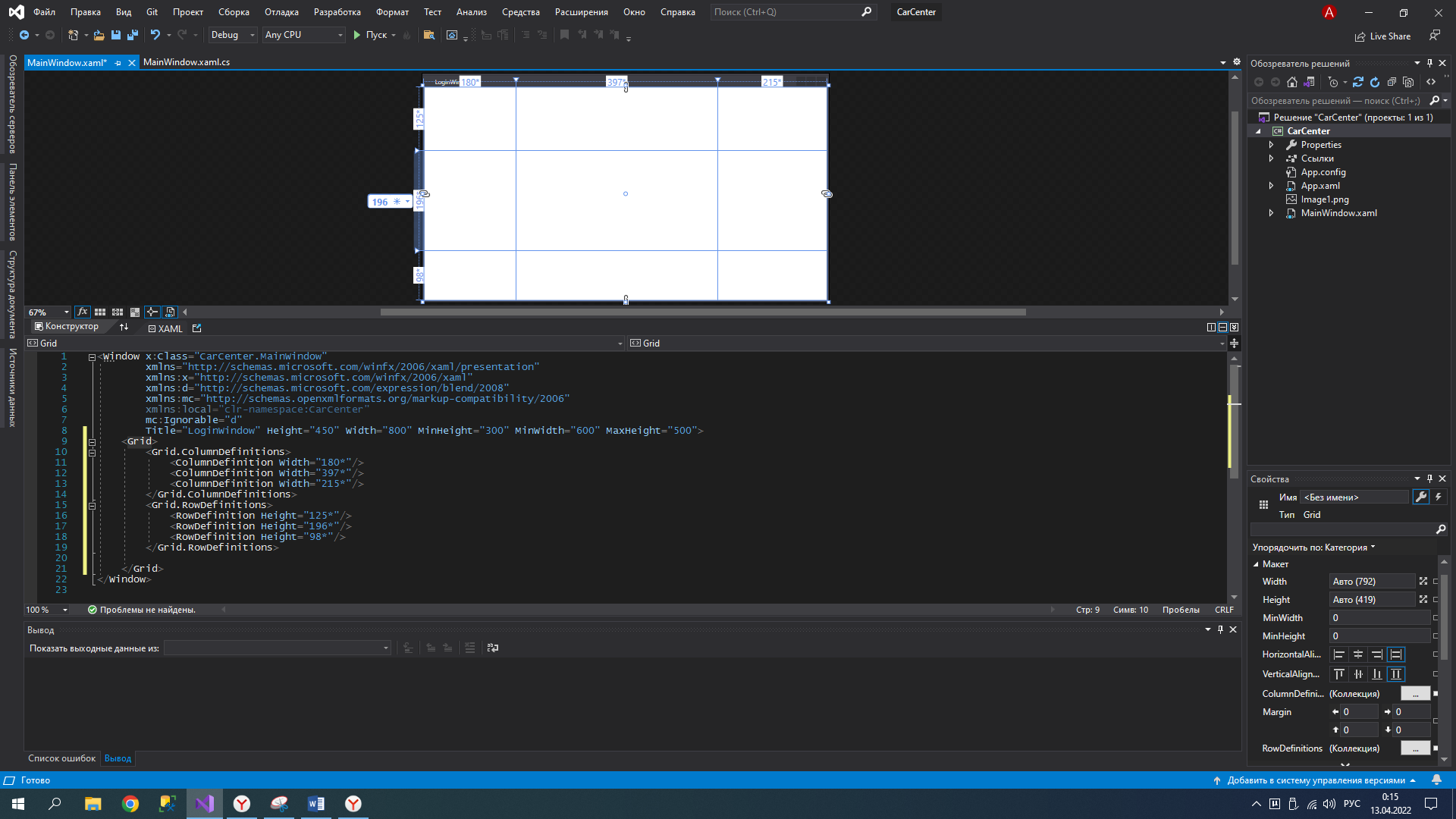
Также не забывайте про минимальные и максимальные размеры окна.

У среды разработки есть приятная особенность, которая повышает скорость разработки, при вводе кода, она предлагает подходящие или предполагаемые конструкции. Например, вы ввели максимальную высоту, следом она предложит ввести максимальную ширину.



Для удобства разработки элементы помещают внутри таблицы. Таблицу также можно задать двумя способами. Рассмотрим быстрый. Поднесите курсор к краю формы, чтобы рядом с ним появился белый плюсик, затем зажмите ЛКМ и тяните до противоположного края, таким образом можно размечать столбцы и строки таблицы.

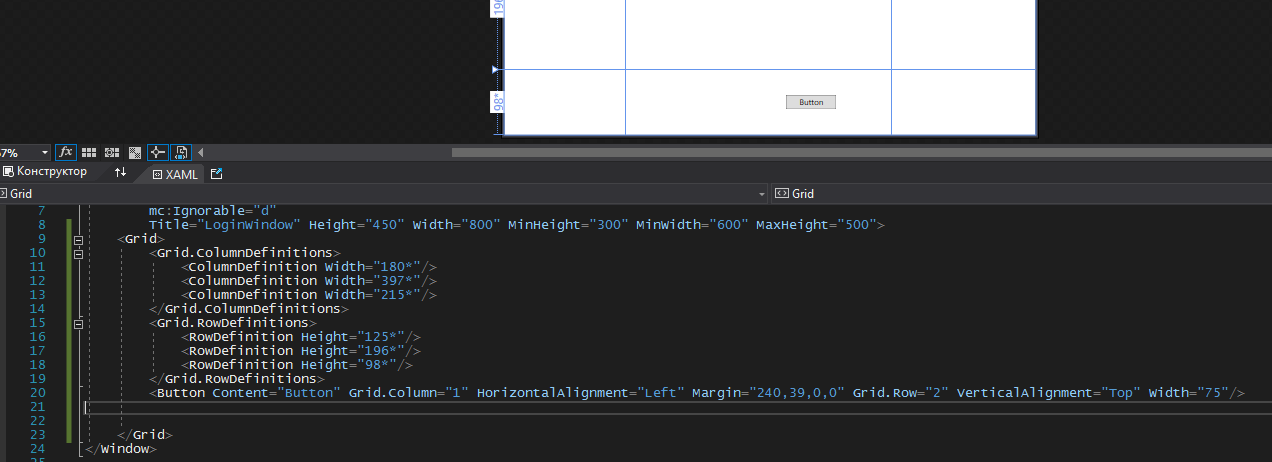
Смотрим, как изменился код. Обращаем внимание, что у строк и столбцов можно указывать размеры. Удобнее всего это делать в коде. Вместо числа можно подставить **auto**, тогда ячейки будут растягиваться по размеру добавленных в них элементов, либо поставить знак **\*** тогда строки или столбцы будут иметь одинаковый размер относительно окна.



Принцип программирования на **XAML** схож с **HTML**, здесь также есть открывающие и закрывающие теги.

Основное действо происходит в теге **Grid**.

Из панели элементов перетащите элемент Button (Кнопка) посмотрите, как изменился код.



Дальнейшие элементы будут располагаться после закрывающего тега **Grid.RowDefinitions**.

Рассмотрим свойства кнопки, которые сформировались автоматически:

**Content**- Отвечает за надпись на кнопке

**Grid.Column**- номер столбца в котором находится элемент (Отсчет с 0)

**HorizontalAlignment**- выравнивание в ячейке по горизонтали (к правому краю, к левому краю, по центру)

**Margin**- отступы от края ячейки до элемента. Это свойство появляется, когда мы перетаскиваем элемент с панели инструментов или мышкой, оно очень мешает корректно расположить элемент, лучше его удалять.

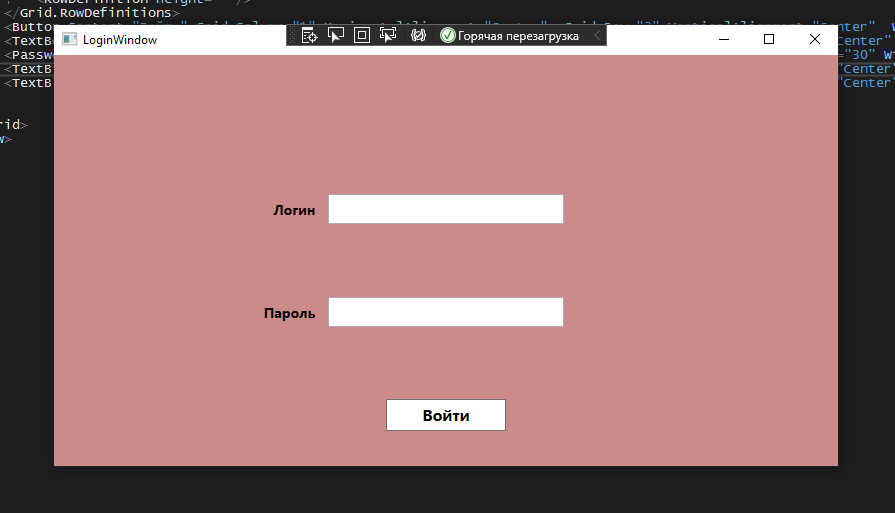
**Grid.Row**- номер строки в котором находится элемент (Отсчет с 0)

**VerticalAlignment**- выравнивание в ячейке по горизонтали (к верхнему краю, к нижнему краю, по центру)

**Width, Height** – Ширина и высота элемента.

**Задание**

1. Создать форму авторизации, как в примере, используя элементы: button, textbox, textblock, passwordbox. Расположить примерно таким же образом. На панели настроек элементов изменить размер шрифта до 14 и изменить цвет фона. Добавить заголовок окна. Посмотреть, как изменится код элементов при этом.



1. Для себя составить таблицу основных элементов и их свойств. Выучить как таблицу умножения. Свойства брать по ссылкам из этого документа или из другого удобного источника.

Подсказка, какие свойства записать и запомнить (но они есть не у всех элементов).

Background

Click

Content

FontFamily

FontSize

Height

Width

Text

GridRow

GridColumn

Name

HorizontalAlignment

VerticalAlignment

Style

SelectedIndex

DisplayMemberPath

Grid.ColumnSpan

Grid.RowSpan