Министерство образования Республики Беларусь

Учреждение образования

“Брестский государственный технический университет”

**Лабораторная работа №1**

**По дисциплине ТРиПС за 3 семестр**  
**Тема: «Введение в XAML и WPF»**

**Выполнил:**

Студент группы ПО-6(1)  
 21-го курса

Мартынович Даниил

**Проверила:**

Войцехович О. Ю.

Брест 2021

Задание 1:

В среде Microsoft Visual C# 2010 Express создайте проект «Приложение WPF».

Среда разработки предоставляет возможность графического и дескрипторного способов разработки пользовательского интерфейса, которые являются равнозначными. Дескрипторный файл MainWindow.xaml и кодовый файл MainWindow.xaml.cs дополняют друг друга при описании одного и того же содержимого - класса MainWindow в пространстве имен WpfApplication1, совпадающим с названием проекта.

Платформа WPF проектировалась в рамках концепции отделения дизайнерской части пользовательского интерфейса от кодовой части программирования функциональности. Дизайнерская часть проектируется декларативно, чаще всего - с помощью графического дизайнера формы, который в автоматическом режиме генерирует соответствующий синтаксически правильный дескрипторный код на языке XAML.

В заготовке дескрипторного XAML-кода видно, что корнем приложения является контейнер <Window>, в который в дальнейшем будут включены дочерние элементы. Все элементы WPF существуют в двух вариантах: дескрипторном и объектном. Объектное описание размещается в пространствах имен, подключаемых в кодовую часть проекта с помощью инструкции using для видимости компилятором. Дескрипторное описание находится в двух пространствах имен: обычном и расширенном. Эти пространства имен подключаются как значения атрибутов xmlns и xmlns:x в корневом дескрипторе проекта

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

Используемые URL-адреса не указывают на какой-либо документ или содержимое на веб-сервере, а используются лишь для определения уникальных пространств имен.

Разместите в коде XAML в содержимом элемента Grid следующий код:

<Button x:Name="Btn1"

HorizontalAlignment="Center"

VerticalAlignment="Center"

Width="150"

Height="30"

FontSize="17"

Content="Обычная кнопка"

Foreground="#006699"

Background="#f0f0f0"

BorderBrush="#303030" />

В приведенном выше примере для элемента Button было задано простое значение атрибута Background. Для этого был использован синтаксис

<ЭЛЕМЕНТ АТРИБУТ=”ЗНАЧЕНИЕ” />

Для задания значения атрибуты может быть использован другой синтаксис:

<ЭЛЕМЕНТ>

<ЭЛЕМЕНТ.АТРИБУТ>

ЗНАЧЕНИЕ\_АТРИБУТА

</ЭЛЕМЕНТ. АТРИБУТ>

</ЭЛЕМЕНТ>

Например, для задания того же значения атрибута Background можно записать:

<Button>

<Button.Background>

#f0f0f0

</Button.Background>

</Button>

Данный синтаксис используется для задания сложных значений атрибутов в виде дерева элементов. Пример задания для фона кнопки линейной градиентной заливки:

<Button.Background>

<LinearGradientBrush>

<LinearGradientBrush.GradientStops>

<GradientStop Color="LightBlue" Offset="0" />

<GradientStop Color="DarkBlue" Offset="1" />

</LinearGradientBrush.GradientStops>

</LinearGradientBrush>

</Button.Background>

Данное дерево элементов задает градиентную заливку с использованием двух цветов: LightBlue и DarkBlue. В атрибуте Offset указывается относительное значение от 0 до 1, соответствующее положению цвета на отрезке от начальной точки до конечной.

**Задание 1:**

<Window x:Class="ТРиПС\_лаб1.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:ТРиПС\_лаб1"

mc:Ignorable="d"

Title="MainWindow" Height="600" Width="800">

<Grid>

<Button Name="Btn1"

HorizontalAlignment="Center"

VerticalAlignment="Center"

Width="150"

Height="30"

FontSize="17"

Content="Обычная кнопка"

Foreground="White"

BorderBrush="#303030">

<Button.Background>

<LinearGradientBrush>

<LinearGradientBrush.GradientStops>

<GradientStop Color="LightBlue" Offset="0" />

<GradientStop Color="DarkBlue" Offset="1" />

</LinearGradientBrush.GradientStops>

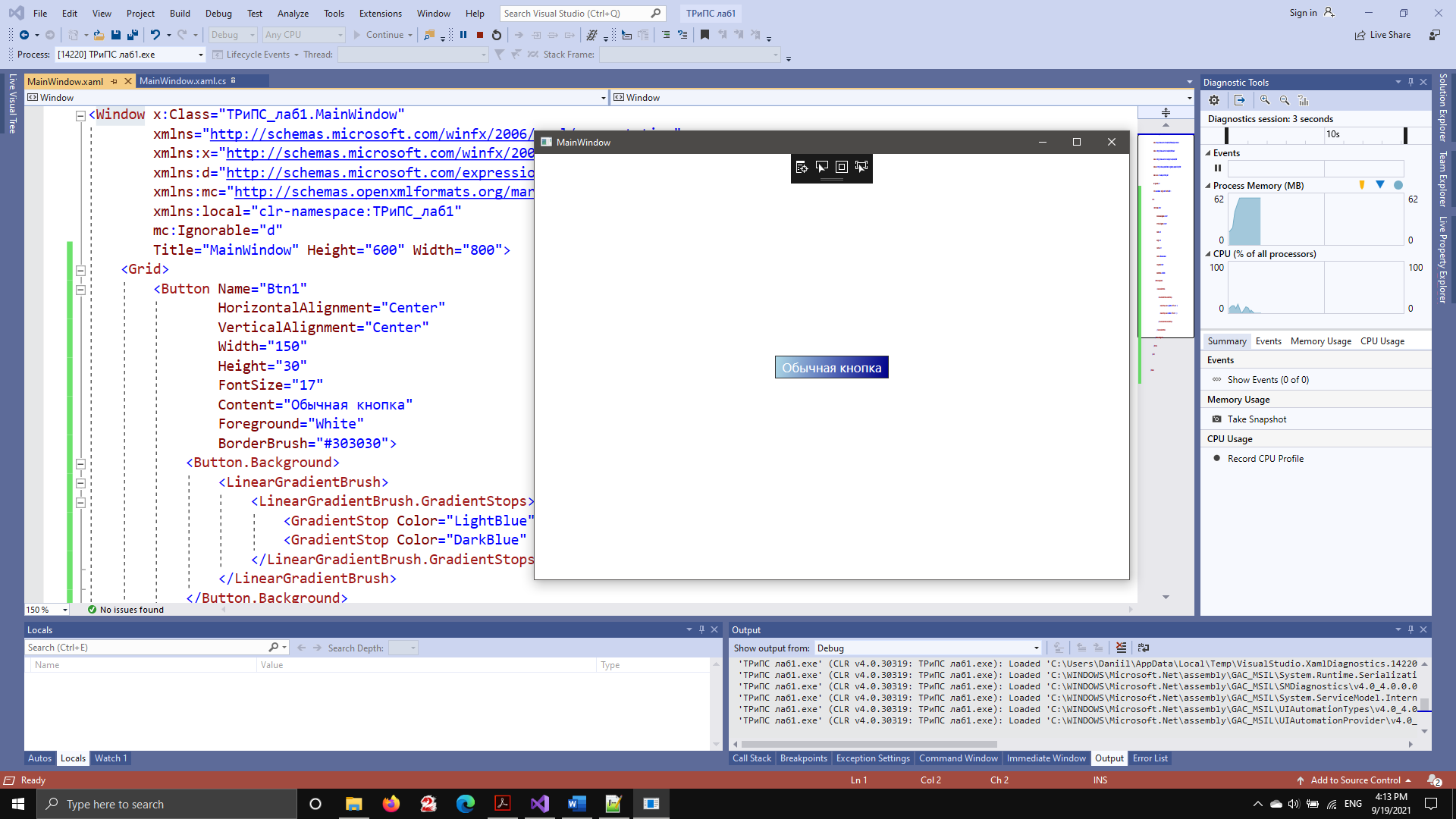
</LinearGradientBrush>

</Button.Background>

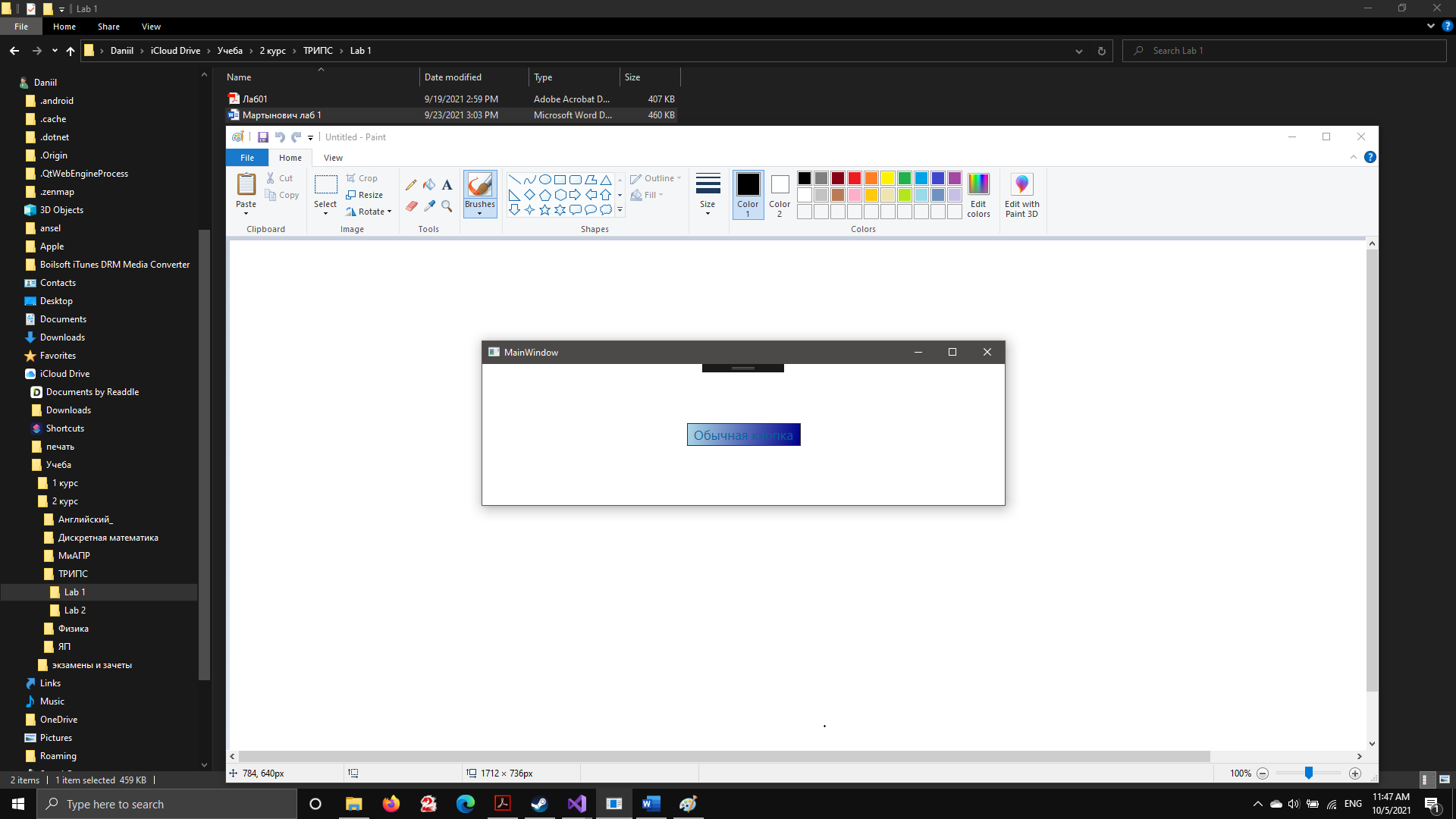
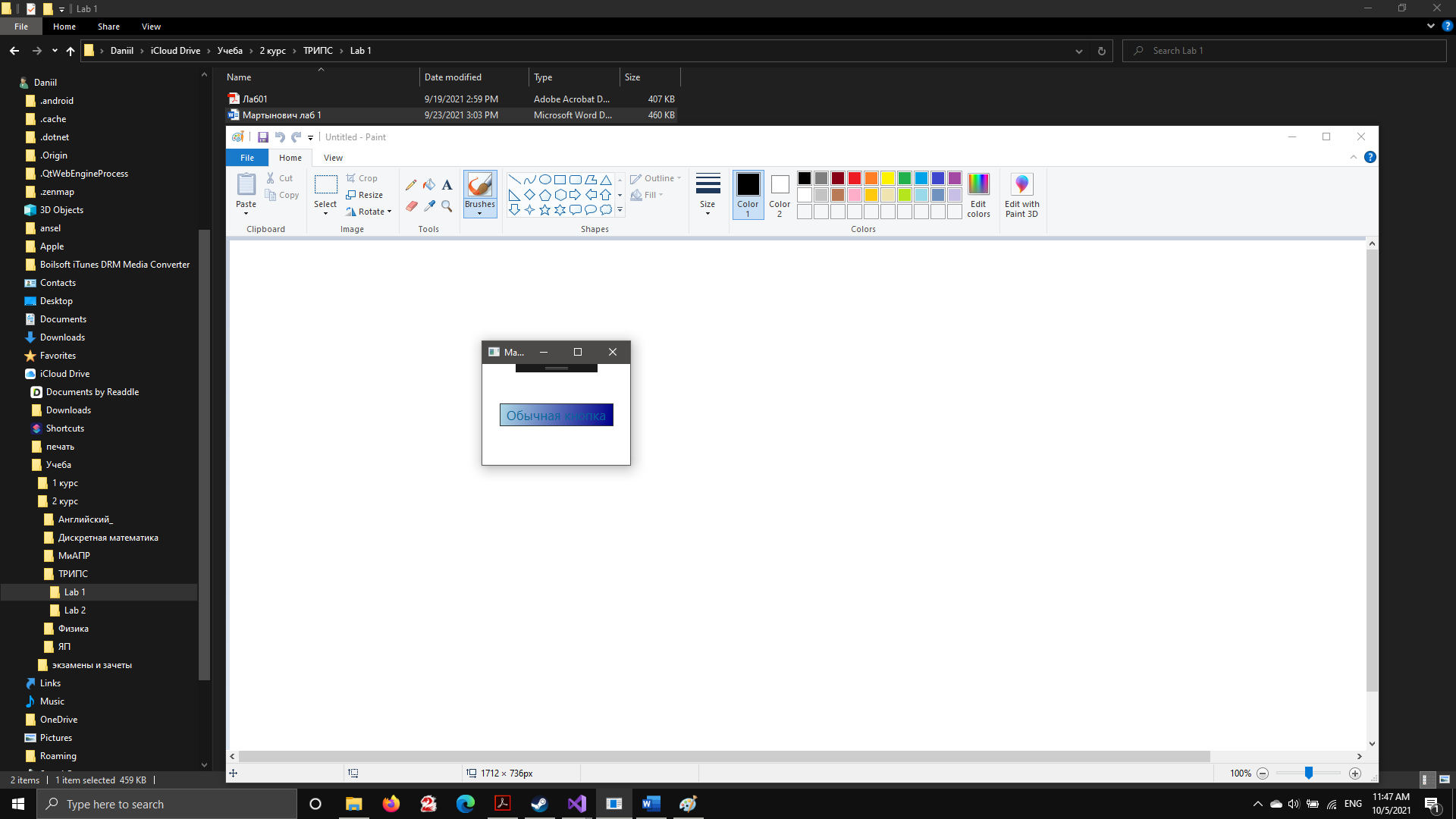
</Button>

</Grid>

</Window>



Запустите приложение и проверьте его поведение при изменении размеров окна:



При изменении размера окна, размер кнопки изменялся пропорционально относительно размеров окна.

**Задание 2:**

В XAML-коде для элемента Windows определите линейную градиентную заливку фона в соответствии с рисунком.

Используемые цвета: DarkBlue и LightBlue.

Необходимо указать четыре промежуточные точки со смещениями 0, 0.2, 0.8 и 1.

Для задания вертикальной заливки необходимо определить атрибуты StartPoint и EndPoint для элемента LinearGradientBrush. Значения этих атрибутов указываются в формате “X,Y”, где X – относительное значение (от 0 до 1) абсциссы точки, Y – относительное значение (от 0 до 1) ординаты точки. Начало координат находится в левом верхнем углу окна. По умолчанию значения атрибутов StartPoint и EndPoint следующие: StartPoint=”0,0” EndPoint=”1,1”.

Window x:Class="ТРиПС\_лаб1.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:ТРиПС\_лаб1"

mc:Ignorable="d"

Title="MainWindow" Height="600" Width="800">

<Window.Background>

<LinearGradientBrush StartPoint="1,0" EndPoint="1,1">

<GradientStop Color="DarkBlue" Offset="0.0" />

<GradientStop Color="LightBlue" Offset="0.2" />

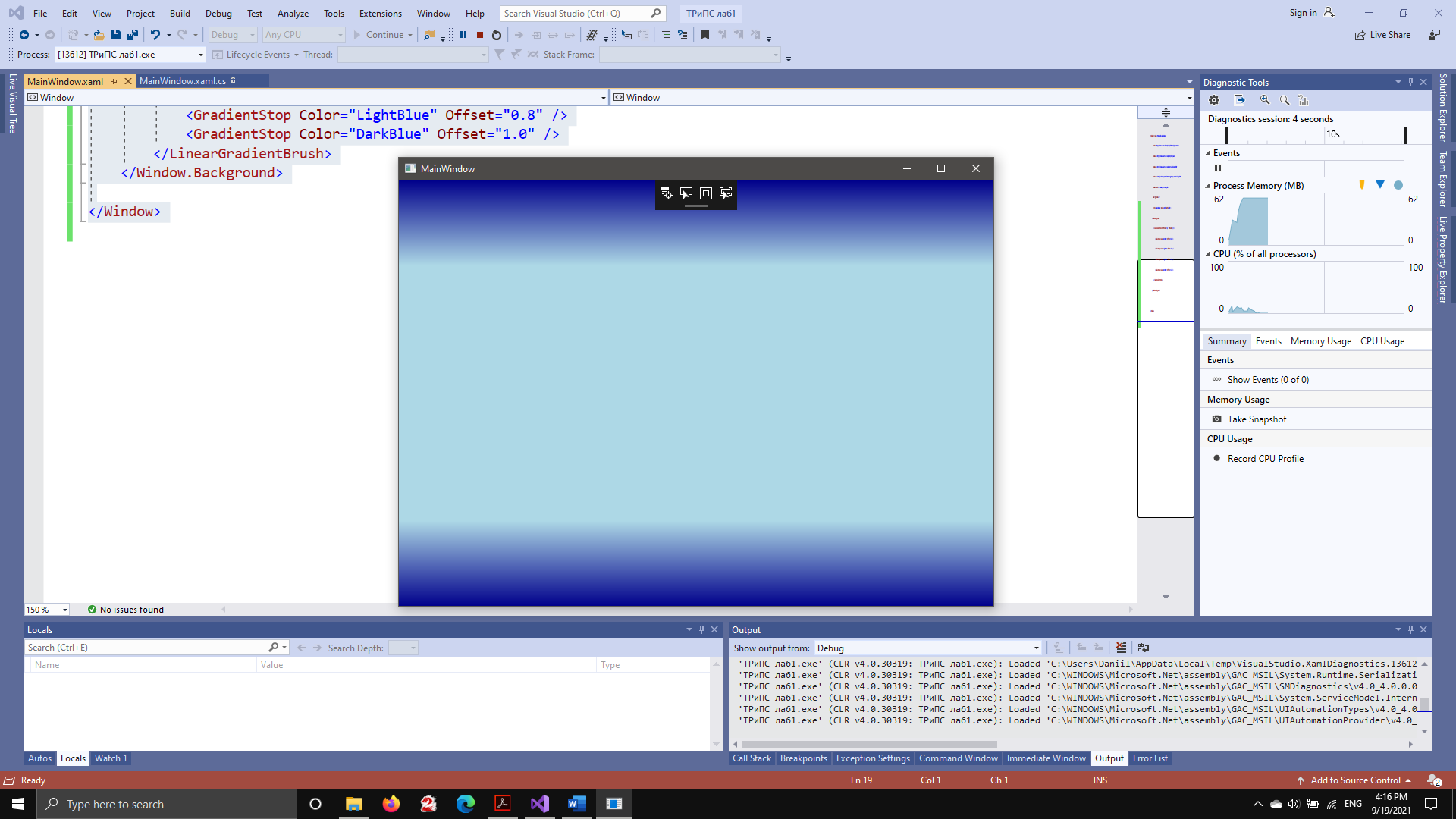
<GradientStop Color="LightBlue" Offset="0.8" />

<GradientStop Color="DarkBlue" Offset="1.0" />

</LinearGradientBrush>

</Window.Background>

</Window>



Вывод: Ознакомился с графической системой Windows Presentation Foundation предназначенной для создания пользовательских интерфейсов, 2D и 3D графики.