**Практическая работа**

**Тема: «Освоение базовых свойств.**  **Введение в разработку форм»**

**1. Цель работы**

Закрепить на практике способы задания свойств. Освоить влияние свойств на функциональные способности приложения. Повторить типы языка C#. Изучение методов построения форм Windows и получение навыков по настройке форм

**2. Ход выполнения**

1 Этап. Ознакомление с теоретическим материалом

1. Презентация «Свойства компонентов».
2. Раздаточный материал «№1. Общие свойства компонентов» и «№2. Специализированные свойства формы».

Формы Windows — это основной компонент пользовательского интерфейса. Формы предоставляют контейнер, который содержит элементы управления, меню и позволяет отображать приложение в уже привычной и единообразной модели. Формы могут реагировать на события мыши и клавиатуры, поступающие от пользователя, и выводить на экран данные для пользователя с помощью элементов управления, которые содержатся в форме.

Формы Windows содержат множество свойств, позволяющих настраивать их внешний вид и поведение. Просматривать и изменять эти свойства можно в окне Properties конструктора при разработке, а также программно во время выполнения приложения.

В следующей таблице перечислены некоторые свойства форм Windows, отвечающие за внешний вид и поведение приложения:

|  |  |
| --- | --- |
| **Свойство** | **Описание** |
| Name | Задает имя классу **Form**, показанному в конструкторе. Данное свойство задается исключительно во время разработки |
| BackColor | Указывает цвет фона формы |
| Enabled | Указывает, может ли форма принимать ввод от пользователя. Если свойству **Enabled** задано значение **False**, все элементы управления формы также блокируются |
| ForeColor | Указывает цвет переднего плана формы, то есть цвет выводимого текста. Если отдельно не указать значение свойства **ForeColor** элементов управления формы, они примут то же значение |
| FormBorderStyle | Указывает вид и поведение границы и строки заголовка формы  Значения свойства: |
| **None** - Форма не имеет границы, не может быть минимизирована или развернута до максимальных размеров и у нее нет экранной кнопки управления окном и кнопки справки |
| **FixedSingle** - Форма имеет тонкую границу, и размеры формы нельзя изменить во время выполнения. Форма  может быть минимизирована, развернута до максимальных размеров, и иметь кнопку справки или кнопку управления окном, что определяется остальными свойствами |
|  |
| **Fixed3D -** Форма имеет объемную границу, и размеры формы нельзя изменить  во время выполнения. Форма может быть минимизирована, развернута до максимальных размеров, и иметь кнопку справки или кнопку управления окном, что определяется остальными свойствами |
| **FixedDialog -** Форма имеет тонкую границу, и размеры формы нельзя изменить во время выполнения. У формы нет экранной кнопки управления окном, но может быть кнопка справки, что определяется остальными свойствами. Форму можно минимизировать и развернуть до максимальных размеров |
| **Sizable** - Форма имеет настройки по умолчанию, но они могут изменяться пользователем. Форма может быть минимизирована, развернута до максимальных размеров, и иметь кнопку справки, что определяется остальными свойствами |
| **FixedToolWindow -** Форма имеет тонкую границу, и размеры формы нельзя изменить во время выполнения.  Форма содержит только кнопку закрытия |
| **SizableToolWindow -** Форма имеет тонкую границу, и размеры формы могут быть изменены пользователем.  Форма содержит только кнопку закрытия |
| **Location** | Когда свойству **StartPosition** задано значение **Manual**, это свойство указывает исходное положение формы относительно верхнего левого угла экрана |
| **MaximizeBox** | Указывает, есть ли у формы кнопка **MaximizeBox** |
| **MaximumSize** | Устанавливает максимальный размер формы. Если задать этому  свойству размер 0; 0, у формы не будет верхнего ограничения размера |
| **MinimizeBox** | Указывает, есть ли у формы кнопка **MinimizeBox** |
| **MinimumSize** | Устанавливает минимальный размер формы, который пользователь может задать |
| **Opacity** | Устанавливает уровень непрозрачности или прозрачности формы от 0 до 100%. Форма, непрозрачность которой составляет 100%, полностью непрозрачна, а форма, имеющая 0 % непрозрачности, наоборот, полностью прозрачна |
| **Size** | Принимает и устанавливает исходный размер формы |
| **StartPosition** | Указывает положение формы в момент ее первого выведения на экран |
| **Text** | Указывает заголовок формы |
| **TopMost** | Указывает, всегда ли форма отображается поверх всех остальных форм,  свойству **TopMost** которых не задано значение **True** |
| **Visible** | Указывает, видима ли форма во время работы |
| WindowState | Указывает, является ли форма минимизированной, развернутой до максимальных размеров, или же при первом появлении ей задан размер, указанный в свойстве **Size** |

## **Упражнение 1*.*** Настройкапрямоугольнойформы ***Windows***

Созданиеновогопроекта

1. Откройте Visual Studio и создайте новый проект Windows Forms. Проект откроется с формой по умолчанию с именем **Form1** в конструкторе.
2. Выберите форму в конструкторе. Свойства формы отображаются в окне **Properties**.
3. В окне **Properties** задайте свойствам значения, как указано ниже:

|  |  |
| --- | --- |
| **Свойство** | **Значение** |
| Text | Trey Research |
| FormBorderStyle | Fixed3D |
| StartPosition | Manual |
| Location | 100; 200 |
| Opacity | 75% |

1. Перетащите три кнопки из **Toolbox** в форму и разместите их так, как вам будет удобно.
2. Поочередно выберите каждую кнопку и в окне **Properties** задайте свойству кнопок **Text** значения **Border Style, Resize** и **Opacity.**
3. Для кнопки **Border Style** задайте свойство **Anchor** – **Top, Left.** Реализацияобработчиковсобытий
4. В конструкторе дважды щелкните кнопку **Border Style**, чтобы открыть окно с кодом обработчика события **Buttonl Click.** Добавьте в этот метод следующую строку кода:

this.FormBorderStyle = FormBorderStyle.Sizable;

1. Возвратитесь в окно конструктора, дважды щелкните кнопку Resize и добавьте следующую строку:

this.Size = new Size(300, 500);

1. Возвратитесь в окно конструктора, дважды щелкните кнопку Opacity и добавьте следующую строку:

this.Opacity = 1;

**Запуск готового решения**

1. Для построения решения выберите меню **Build (**Построение**)**, далее команду **Build Solution (**Построитьрешение**)**. При наличии ошибок исправьте их и снова постройте решение. В дальнейшем при необходимости выбора последовательности действий очередность команд будет описываться, например, так: **Build |** **Build Solution.**
2. Нажмите Ctrl + F5 или выберите **Debug (**Отладка**) | Start Without Debugging (**Запускбезотладки**)**, чтобы запустить приложение. Щелкайте каждую кнопку и наблюдайте, как изменяется вид формы.
3. Измените поочередно расположение левой и верхней границ формы и сравните поведение кнопок внутри формы. Обратите внимание, что расстояние до этих границ от кнопки **Border Style** остаетсяпостоянным. Почему?

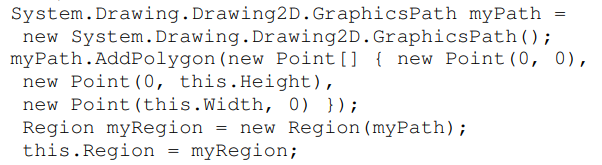
## **Упражнение *2.* Созданиенепрямоугольнойформы *Windows***

В этом упражнении вы создадите треугольную форму Windows.

1. Откройте Visual Studio и создайте новый проект Windows Forms. Проект откроется с формой по умолчанию с именем **Form1** в конструкторе.
2. В окне Properties задайте свойству **FormBorderStyle** значение **None,** а свойству **BackColor** значение **Red.** В этом случае форму легче будет увидеть при тестировании приложения.
3. Перетащите кнопку из Toolbox в левый верхний угол формы. Задайте свойству **Text** кнопки значение **Close Form.**
4. Дважды щелкните кнопку Close Form и добавьте в обработчик события **Button1 Click** следующий код:

this.Close();

1. В конструкторе дважды щелкните форму, чтобы открыть обработчик события **Form1 Load.** Добавьте в этот метод следующий код (он задает области формы треугольную форму указанием многоугольника с тремя углами):



1. Постройте и запустите приложение. Появится треугольная форма.

Задание 2.

1. Прочитать задание на практическую работу. Демонстрационный пример выполненного задания можно увидеть на скриншоте ниже.
2. Изучить фрагменты программного кода для реализации функционала задания.

**3. Задание на практическую работу**

На форму добавить компоненты и настроить их следующим образом:

1. Контейнеры (класс GroupBox). Настроить заголовок каждого контейнера (свойство *Text*) для пояснения их назначения.
2. Расположить в каждом контейнере нужное количество кнопок (класс Button) для реализации поставленных действий. Для каждой кнопки настроить свойство *Name* именем, которое соответствует назначению кнопки. В свойстве *Text* задать заголовок кнопки.
3. Невизуальный компонент Локальное меню (contextMenuStrip). Связать этот компонент с формой через ее свойство *ContexMenuStrip*.
4. Невизуальный компонент ToolTip для организации всплывающих подсказок для элементов интерфейса.
5. Создать глобальные переменные (уровень класса) для задания размера (класс Size) и расположения (класс Point) формы.

**На «Удовлетворительно»**

1. В отдельном контейнере реализовать действия, связанные с расположением формы относительно экрана: во всех четырех углах и по центру. Использовать текущие настройки разрешения монитора и свойства *Location* или *Left*, *Top* формы и объект класса Point.
2. Обеспечить контроль над границами размера формы – установить предельные размеры.
3. Запретить пользователю менять размеры формы.
4. В отдельном контейнере реализовать действия, связанные с состоянием формы: нормальное, развернуто, минимизировано (использовать свойство *WindowState* формы), фиксированного размера и изменяемого размера (использовать свойство *FormBorderStyle* формы).
5. В отдельном контейнере реализовать действия, связанные с размером формы: 800х400, 1000х500, «Стандартный» (размер формы при запуске приложения). Использовать свойство *Size* формы и объект класса Size.
6. Создать собственный курсор. Установить его для формы.
7. К каждому элементу интерфейса добавить всплывающую подсказку.
8. Создать собственную иконку. Установить в заголовок формы.
9. В отдельном контейнере вывести текущие характеристики: разрешение монитора, размер формы, размеры рабочего стола.

**На «Хорошо»**

1. В локальном меню формы создать пункт «Цвета» с подпунктами «Фон», «Шрифт», «Кнопки», «Панели». В нем создать подпункты с несколькими цветами. Для каждого подпункта создать обработчик события Click, в котором окрашивать форму выбранным цветом.
2. В локальном меню формы создать пункт «Тема», с помощью которого провести цветовые настройки сразу комбинацией подобранных цветов.
3. В локальном меню формы создать пункт «Тип курсора». В нем создать подпункты с несколькими типами курсора. Для каждого подпункта создать обработчик события Click, в котором устанавливать для формы курсор. Использовать свойство *Cursor* компонентов.
4. В локальном меню формы создать пункт «Прозрачность формы». В нем создать подпункты: 20%, 50%, 100%. Для каждого подпункта создать обработчик события Click, в котором устанавливать прозрачность формы. Использовать свойство *Opacity* формы.
5. В локальном меню формы создать пункт «Разрешить/Запретить менять размеры формы».
6. Удалить кнопки системного меню.
7. Добавит кнопку «Выход».

**На «Отлично»**

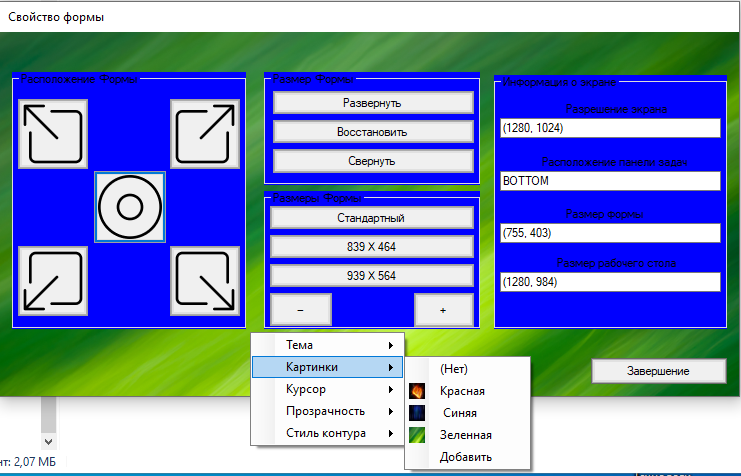
1. При изменении размера формы, выполнять растяжение и изменение разметов контейнеров так, чтобы не нарушились пропорции интерфейса. Использовать свойства *Anchor* и *Dock*.
2. Предоставить возможность разместить на поверхности формы картинку. Выбор картинки организовать через пункты локального меню. Использовать свойство *BackGroundImage* формы.

**Подсказка**

int ws = Screen.PrimaryScreen.WorkingArea.Width; //Ширина рабочего стола

int hs = Screen.PrimaryScreen.WorkingArea.Height; //Высота рабочего стола

Size resolution = Screen.PrimaryScreen.Bounds.Size; //Разрешение экрана



**4. Фрагмент кода**

public partial class Form1 : Form

{

int wf, hf; //Глобальные переменные ширины и высота формы

//Задать текущие размеры формы для ее восстановления

private void Form1\_Load(object sender, EventArgs e)

{

wf = this.Width;

hf = this.Height;

}

//Перенести форму в центр экрана

private void buttonCenter\_Click(object sender, EventArgs e)

{

this.Left = Screen.PrimaryScreen.WorkingArea.Width/2 - this.Width/2;

this.Top = Screen.PrimaryScreen.WorkingArea.Height/2 - this.Height/2;

}

//Размер формы уменьшать на 10 с проверкой допустимых размеров

private void buttonLower\_Click(object sender, EventArgs e)

{

if (this.Width > 400)

{

this.Width = this.Width - 10;

this.Height = this.Height - 10;

}

}