УДК 004.021

ОСОБЕННОСТИ СОЗДАНИЯ ГРАФИЧЕСКИХ ИНТЕРФЕЙСОВ С ПОМОЩЬЮ ТЕХНОЛОГИЙ WINDOWS FORMS

М.А. ТРАЩЕНКО
(Представлено: канд. техн. наук, доц. И.Б. БУРАЧЁНОК)

Рассматриваются принципы построения графического интерфейса пользователя. Рассмотрены положительные стороны использования Windows Forms, его возможности для быстрого и гибкого в разработке интерфейса. Разобраны часто используемые при реализации графических интерфейсов свойства графических элементов.

Для создания графических интерфейсов с помощью платформы .NET применяются разные технологии – Window Forms, WPF, приложения для магазина Windows Store (для ОС Windows 8/8.1/10). Однако наиболее простой и удобной платформой до сих пор остается Window Forms или просто формы.

В представленной статье подробно остановимся на описании создания современных графических интерфейсов с помощью технологий Windows Forms.

Приложение Windows Forms представляет собой событийно-ориентированное приложение, поддерживаемое Microsoft .NET Framework. В отличие от пакетных программ, при работе приложения большая часть времени тратится на ожидание от пользователя каких-либо действий, как, например, ввод текста в текстовое поле или клика мышкой по кнопке.

Форма – это панель, на которой выводится информация для пользователя. Обычно приложение Windows Forms строится путем помещения элементов управления на форму и написания кода для реагирования на действия пользователя, такие как щелчки мыши или нажатия клавиш. Элемент управления – это отдельный элемент пользовательского интерфейса, предназначенный для отображения или ввода данных. При выполнении пользователем какого-либо действия с формой или одним из ее элементов управления создается событие. Приложение реагирует на эти события с помощью кода и обрабатывает события при их возникновении.

Большим недостатком Windows Forms является то, что командам программистов и дизайнеров приходиться работать очень тесно, чтобы получился достойный проект. То есть, дизайнер рисует интерфейс, а программист его реализует. Причем программисту приходиться постоянно отвлекаться от своей непосредственной задачи – написания логики программы. Он то и дело, подгоняет кнопки под размеры формы, вставляет картинки и т.п. Чтобы избавиться от этого недостатка Windows Forms, в .NET Framework включена технология Windows Presentation Foundation (WPF), являющаяся большим шагом в сторону улучшения разработки интерфейсов. По сравнению с Windows Forms – было сделано следующее: программист полностью погружается в разработку логики программы, а дизайнер может сразу же создавать дизайн программы, практически не зависимо от программиста.

Популярность Windows Forms постепенно падает, однако ее продолжают использовать в простых, не требующих особых представлений, интерфейсах, программах. Также во всех версиях .NET Framework существует поддержка Windows Forms, куда включены некоторые дополнения и улучшения.

Несмотря на то, что в Windows Forms весь интерфейс можно построить только с использованием мыши, программный код разметки также доступен для программиста, и при желании, может быть изменен.

Пример кода дизайнера Windows Forms представлен в листинге.

```
namespace HelloApp
{
    partial class Form1
    {
        private System.ComponentModel.IContainer components = null;
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
```

```
                base.Dispose(disposing);
            }
            #region Windows Form Designer generated code
            private void InitializeComponent()
            {
                this.SuspendLayout();
                this.AutoScaleDimensions = new System.Drawing.SizeF(6F,
13F);
                this.AutoScaleMode = Sys-
tem.Windows.Forms.AutoScaleMode.Font;
                this.ClientSize = new System.Drawing.Size(284, 261);
                this.Name = "Form1";
                this.Text = "Привет мир!";
                this.ResumeLayout(false);
            }
            #endregion
        }
    }
```

Листинг. – Пример кода дизайнера Windows Forms

С помощью специального окна Properties, представленного на рисунке, среда разработки Visual Studio предоставляет нам удобный интерфейс для управления свойствами элемента.

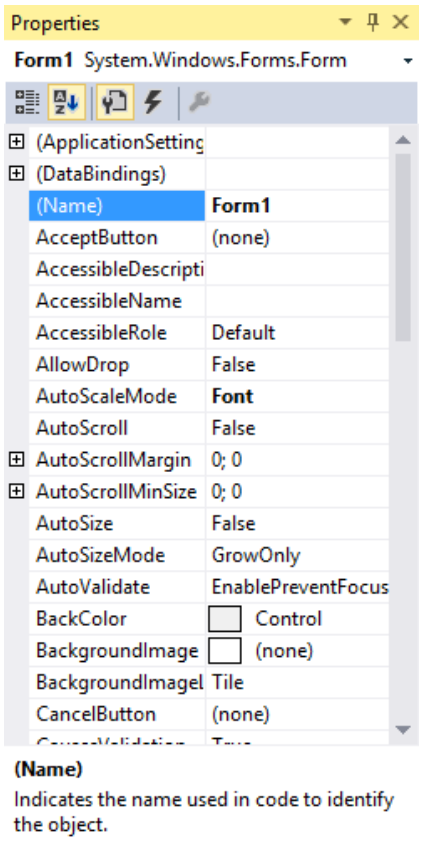

Рисунок. – Окно свойств элемента

Большинство этих свойств оказывает влияние на визуальное отображение формы. Разберем основные свойства:

– Name: устанавливает имя формы – точнее имя класса, который наследуется от класса Form;

– BackColor: указывает на фоновый цвет формы. Щелкнув на это свойство, мы сможем выбрать тот цвет, который нам подходит из списка предложенных цветов или цветовой палитры;

– BackgroundImage: указывает на фоновое изображение формы;
– BackgroundImageLayout: определяет, как изображение, заданное в свойстве BackgroundImage, будет располагаться на форме;
– ControlBox: указывает, отображается ли меню формы;
– Cursor: определяет тип курсора, который используется на форме;
– Enabled: если данное свойство имеет значение false, то она не сможет получать ввод от пользователя, то есть мы не сможем нажать на кнопки, ввести текст в текстовые поля и т.д.;
– Font: задает шрифт для всей формы и всех помещенных на нее элементов управления. Однако, задав у элементов формы свой шрифт, мы можем тем самым переопределить его;
– ForeColor: цвет шрифта на форме;
– FormBorderStyle: указывает, как будет отображаться граница формы и строка заголовка. Устанавливая данное свойство в None можно создавать внешний вид приложения произвольной формы;
– HelpButton: указывает, отображается ли кнопка справки формы;
– Icon: задает иконку формы;
– Location: определяет положение по отношению к верхнему левому углу экрана, если для свойства StartPosition установлено значение Manual;
– MaximizeBox: указывает, будет ли доступна кнопка максимизации окна в заголовке формы;
– MinimizeBox: указывает, будет ли доступна кнопка минимизации окна;
– MaximumSize: задает максимальный размер формы;
– MinimumSize: задает минимальный размер формы;
– Opacity: задает прозрачность формы;
– Size: определяет начальный размер формы;
– StartPosition: указывает на начальную позицию, с которой форма появляется на экране;
– Text: определяет заголовок формы;
– TopMost: если данное свойство имеет значение true, то форма всегда будет находиться поверх других окон;
– Visible: видима ли форма, если мы хотим скрыть форму от пользователя, то можем задать данному свойству значение false;
– WindowState: указывает, в каком состоянии форма будет находиться при запуске: в нормальном, максимизированном или минимизированном.

Таким образом, принято решение при разработке графического интерфейса для автоматизированной информационной системы учета товаров с целью продвижения малого бизнеса компании «GRANDDECOR» использовать технологии Windows Forms, что позволит сократить временные затраты на разработку интерфейса и уделить больше внимания логике автоматизированной информационной системы.

ЛИТЕРАТУРА

1. Metanit [Электронный ресурс] // Руководство по программированию в Windows Forms. – Режим доступа: https://metanit.com/sharp/windowsforms/. – Дата доступа: 13.09.18.
2. CitForum [Электронный ресурс] // Пошаговые руководства по Windows Forms. – Режим доступа: https://msdn.microsoft.com/ru-ru/library/zftbwa2b(v=vs.110).aspx/. – Дата доступа: 14.09.18.