

Учебный курс
**Технологии и средства разработки
корпоративных систем**

Лекция 5

**Создание графического интерфейса
пользователя**

Лекции читает

кандидат технических наук, доцент

Зыков Сергей Викторович

Технология Windows Forms

Содержание

- Основные понятия (WinForms, Smart Client и др.)
- Элементы интерактивных интерфейсов WinForms
- Обработка событий и создание кода для нее
- Основные типы элементов управления в WinForms
- Создание пользователями собственных элементов управления
- Инструмент Windows Forms Designer среды Visual Studio
- Особенности и преимущества WinForms
 - Отображение и манипулирование данными
 - Развертывание приложений
 - Другие особенности
- Литература

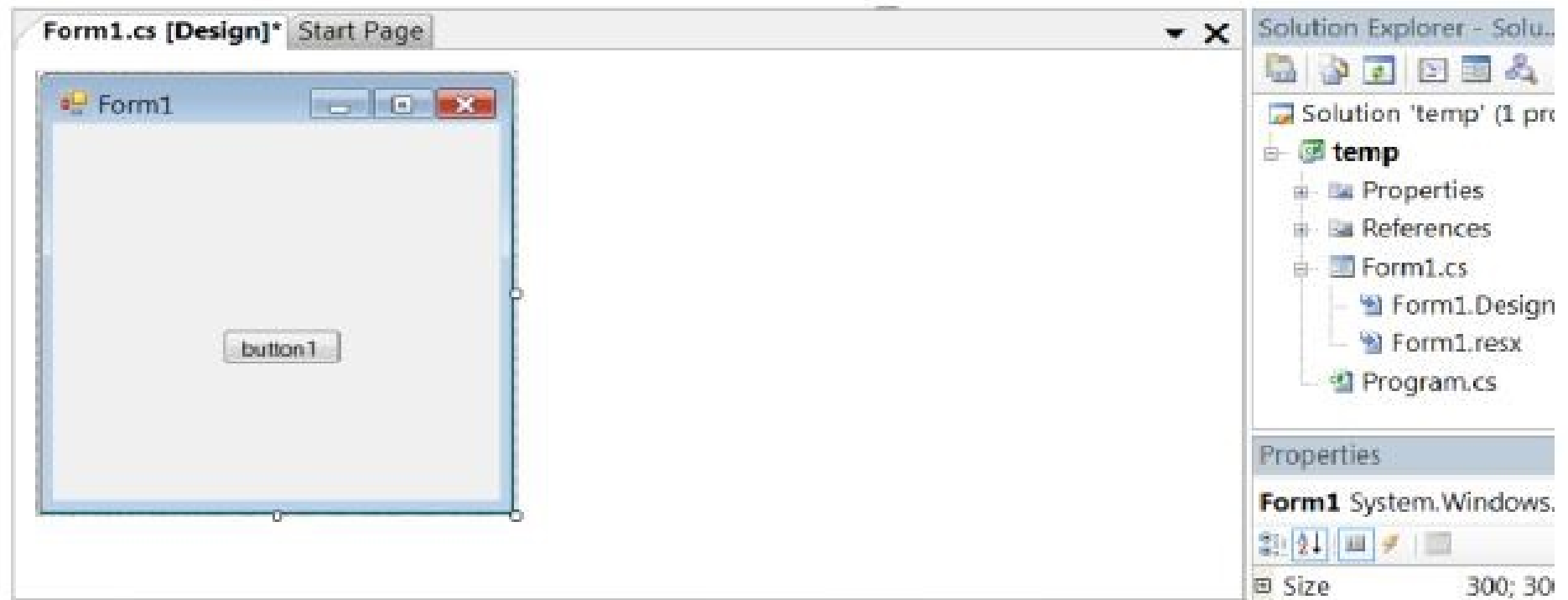
Windows Forms и Smart Client-приложения

- Windows Forms - технология для платформы .NET Framework в форме набора библиотек, упрощающих выполнение типичных задач приложений (чтение и запись в файловую систему и т.п.).
- Возможности приложений Windows Forms в среде разработки Microsoft Visual Studio .NET:
 - вывод информации;
 - ввод данных пользователем;
 - обмен информацией с удаленными компьютерами через сетевое соединение

Win Forms: Элементы интерактивных интерфейсов (1)

- Форма – визуальная поверхность, на которой отображается информация для пользователя.
- Контрол (элемент управления) – это элемент пользовательского интерфейса, отображающий данные или принимающий ввод данных.
- Создание приложения Win Forms – процесс добавления контролов на форму и описания реакций на действия пользователя, (щелчок мыши, нажатие клавиши и т.д.).

Win Forms: Элементы интерактивных интерфейсов (2)

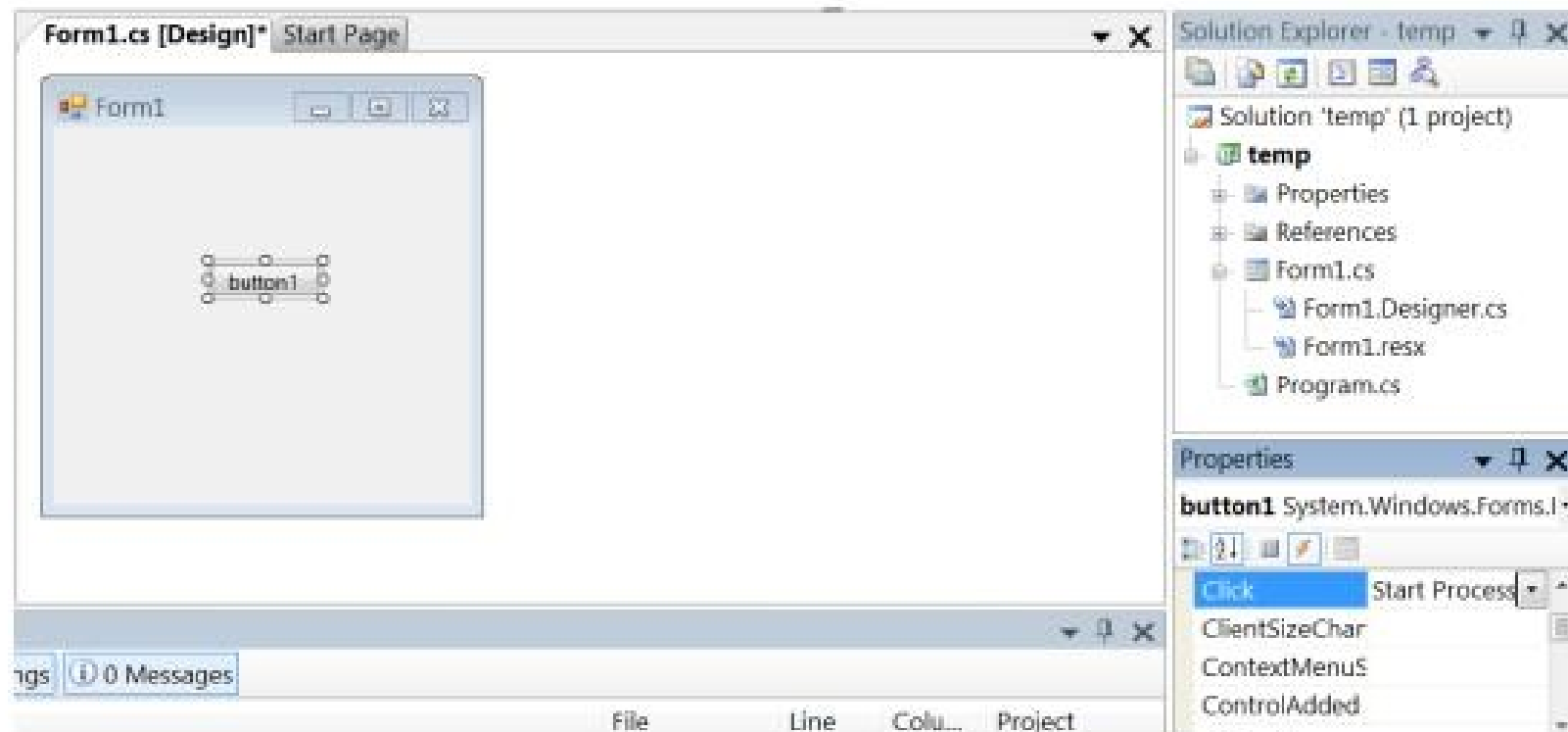


Пример создания приложения `Form1` по технологии Win Forms в среде Visual Studio с элементом управления – кнопкой `button1`

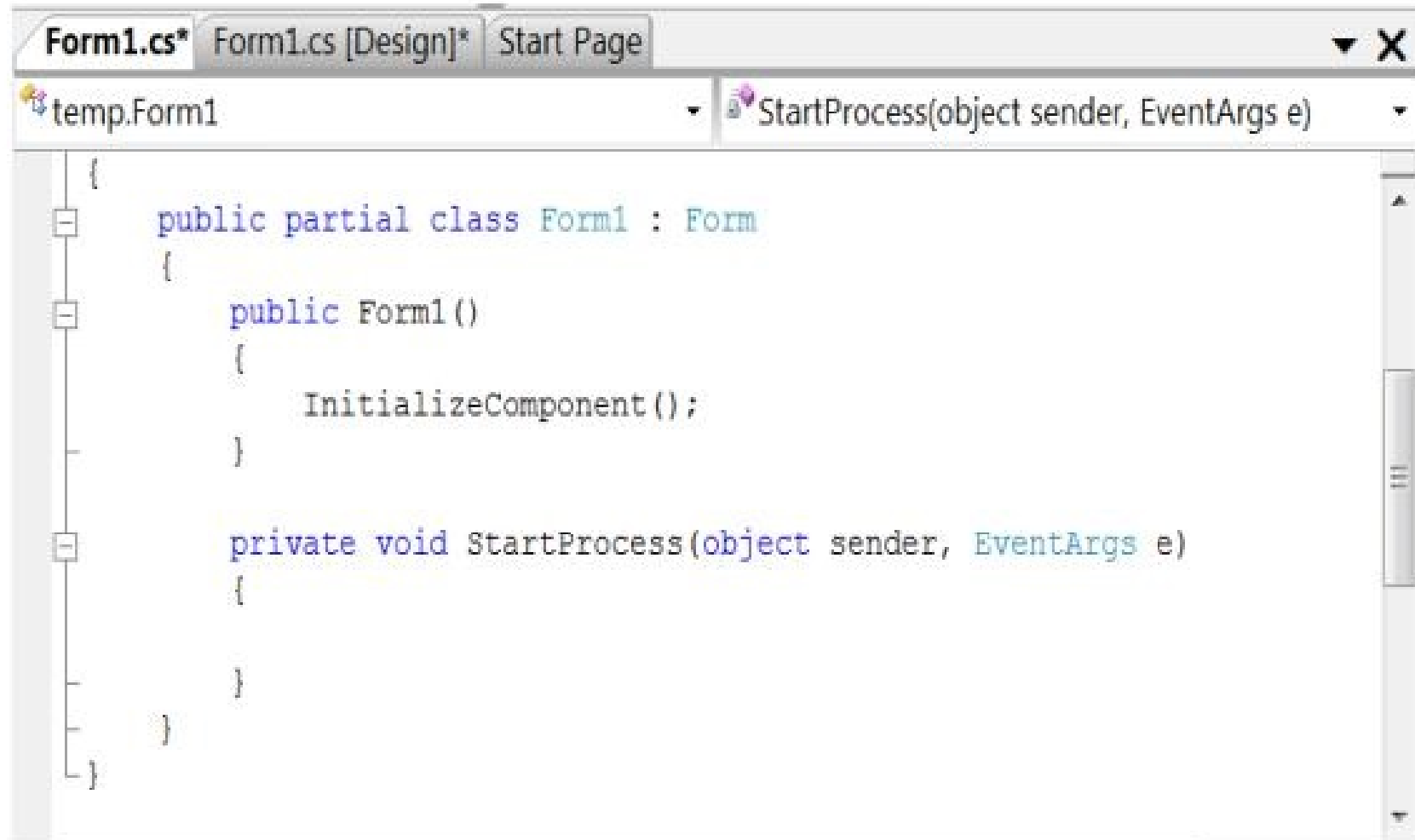
Обработка события

При взаимодействии пользователя с формой или контролом, генерируется событие (event).

Приложение реагирует на событие с помощью кода - обрабатывает его, как только оно происходит.



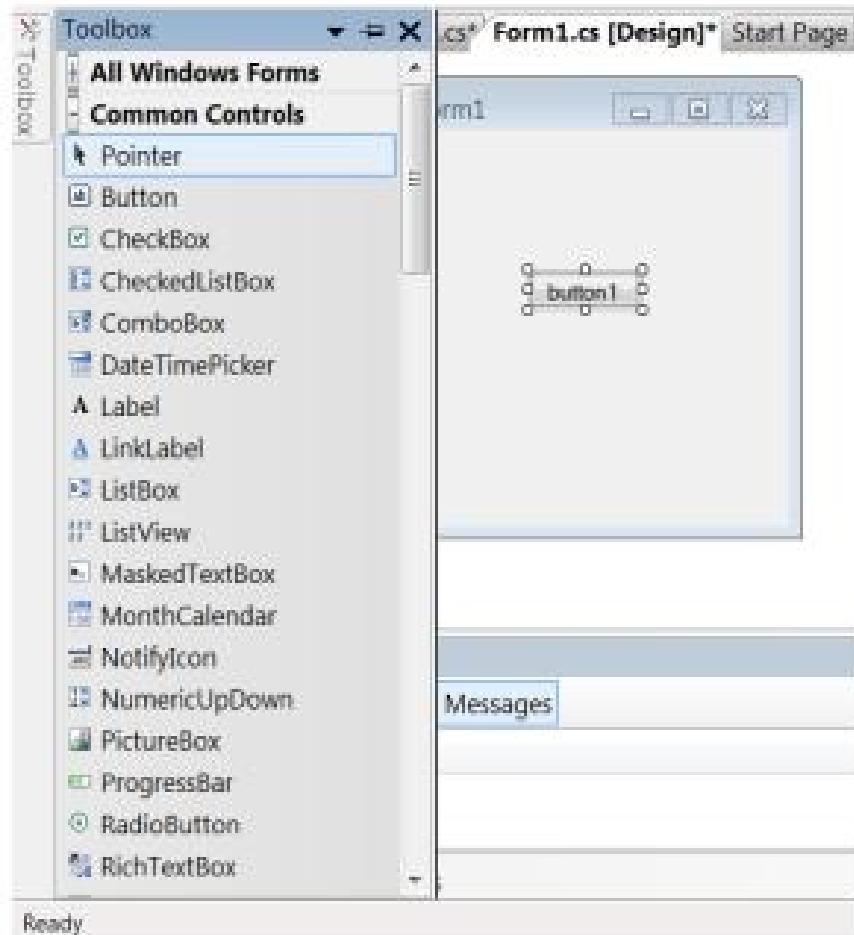
Код обработки события на языке C#



```
Form1.cs* Form1.cs [Design]* Start Page
temp.Form1 StartProcess(object sender, EventArgs e)
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void StartProcess(object sender, EventArgs e)
        {
        }
    }
}
```


Разнообразие элементов управления (контролов)



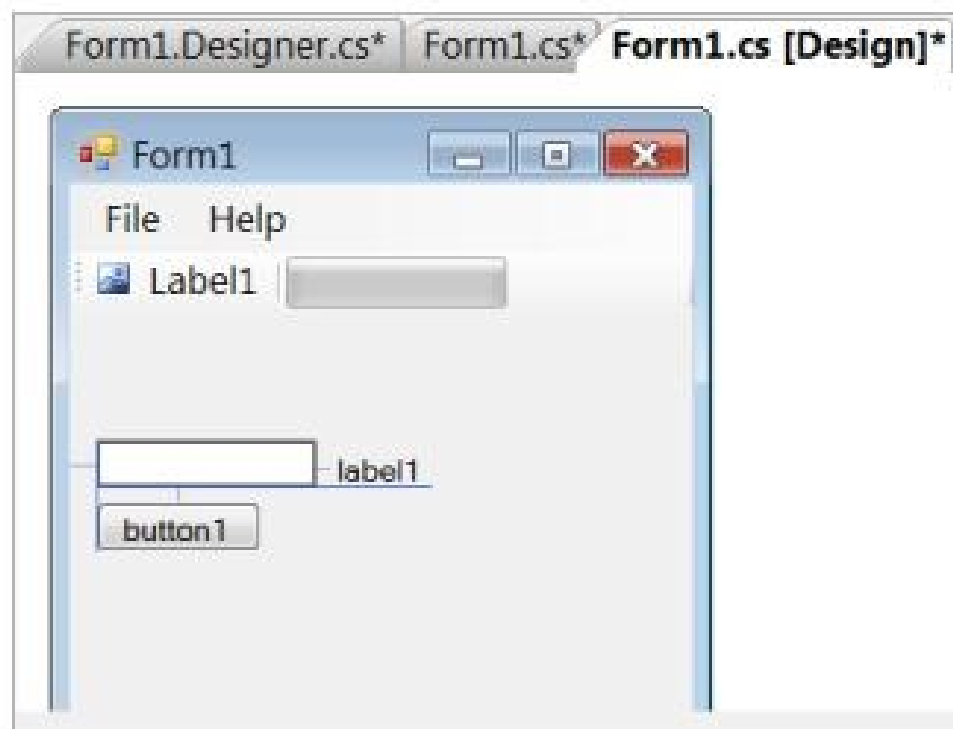
Windows Forms содержит широкий спектр контролов различных видов, которые можно разместить на форму:

- Текстовое поле (text box);
- Кнопка (button);
- Выпадающий список (drop-down box);
- Переключатель (radio button);
- Web-страница.

Пользовательские элементы управления

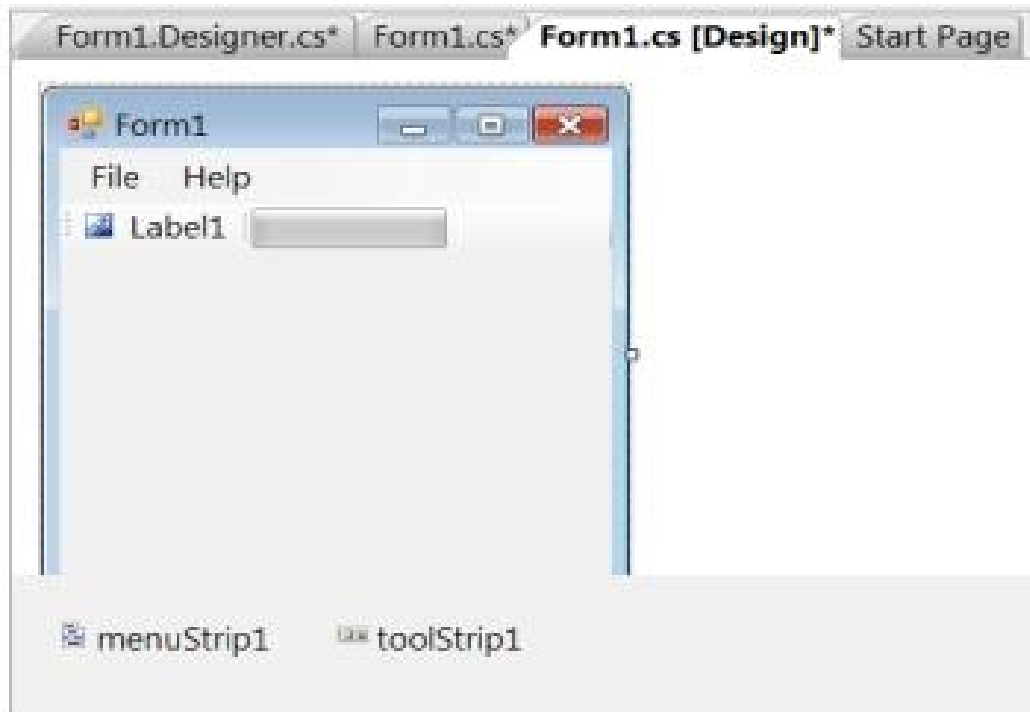
- Формы ввода и распределенные приложения поддерживает возможность создания пользователями / разработчиками собственных элементов управления
- Для этого используется класс `UserControl` библиотеки `Microsoft .NET Framework`
- Кроме того, возможно использование свойств уже реализованных существующих элементов управления посредством наследования

Windows Forms Designer



- С помощью инструментария Windows Forms Designer в среде Visual Studio можно:
- создавать Windows Forms-приложения посредством мыши, «перетаскивая» (drag-and-drop) нужные контролы на Windows-форму.
 - выравнивать контролы относительно друг друга и Windows-формы.

Создание эргономичных, интерактивных интерфейсов



В Windows Forms встроены контролы пользовательского интерфейса, реализующие важные особенности приложений Microsoft Office

Так, контролы ToolStrip и MenuStrip помогают создавать панели инструментов и меню, которые содержат:

- текст
- Изображения
- Подменю
- другие контролы (textbox, и combobox и др.)

Создание собственных элементов интерфейса

Для создания собственных элементов интерфейса пользователя, разработчиками применяются классы системного пространства имен `System.Drawing` из `Microsoft .NET Framework`, позволяющие осуществлять непосредственно на форме прорисовку:

- линий,
- окружностей,
- других видов графических объектов.

Отображение данных из гетерогенных источников (1)

Windows Forms предоставляет контрол DataGridView для отображения данных в форме таблиц из следующих источников данных:

- базы данных;
- файлы формата XML;
- Web-сервисы формата XML;
- другие источники данных.

Возможности отображения данных в DataGridView:

- настройка представления отдельных ячеек таблицы;
- фиксация отдельных строк/столбцов;
- отображение сложных контролов внутри ячеек;
- другие особенности.

Отображение данных из гетерогенных источников (2)

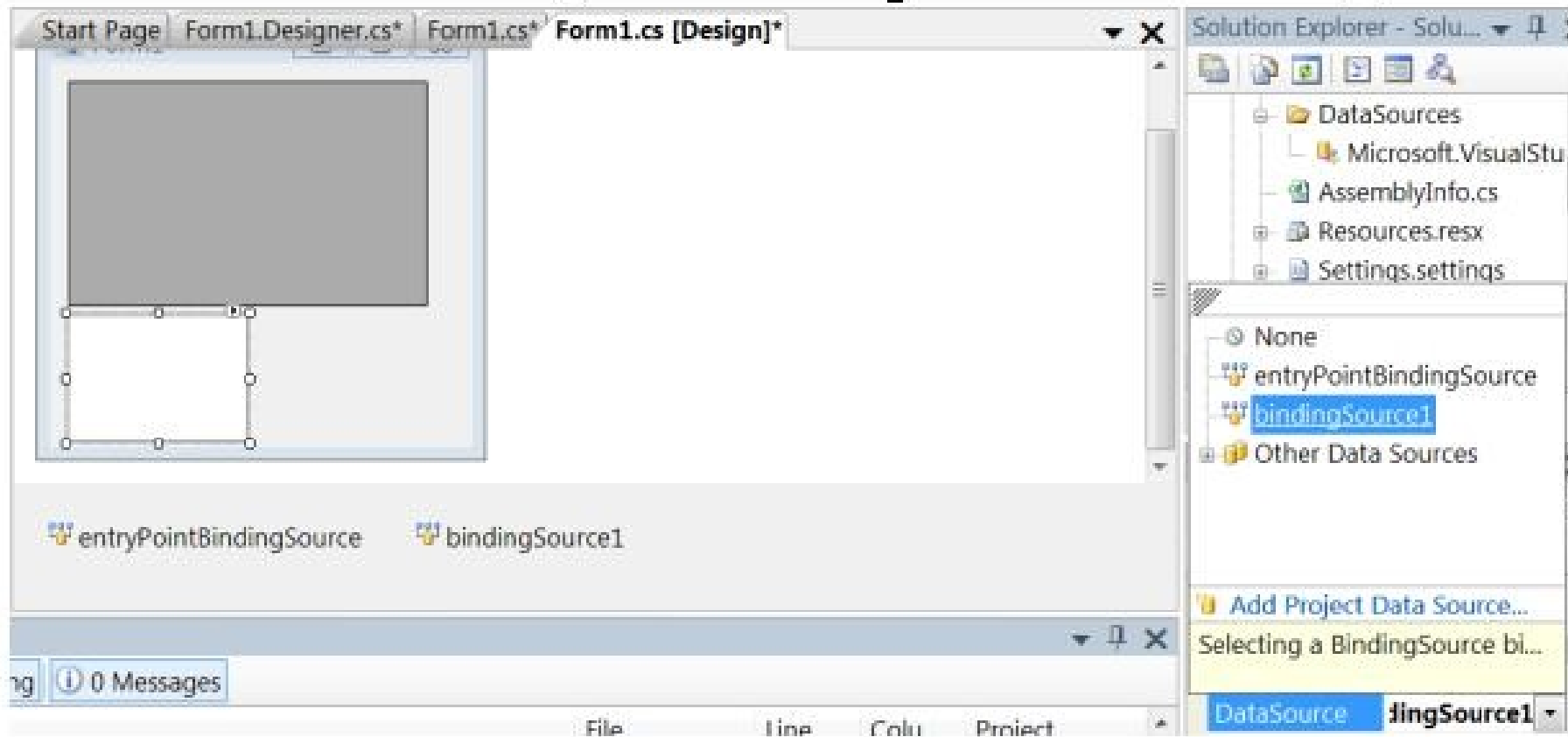


Пример использования DataGridView в Windows Forms для отображения табличном формате из гетерогенных источников данных

Связь с источником данных через сетевое соединение (1)

- Технология «интеллектуальных клиентов» (Smart clients) для Windows Forms обеспечивает связь с источником данных через сетевое соединение.
- Компонент BindingSource среды Microsoft .NET Framework позволяет управлять:
 - параметрами соединения с источником данных;
 - организацией связи данных с контролами;
 - навигацией между записями источника данных;
 - редактирования записей источника данных;
 - записи изменений в источник данных.
- Контроль BindingNavigator предоставляет разработчикам интерфейс к компоненту BindingSource для визуальной обработки записей данных

Связь с источником данных через сетевое соединение (2)



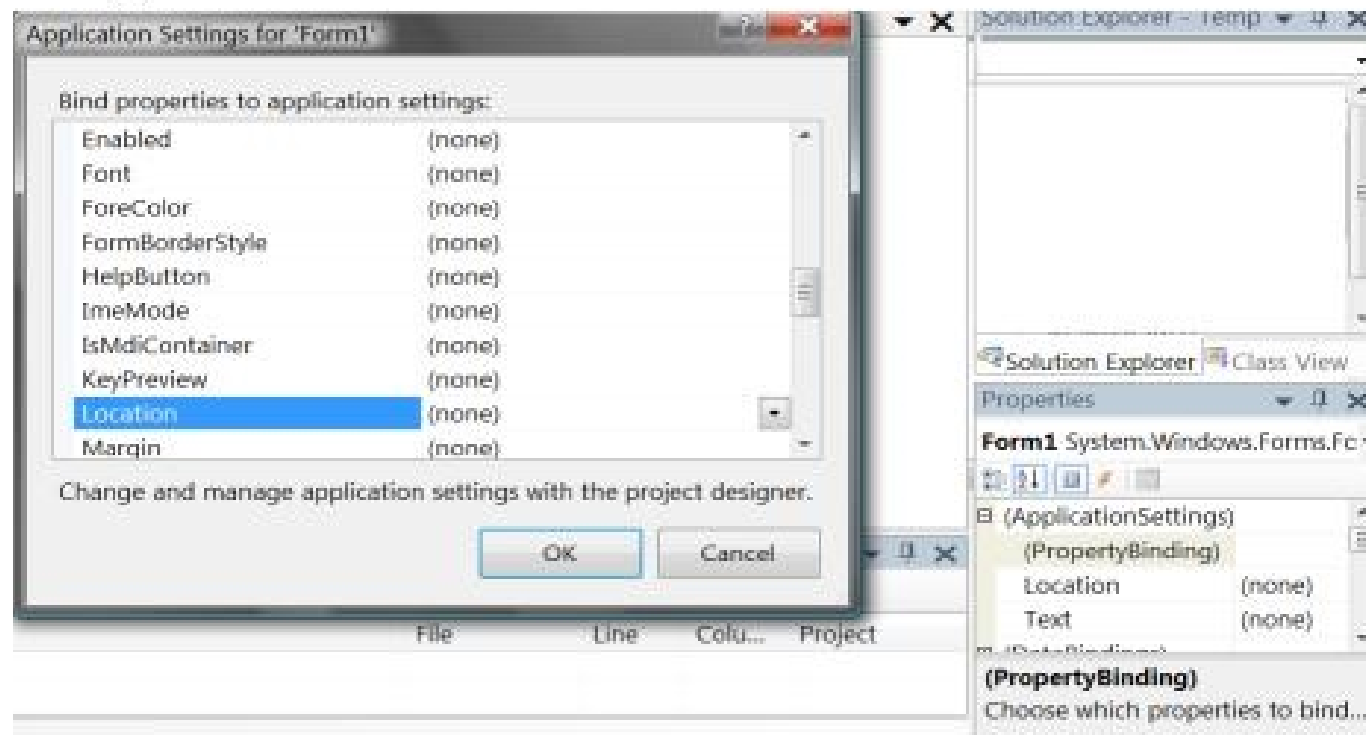
Пример использования элемента управления BindingNavigator в Windows Forms для организации визуального интерфейса пользователя с гетерогенными источниками данных

Application Settings: альтернативный способ связи с гетерогенными источниками данных (1)

Application Settings - альтернативный способ связи разработчика smart client-приложений с гетерогенными источниками данных по технологии Windows Forms

- **Формат хранения информации:** XML
- **Состав – *состояние*** приложения, в т.ч.:
 - последний линейный размер формы на экране;
 - персональные предпочтения пользователя;
 - место хранения файлов по умолчанию;
 - другие параметры.
- **Особенность:** автоматическая загрузка в память во время выполнения приложения

Application Settings: альтернативный способ связи с гетерогенными источниками данных (2)



Пример настройки элемента управления Application Settings для визуального интерфейса пользователя smart client приложения (определение места хранения файлов)

Развертывание приложений (1)

Развертывание приложения – его установка пользователем на своем (клиентском) компьютере после завершения разработки для последующего использования

Проблемы:

- массовая рассылка,
- упрощение установки приложений

Решение: технология ClickOnce – развитие механизма сборок (assembly).

Достоинства:

- Интернет-ориентированность (пользователю предоставляется URL приложения)
- Простота использования – развертывание приложения из Visual Studio одним или несколькими щелчками «мышь».

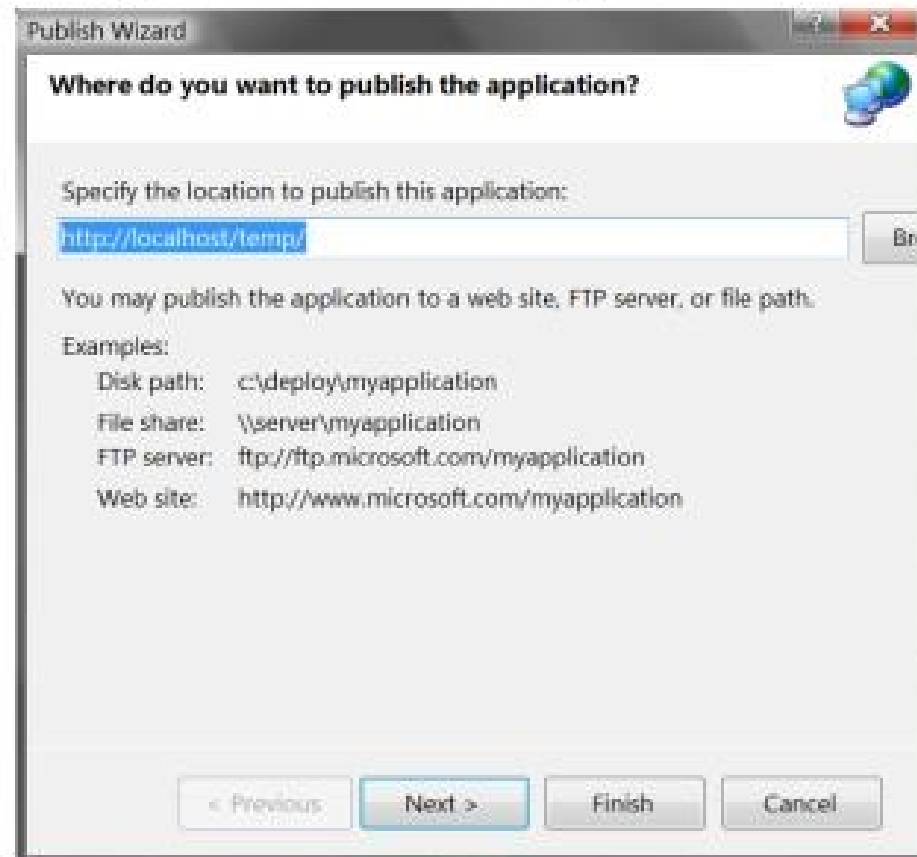
Развертывание приложений (2)

Преимущества технологии ClickOnce:

- унификация управления элементами и зависимостями (метаданными) приложения;
- контроль корректности установки приложения на компьютере клиента;
- автоматическое определение доступности обновлений в Интернет;
- автоматическое обновление версии приложения на компьютере клиента;
- при обновлении приложения разработчику достаточно опубликует новый манифест (manifest) и новую копию приложения на Web-сервере по указанному URL.

Корпоративные системы
Формы ввода и распределенные приложения

Развертывание приложений (3)



Пример настройки URL для размещения разработчиком новой версии приложения в Интернет для автоматического обновления посредством технологии ClickOnce

Windows Forms - другие особенности:

- поддержка диалоговых окон,
- печать;
- добавление справочной информации,
- добавление документации,
- локализация (перевод на другой язык) приложения;
- использование встроенной в .NET Framework системы информационной безопасности.

Благодарю за внимание!

Вопросы?

- <http://zykov.altweb.ru>
- szykov@hotmail.com
- sergey.zykov@tekama.com