

ЛАБОРАТОРНАЯ РАБОТА 16. СОЗДАНИЕ ПРИЛОЖЕНИЯ ПО ТЕХНОЛОГИИ WINDOWS FORMS

1. Цель и содержание

Цель лабораторной работы: научиться создавать приложение Windows в среде MS Visual Studio и программировать алгоритмы с использованием простых элементов управления.

Задачами лабораторной работы являются:

- разработка приложения с использованием Windows Forms,
- изучение основных членов класса Form;
- обработка событий от простых элементов управления.

2. Формируемые компетенции

Лабораторная работа направлена на формирование следующих компетенций:

- способность к проектированию базовых и прикладных информационных технологий (ПК-11);
- способность разрабатывать средства реализации информационных технологий (методические, информационные, математические, алгоритмические, технические и программные) (ПК-12).

3. Теоретическая часть

Файловая структура проекта MS Visual Studio в случае использования типа проекта Windows Form Application выглядит следующим образом (рис. 21.1):

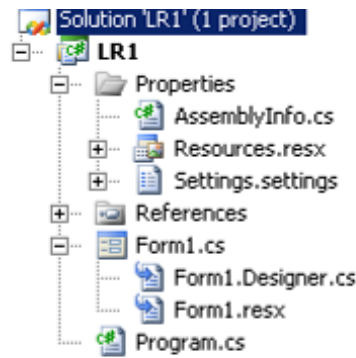


Рисунок 21.1 – Вид окна Solution Explorer для Windows Form Application.

Файл Program.cs содержит класс Program и статический метод Main(), с которого начинается выполнение приложения (рис. 21.2).

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Windows.Forms;

namespace LR1
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Form1());
        }
    }
}

```

Рисунок 21.2 – Листинг Program.cs.

В самом начале файла Program.cs выполняется объявление используемых пространств имен с использованием using.

Класс Program содержит метод Main(), который используя статический метод Run класса Application создает и выводит на экран главную форму приложения: Application.Run (new Form1()).

Таким образом, реализуется один из принципов объектно-ориентированного программирования: разграничение обязанностей (т.е. каждый класс выполняет минимально возможное количество операций).

Класс главной формы Form1 (по умолчанию) представлен двумя связанными C#-файлами. Для отображения содержимого Form1.cs необходимо щелкнуть правой кнопкой мыши в окне проектирования главной формы на самой форме или на пиктограмме Form1.cs в окне Solution Explorer (рис. 21.1) и в появившемся контекстном меню выбрать пункт «View Code».

Листинг Form1.cs представлен на рис 21.3.

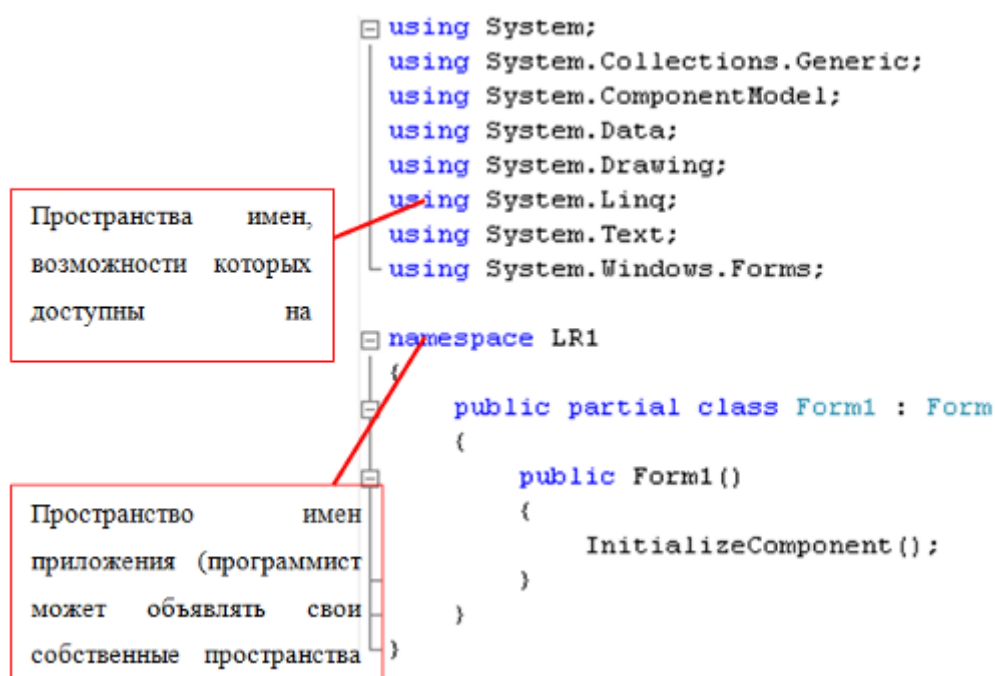


Рисунок 21.3 –Листинг Form1.cs.

Конструктор, созданный по умолчанию, вызывает метод InitializeComponent, который определен в соответствующем файле Form1.Designer.cs (рис. 21.1).

Этот метод создается автоматически, в нем отображаются все изменения, производимые программистом в окне визуального проектирования формы.

Пространства имен, доступные для использования, объявляются в начале файла.

System является базовым пространством имен – в него входят все остальные типы и пространство имен. Конструкция `using System;` в начале файла программы указывает на то, что весь программный код будет выполняться в данном пространстве имен, поэтому при использовании типов (например `Int32`), определенных в пространстве `System` нет необходимости указывать само имя пространства (то есть нет необходимости писать `System.Int32`).

Программист может самостоятельно вводить пространства имен при написании приложений.

Обработка событий в режиме проектирования. Окно проектирования главной формы приложения (рис. 21.4) можно открыть дважды щелкнув на пиктограмме `Form1.cs` в окне `Solution Explorer` (рис. 21.1). При этом откроется вкладка `Form1.cs [Design]`.

Окно свойств примет вид, показанный на рисунке 21.4.

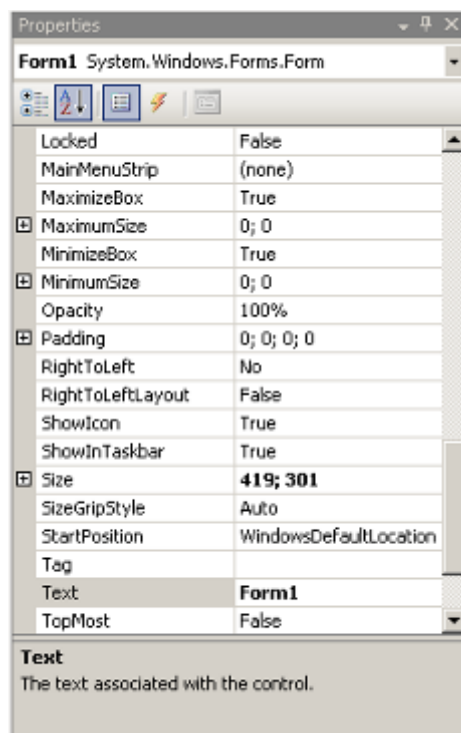



Рисунок 21.4 – Вид окна Свойств (Properties) в режиме проектирования формы.

Кнопка  позволяет отобразить список событий, характерных для элемента управления, активного в данный момент в окне проектирования. Например, если выделить главную форму приложения, то отобразятся события, характерные для класса Form.

Для отработки механизма регистрации событий, рассмотрим обработку наиболее простых событий – событий от мыши.

Для добавления обработчика события мыши необходимо выбрать соответствующее событие (Click, DoubleClick, MouseEnter, MouseLeave,MouseDown, MouseUp, MouseMove, MouseHover, MouseWheel) и дважды щелкнуть на поле, расположенном в соседнем от него столбце в окне свойств (таким образом обработчику присвоится имя по умолчанию). В данном поле можно ввести свое название для нового обработчика, или выбрать из выпадающего списка уже существующие обработчики.

После регистрации события в окне «Properties», автоматически добавляются строки кода в файлы Form1.Designer.cs (регистрируется обработчик события) и Form1.cs (добавляется метод, ассоциированный с данным событием).

Форма приложения должна быть функциональной и максимально эргономичной. Такие простые вещи, как цветовые комбинации, размеры шрифтов и окон могут сделать приложение намного более привлекательным для пользователя.

В случае если свойство Icon формы установлено, а для свойства ControlBox не указано значение false, то значок появится в левом верхнем углу формы. Обычно принято устанавливать здесь значок приложения.

Свойство FormBorderStyle задает тип рамки, которая появляется вокруг формы. Для этого используется перечисление FormBorderStyle. Допустимые значения этого перечисления: Fixed3D, FixedDialog, FixedSingle, FixedToolWindow, None, Sizable, SizableToolWindow.

Для выполнения лабораторной работы потребуется основная информация о событиях мыши:

Все обработчики событий мыши принимают параметр типа MouseEventArgs. Поступающий на вход обработчика событий мыши объект типа MouseEventArgs позволяет получить дополнительную информацию о действии мыши, путем введения ряда специальных членов класса (табл. 21.1).

Таблица 21.1 – Свойства типа MouseEventArgs

Свойство	Описание
Button	Содержит информацию о том, какая клавиша была нажата, в соответствии с определением перечня MouseButtons
Clicks	Содержит информацию о том, сколько раз была нажата и отпущена клавиша мыши
Delta	Содержит информацию со знаком, соответствующее числу щелчков, произошедших при вращении колесика мыши
X	Содержит информацию о координате X
Y	Содержит информацию о координате Y

4. Оборудование и материалы

Для выполнения лабораторной работы рекомендуется использовать персональный компьютер со следующими характеристиками: 64-разрядный (x64) процессор с тактовой частотой 1 ГГц и выше, оперативная память – 1 Гб и выше, свободное дисковое пространство – не менее 1 Гб, графическое устройство DirectX 9. Программное обеспечение: операционная система WINDOWS 7 и выше, Microsoft Visual Studio 20112 и выше.

5. Указания по технике безопасности

Техника безопасности при выполнении лабораторной работы определяется общепринятой для пользователей персональных компьютеров. Самостоятельно не производить ремонт персонального компьютера, установку и удаление программного обеспечения; в случае неисправности персонального

компьютера сообщить об этом обслуживающему персоналу лаборатории; не касаться электрических розеток металлическими предметами; рабочее место пользователя персонального компьютера должно содержаться в чистоте; не разрешается возле персонального компьютера принимать пищу, напитки.

6. Методика и порядок выполнения работы

1. Создайте в среде разработки MS VS проект. В качестве типа проекта выбрать «Windows Application» (или «Windows Forms Application» в зависимости от версии .NET Framework)

2. После создания и сохранения проекта измените программу таким образом, чтобы координаты курсора мыши выводились в заголовке главного окна приложения.

Для этого в обработчик события движения мыши над оконной формой (MouseMove) добавьте строку кода:

```
Text = string.Format("Координаты: {0}, {1}", e.X, e.Y);
```

3. Добавьте текстовое поле (TextBox) в режиме разработки (для этого необходимо использовать панель элементов управления «ToolBox»). Дополните обработчик движения мыши таким образом, чтобы в текстовом поле отображалась сумма координат указателя мыши.

4. Работающую программу необходимо представить преподавателю.

5. После этого выполните индивидуальное задание в соответствии с вариантом. В каждом задании необходимо вывести значение выражения, предварительно введя значения переменных в соответствующие текстовые поля формы главного окна приложения. Результат выводится в заголовок окна в ответ на нажатие кнопки (кнопку также необходимо поместить на форму).

6. Для выполнения индивидуального задания необходимо использовать математические функции, которые доступны в виде статических методов класса Math.

Индивидуальное задание.

Перед выполнением задания требуется самостоятельно определить закономерность изменения членов последовательности, чтобы применить цикл, условный оператор или, если потребуется, оператор выбора.

Вариант	Выражение для вычисления
1	$Z = \sqrt{\left \frac{a \cdot x}{w^a} \right } + \sqrt{ b } - x + \cos(y) $
2	$F = \sqrt{ d2 } \cdot x + \sqrt{ b^3 } - x^2 + \cos(y) $
3	$A = -d \cdot \frac{z}{\sqrt{ e }} + \sin(e) + \cos(y) $
4	$U = \sqrt{\left \frac{f - e}{w} \right } + \left \sin^2\left(\frac{e}{w}\right) + \cos(y) \right $
5	$t = \lg(f) - e + \left \sin\left(\frac{w}{t}\right) + \sqrt{ e } \right $
6	$V = \frac{h - e}{q + a} + \left \sin(f) + \sqrt{ \sin(b) } \right $
7	$Q = \sin\left(h + \frac{d}{e^{o1}}\right) - o1 + \left \sin(f) + \sqrt{ \sin(b) } \right $
8	$E = \sin\left(h11 + \frac{d12}{\ln(h11)}\right) - v6 + \left \sin(h3) + \sqrt{ \sin(f) } \right $
9	$Z = z \cdot ax1 + \sqrt{\left e7 + \frac{x}{y} \right } - x + \cos(as_9 + y) $
10	$RES = \frac{\arg 01 - \arg 02 + \left \sin(\arg 03) + \sqrt{ \arg 04 } \right }{\arg 01^{\arg 5} - \lg(\arg 02)}$
11	$U = \frac{v - e_2 + \left \cos(\arg 3) + \sqrt{ t } \right }{a_del * tg(t)}$
12	$t = W + \frac{\cos(g \cdot q)}{W} - e + \left \sin(e) + \sqrt{ o } \right $
13	$Z = z \cdot \cos(y) + \sqrt{e7} - x + \cos(as_9 + y) $
14	$R = q + \left \sin^2(e) + \cos(y) \right \cdot \cos(s + g)$
15	$F = \sin\left(\frac{v \cdot x}{y}\right) \cdot x + \sqrt{ b^3 } - x^2 + \cos(y) $

16	$res = \frac{F}{\cos(w)} - e + \frac{\left \sin\left(\frac{w}{t}\right) + \sqrt{ e } \right }{e}$
17	$t = W + \frac{tg(g \cdot q)}{\ln(20 \cdot x)} - \frac{e}{W} + \left \sin(e) + \sqrt{ W - e } \right $
18	$t = W + \frac{tg(g + q)}{\ln(20 \cdot x)} - \frac{e}{W} + \left \sin(e) + \sqrt{ W - e } \right $
19	$F = \sqrt{ d2 - x } \cdot x + \sqrt{ b^3 } - x^2 + \cos(y) $
20	$t = W + \frac{\cos(g \cdot q)}{W} - e + \left \sin(e) + \sqrt{\left \frac{o}{e^4} \right } \right $
21	$U = \frac{v^3 - e_2 + \left \cos(\arg 3) + \sqrt{ t } \right }{a_del * tg(t)}$
22	$F = \sin\left(\frac{v \cdot x}{y}\right) \cdot x + \sqrt{ b^3 } - x^2 + \cos(y) $
23	$v = \sqrt[4]{e_2} \cdot \sqrt{\frac{\sin(12 * \arg 3)}{e_2 - 1}} + \left \cos(\arg 3) + \sqrt{ t } \right $
24	$res = \frac{F^{ e }}{\cos(w + e)} - e + \frac{\left \sin\left(\frac{w}{t}\right) \right }{25 + \sqrt{ e }}$
25	$t = \frac{az}{\sqrt{g^3}} + \frac{\cos(g \cdot q)}{az} - g + \left \sin(q) + \sqrt{ 12 - q } \right $

7. Содержание отчета и его форма

Отчет по лабораторной работе должен содержать:

1. Номер и название лабораторной работы.
2. Цели лабораторной работы.
3. Ответы на контрольные вопросы.
4. Экранные формы и листинг программного кода, показывающие порядок выполнения лабораторной работы, и результаты, полученные в ходе её выполнения.

Отчет о выполнении лабораторной работы в письменном виде сдается преподавателю.

8. Контрольные вопросы

1. Какие файлы описывают класс формы?
2. Какие действия необходимо выполнить для создания обработчика события?
3. Где описывается код обработчика события? В каком файле регистрируется обработчик события (метод привязывается к событию)?
4. Как получить доступ к координатам курсора мыши?
5. Какой класс содержит методы, реализующие математические функции?

9. Список литературы

Для выполнения лабораторной работы, при подготовке к защите, а также для ответа на контрольные вопросы рекомендуется использовать следующие источники: [6].

ЛАБОРАТОРНАЯ РАБОТА 17. ПРИМЕНЕНИЕ ЭЛЕМЕНТОВ УПРАВЛЕНИЯ В ПРИЛОЖЕНИЯХ WINDOWS

1. Цель и содержание

Цель лабораторной работы: изучить принципы использования стандартных элементов управления в приложениях Windows.

Задачи лабораторной работы:

- научиться использовать кнопки, флажки, переключатели;
- научиться использовать списки, выпадающие списки;
- научиться .

2. Формируемые компетенции

Лабораторная работа направлена на формирование следующих компетенций:

- способность к проектированию базовых и прикладных информационных технологий (ПК-11);
- способность разрабатывать средства реализации информационных технологий (методические, информационные, математические, алгоритмические, технические и программные) (ПК-12).

3. Теоретическая часть

Понимание иерархии классов элементов управления Windows имеет важное значение при проектировании интерфейсов пользователя, особенно при разработке собственных элементов управления.

В пространстве имен `System.Windows.Forms` есть один особенный класс, служащий базовым для почти каждого создаваемого элемента управления и формы. Это `System.Windows.Forms.Control`. Класс `Control` реализует базовую