

Практическая работа 28

Разработка пользовательского интерфейса в WindowsForms

Цель занятия: Получить практический опыт разработки пользовательского Windows-интерфейса с использованием WindowsForm

Перечень оборудования и программного обеспечения

Персональный компьютер
Microsoft Office (Word, Visio)
Microsoft Visual Studio

Краткие теоретические сведения

Пространство имен System.Windows.Forms содержит классы для создания приложений Windows, которые позволяют наиболее эффективно использовать расширенные возможности пользовательского интерфейса, доступные в операционной системе MicrosoftWindows.

Категория класса	Подробно
Элементы управления, пользовательские элементы управления и формы	Большинство классов в пространстве имен System.Windows.Forms являются производными от класса Control. Класс Control предоставляет основные функциональные возможности для всех элементов управления, отображаемых в Form. Класс Form представляет окно в приложении. Оно включает диалоговые окна, немодальные окна, а также клиентские и родительские окна интерфейса MDI. Можно также создать собственные элементы управления путем наследования от класса UserControl.
Меню и панели инструментов	WindowsForms включает широкий набор классов, которые позволяют создавать пользовательские панели инструментов и меню, отличающиеся современным обликом и поведением. ToolStrip, MenuStrip, ContextMenuStrip и StatusStrip позволяют создавать панели инструментов, строки меню, контекстные меню и строки состояния, соответственно.
Элементы управления	Пространство имен System.Windows.Forms предоставляет большое количество классов элементов управления, которые позволяют создавать пользовательские интерфейсы с расширенными возможностями. Некоторые элементы управления

	<p>предназначены для ввода данных в приложении, например, элементы TextBox и ComboBox. Другие элементы управления отображают данные приложений, например, Label и ListView. Это пространство имен также предоставляет элементы управления для вызова команд в приложении, например, Button. Элемент управления WebBrowser и такие классы управляемых HTML-страниц, как HtmlDocument, позволяют отображать HTML-страницы и выполнять с ними определенные действия в области управляемого приложения WindowsForms. Элемент управления MaskedTextBox представляет собой улучшенный элемент управления вводом данных, который позволяет создавать маску для принятия или отклонения введенных пользователем данных в автоматическом режиме.</p> <p>Кроме того, с помощью элемента управления PropertyGrid можно создать собственный конструктор WindowsForms, отображающий видимые конструктором свойства элементов управления.</p>
Макет	<p>Несколько важных классов в WindowsForms помогают контролировать расположение элементов управления на отображаемой поверхности, например, в форме или элементе управления. FlowLayoutPanel располагает все элементы управления, которые содержит в последовательном режиме, а TableLayoutPanel позволяет определять ячейки и строки для расположения элементов управления в фиксированной сетке. SplitContainer разделяет поверхность отображения на две или более корректируемых части.</p>
Данные и привязка данных	<p>WindowsForms обеспечивает расширенную архитектуру для привязывания к таким источникам данных, как базы данных и XML-файлы. Элемент управления DataGridView предоставляет настраиваемую таблицу для отображения данных и позволяет настраивать формат ячеек, строк, столбцов и границ.</p> <p>Элемент управления BindingNavigator представляет стандартный способ навигации и работы с данными в форме; BindingNavigator часто используется в сочетании с элементом управления BindingSource для перемещения от одной записи к другой в форме,</p>

	а также для выполнения операций с записями.
Компоненты	Помимо элементов управления пространство имен System.Windows.Forms предоставляет другие классы, которые не являются производными от класса Control, но также обеспечивают визуальные функции для приложений Windows. Такие классы, как ToolTip и ErrorProvider, расширяют возможности или предоставляют сведения пользователям. Классы Help и HelpProvider позволяют отображать текст справки для пользователя, который работает с приложениями.
Общие диалоговые окна	Windows предоставляет несколько основных диалоговых окон, позволяющих обеспечить единообразие пользовательского интерфейса в приложениях Windows при выполнении таких операций как открытие и сохранение файлов, задание цвета шрифта или текста и печать. Классы OpenFileDialog и SaveFileDialog предоставляют возможность отображения диалогового окна, в котором пользователь может выполнить поиск файла, а также ввести имя файла, который необходимо открыть или сохранить. Класс FontDialog отображает диалоговое окно для изменения элементов Font, используемого приложением. Классы PageSetupDialog, PrintPreviewDialog и PrintDialog отображают диалоговые окна, позволяющие пользователю управлять параметрами

Элемент управления MaskedTextBox использует маску, чтобы отличить допустимые данные, вводимые пользователем, от недопустимых.

Свойство Mask используется по умолчанию для класса MaskedTextBox.

Маска **Mask** должна быть строкой, состоящей из одного или нескольких элементов маски, как показано в следующей таблице. Язык маски, используемый в элементе управления MaskedTextBox, определяется сопоставленным с ним объектом MaskedTextBoxProvider.

Maskingelement	Описание
0	Цифра, необходимый символ. Данный элемент позволяет вводить любое число от 0 до 9
9	Цифра или пробел, необязательный символ.

#	<p>Цифра или пробел, необязательный символ.</p> <p>Если эта позиция в маске пуста, в свойстве Text она будет представлена пробелом. Допускаются знаки плюса (+) и минуса (-).</p>
L	<p>Буква, необходимый символ. Ограничивает вводимые данные наборами букв ASCII от a до z и от A до Z. Этот элемент маски аналогичен регулярному выражению [a-zA-Z].</p>
?	<p>Буква, необязательный символ. Ограничивает вводимые данные наборами букв ASCII от a до z и от A до Z. Этот элемент маски эквивалентен выражению [a-zA-Z]? inregular expressions.</p>
&	<p>Символ, необходимый.</p> <p>Если свойство AsciiOnly имеет значение true, этот элемент действует аналогично элементу "L".</p>
C	<p>Символ, необязательный. Любой неуправляющий символ. Если свойство AsciiOnly имеет значение true, этот элемент действует аналогично элементу "?".</p>
a[x]	<p>Буква или цифра, необходимый символ.</p> <p>Если свойство AsciiOnly имеет значение true, будут приниматься только буквы ASCII от a до z и от A до Z.</p> <p>Этот элемент маски действует аналогично элементу "A".</p>
a[x]	<p>Буква или цифра, необязательный символ.</p> <p>Если свойство AsciiOnly имеет значение true, будут приниматься только буквы ASCII от a до z и от A до Z.</p> <p>Этот элемент маски действует аналогично элементу "A".</p>
.	<p>Местозаполнитель для десятичного разделителя.</p> <p>Символ десятичного разделителя, который будет фактически отображаться, соответствует поставщику форматирования, определяемому свойством FormatProvider элемента управления.</p>
,	<p>Местозаполнитель для разделителя групп разрядов.</p> <p>Символ разделителя групп разрядов, который будет фактически отображаться, соответствует поставщику</p>

	форматирования, определяемому свойством <code>FormatProvider</code> элемента управления.
:	<code>Timeseparator</code> . Символ разделителя компонентов времени, который будет фактически отображаться, соответствует поставщику форматирования, определяемому свойством <code>FormatProvider</code> элемента управления.
/	<code>Dateseparator</code> . Символ разделителя компонентов даты, который будет фактически отображаться, соответствует поставщику форматирования, определяемому свойством <code>FormatProvider</code> элемента управления.
\$	Символ денежной единицы. Символ денежной единицы, который будет фактически отображаться, соответствует поставщику форматирования, определяемому свойством <code>FormatProvider</code> элемента управления.
<	Переключение в нижний регистр. Преобразует все последующие символы в нижний регистр.
>	Переключение в верхний регистр. Преобразует все последующие символы в верхний регистр.
	Отмена предыдущего изменения регистра.
\	Escape-знак. Заменяет символ маски на литерал. " <code>\\</code> " — escape-последовательность для обратной косой черты.
<code>AllOtherCharacters</code>	<code>Literals</code> . Все элементы, не образующие маску, отображаются в элементе <code>MaskedTextBox</code> без изменений. Литералы всегда занимают статическую позицию в маске во время выполнения, и пользователь не может перемещать или удалять их.

Если изменить маску в момент, когда элемент управления `MaskedTextBox` содержит введённые данные, отфильтрованные предыдущей маской, элемент управления `MaskedTextBox` выполнит попытку переноса этих данных в новое определение маски.

В случае неудачи существующие данные будут сброшены. Если задать в качестве маски строку нулевой длины, будут сохранены все существующие данные в элементе управления. Элемент управления `MaskedTextBox` с маской нулевой длины работает аналогично однострочному элементу управления `TextBox`.

Десятичный разделитель (.), разделитель групп разрядов (,), разделитель компонентов времени (:), разделитель компонентов даты (/) и символы денежной единицы (\$) по умолчанию отображаются в соответствии с языком и региональными параметрами приложения. Можно принудительно отображать символы для других региональных параметров, используя свойство `FormatProvider`.

Вставка символов в маску во время выполнения управляется свойством `InsertKeyMode`. Пользователи могут перемещаться по маске, используя клавиши со стрелками влево и вправо или курсор мыши, а ввод пробела позволяет пропустить необязательные позиции в маске.

В следующей таблице приведены примеры масок.

Mask	Назначение
00/00/0000	Дата (день, месяц в числовом формате, год) в международном формате. Знак "/" в маске является логическим разделителем компонентов даты. Пользователю будет отображаться разделитель компонентов даты, соответствующий текущему языку и региональным параметрам приложения.
00->L<LL-0000	Дата (день, сокращенное наименование месяца и год) в формате США, в котором трехбуквенное сокращенное наименование месяца отображается с первой буквой в верхнем регистре и двумя следующими — в нижнем.
(999)-000-0000	Номер телефона в США и код города (необязательно). Чтобы не вводить необязательные символы, пользователь может вводить вместо них пробелы или переместить указатель мыши в позицию маски, представленную первой цифрой 0.
\$999,999.00	Значение денежных единиц лежит в диапазоне от 0 до 999999. Символы денежной единицы, разделителя групп разрядов и десятичного разделителя будут заменены во время выполнения на эквиваленты, соответствующие текущему языку и региональным параметрам.

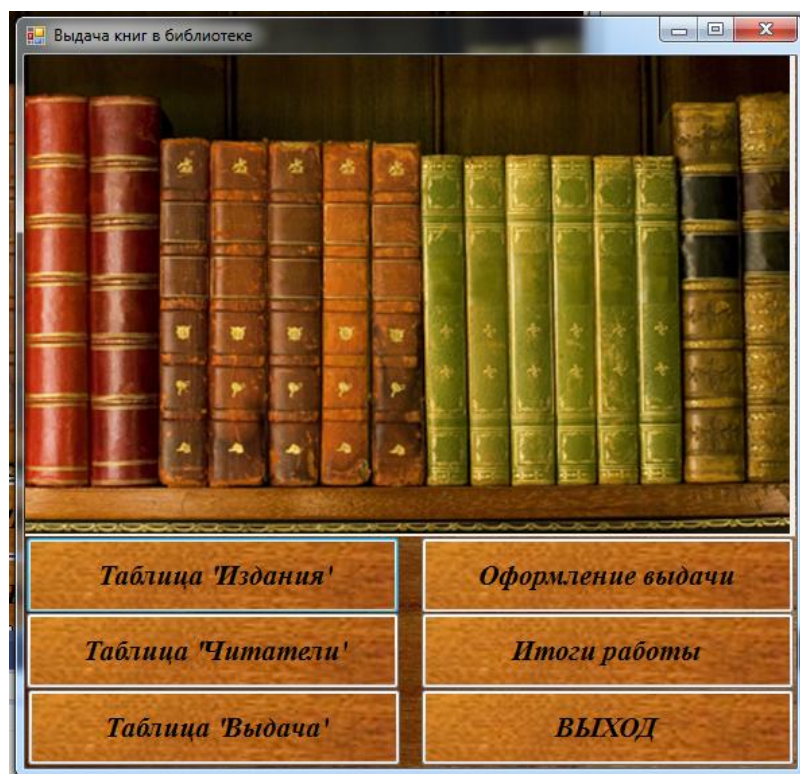
Задания

- 1 Изучить теоретические сведения и задание к работе
- 2 В соответствии с вариантом задания разработать интерфейс основных программных модулей с использованием `WindowsForms`.

- На главной странице расположить меню, в котором осуществить вызов модулей. Для полей ввода использовать MaskedTextBox с использованием маски ввода.
- Для подсказки ввода использовать элементы управления StatusStrip, ToolTip
- На формах внести изменения во внешний вид используемых элементов управления.

Порядок выполнения работы

В Visual Studio создадим проект, C# и приложение Windows Forms. Начнем с главной кнопочной формы. На нее поместим надписи, изображения и кнопки. Одна из кнопок должна обеспечивать окончание работы с информацией. Оформим с помощью свойств созданные объекты.



Добавим формы в проект в соответствии с темой (нажмем на имя проекта в окне Solution Explorer (Обозреватель решений) правой кнопкой мыши и выберем Add(Добавить)->Windows Form...). Организуем взаимодействие между двумя формами, добавив код вызова форм:

```
owsformsapp15 windowsformsapp15.form1
ссылка:1
public Form1()
{
    InitializeComponent();
}

ссылка:1
private void button6_Click(object sender, EventArgs e)
{
    this.Close();
}

ссылка:1
private void button1_Click(object sender, EventArgs e)
{
    Form nF2 = new Form2();
    nF2.Show();
    this.Hide();
}

ссылка:1
private void button2_Click(object sender, EventArgs e)
{
    Form nF3 = new Form3();
    nF3.Show();
    this.Hide();
}

ссылка:1
private void button3_Click(object sender, EventArgs e)
{
    Form nF4 = new Form4();
    nF4.Show();
    this.Hide();
}
```

Тогда после вызова новой формы текущая будет скрыта, а после завершения работы в новой форме, можно снова открыть главную.

Если нужно из второй кнопочной формы, например, вызвать третью, то соответствующей кнопке задаем код:

```
Form nF4 = new Form4();
nF4.Show();
this.Hide();
```

А для возврата код:

```
this.Close();
Form nF2 = Application.OpenForms[1];
nF2.Show();
```

...

Содержание отчета

- 1 Название работы
- 2 Цель работы
- 3 Технические средства обучения
- 4 Задания (условия задач)
- 5 Порядок выполнения работы
- 6 Вывод

Варианты заданий

1 Автотранспортное предприятие. Основные объекты предметной области: автомобили, марки автомобилей, водители, заказчики, грузы. Основные функции: учет заказов по перевозке грузов и подбор автомобилей. Выходные документы: накладная на перевозку груза для водителя.

2 Агентство недвижимости. Основные объекты предметной области: квартиры, продавцы, покупатели, агенты, сделки. Основные функции: учет заказов и подбор вариантов в соответствии с требованиями. Выходные документы: договор на осуществление сделок.

3 Банк. Основные объекты предметной области: вкладчики, типы счетов, счета, поступления на счёт, снятия со счета. Основные функции: учет вкладов и начисление процентов по вкладам. Выходные документы: договор на открытие вклада.

4 Поликлиника. Основные объекты предметной области: пациенты, посещения, врачи, специальности, диагнозы. Основные функции: учет посещений и ведение медицинских карт. Выходные документы: журнал посещений врача.

5 Начисление зарплаты. Основные объекты предметной области: сотрудники, ставки, оклады, квалификации сотрудников. Основные функции: начисление зарплаты. Выходные документы: ведомость начисления зарплаты.

6 Юридическое агентство. Основные объекты предметной области: клиенты, услуги, номенклатура услуг. Основные функции: учет предоставленных услуг. Выходные документы: ведомость оказания услуг.

7 Отдел продаж. Основные объекты предметной области: товары, продажи, продавцы, покупатели. Основные функции: учет продаж. Выходные документы: товарная накладная.

8 Отель. Основные объекты предметной области: номерной фонд, типы номеров, занятость номеров, клиенты, заказы, сотрудники. Основные функции: учет занятости номеров. Выходные документы: журнал учета проживающих.

9 Производство металлопластиковых окон. Основные объекты предметной области: заказы, готовые изделия, материалы, спецификации, клиенты. Учет расхода материалов. Выходные документы: накладная на производство заказа.

10 Фитнес-клуб. Основные объекты предметной области: клиенты, тренеры, абонементы. Основные функции: учет продажи абонементов. Выходные документы: журнал продажи абонементов.

11 Заказ такси. Основные объекты предметной области: клиенты, поездки, автомобили, водители. Основные функции: Учет заказов. Выходные документы: журнал заказов.

12 Турфирма. Основные объекты предметной области: клиенты, продажи, страны, туры. Основные функции: подбор туров. Выходные документы: журнал продаж.

13 Центр детского творчества. Объекты предметной области: учащиеся, кружки, занятия. Основные функции: учет учащихся центра. Выходные документы: журнал учащихся.

14 Аптека. Объекты предметной области: лекарства, показания, поставщики, пациенты. Основные функции: учет заказов лекарств. Выходные документы: журнал заказов

15 Центр занятости. Объекты предметной области: вакансии, резюме, предприятия, организации, повышение квалификации. Выходные документы: резюме.

16 Магазин бытовой техники. Объекты предметной области: категории товаров, условия кредитов, условия доставки. Выходные документы: прайс-лист.

17 Ателье пошива одежды. Объекты предметной области: портные, услуги, материалы. Выходные документы: прайс-лист.

18 Театр. Объекты предметной области: актеры, спектакли. Выходные документы: афиша.

19 Зоопарк. Объекты предметной области: животные, мероприятия. Выходные документы: расписание мероприятий.

20 Станция техобслуживания. Объекты предметной области: оказываемые услуги, обслуживающий персонал. Выходные документы: прайс-лист.

21 Кредитный отдел банка. Основные объекты предметной области: клиенты, типы кредитов, счета, график платежей. Основные функции: учет выплат и начисление процентов по вкладам. Выходные документы: договор на открытие кредита.

22 Пункт обмена валют. Объекты предметной области: курс валют, клиенты, заказ валют, наличие. Выходные документы: журнал заказа.

23 Отдел кадров. Основные объекты предметной области: должности, отделы, сотрудники, квалификации сотрудников. Основные функции: начисление зарплаты. Выходные документы: ведомость сотрудников.

Используемая литература

1 Гниденко, И. Г. Технология разработки программного обеспечения: учеб. пособие для СПО / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — М.: Издательство Юрайт, 2017.

2 Шарп Джон Ш26 Microsoft Visual C#. Подробное руководство. 8-е изд. — СПб.: Питер, 2017.

3 Васильев А.Н. Программирование на C# для начинающих. Основные сведения. — Москва: Эксмо, 2018.

4 Васильев А.Н. Программирование на C# для начинающих. Особенности языка. — Москва: Эксмо, 2019.

5 <http://msdn.microsoft.com/ru-ru/library/67ef8sbd.aspx>.