

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ**

**РОССИЙСКОЙ ФЕДЕРАЦИИ**

Брянский государственный технический университет

**Утверждаю**

**Ректор университета**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_О.Н.Федонин**

**«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2017 г.**

**Программирование в среде Microsoft .NET**

**Организация ввода/вывода с помощью элементов управления Windows FoRms**

**Методические указания**

**к выполнению лабораторной работы №3**

**для студентов очной формы обучения по направлениям подготовки 02.03.03 – «Математическое обеспечение и администрирование информационных систем», 09.03.01 – «Информатика и вычислительная техника» и 09.03.04 – «Программная инженерия»**

**Брянск 2017**

УКД 004.43

Программирование в среде Microsoft .Net. Организация ввода/вывода с помощью элементов управления Windows Froms [Текст] + [Электронный ресурс]: Методические указания к выполнению лабораторной работы №3 для студентов очной формы обучения по направлениям подготовки 02.03.03 – «Математическое обеспечение и администрирование информационных систем», 09.03.01 – «Информатика и вычислительная техника» и 09.03.04 – «Программная инженерия». – Брянск: БГТУ, 2017. – 8 с.

Разработали:

Д.А.Коростелёв

канд. техн. наук, доц.

Рекомендовано кафедрой «Информатика и программное обеспечение» БГТУ (протокол №2 от 16.09.2016)

Научный редактор А.А.Азарченков

Редактор издательства Л.И.Афонина

Компьютерный набор Д.А.Коростелёв

Темплан 2017 г., п.273

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подписано в печать 23.09.17. Формат 60х84 1/16 Бумага офсетная. Офсетная печать. Усл.печ.л. 1,8. Уч.-изд.л. 1,8 Тираж 1 экз. Заказ Бесплатно.

Издательство Брянского государственного технического университета

241035, Брянск, бульвар 50-летия Октября, 7, БГТУ. 58-82-49.

Лаборатория оперативной полиграфии БГТУ, ул. Институтская, 16.

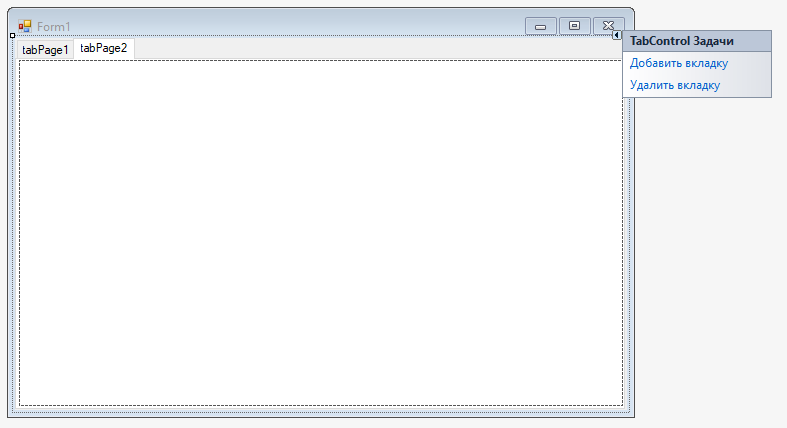
**Цель работы**

Целью работы является изучение основ ввода/вывода информации для элементов управления Windows Forms.

Продолжительность работы – 2 часа.

**Элемент управления TabControl**

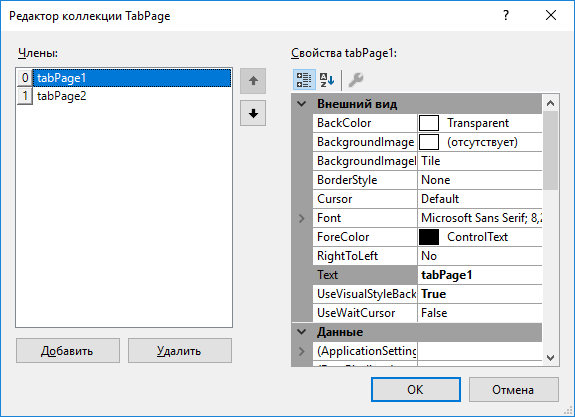
Элемент управления TabControl (рис. 1) позволяет управлять связанным набором страниц вкладок. Этот элемент управления может быть очень полезен, если необходимо разместить на форме достаточно большое количество других элементов управления.



***Рис. 1. Интерфейс элемента управления TabControl***

TabControl можно добавить на форму в режиме конструктора из панели элементов. Для добавления или удаления вкладок в TabControl можно воспользоваться смарт-тегом или свойством TabPages (рис. 2).

Рекомендуется для этого элемента управления задавать свойство Dock равным Fill. Выбор активной вкладки в режиме конструктора осуществляется одиночным щелчком на её названии.



***Рис. 2. Редактор коллекции TabPage***

**Элемент управления NumericUpDown**

Элемент управления Windows Forms NumericUpDown представляет собой сочетание текстового поля и пары кнопок со стрелками для выбора значения пользователем (рис. 3). Он выводит и задает отдельное числовое значение в списке вариантов. Пользователь может увеличивать и уменьшать число, нажимая кнопки со стрелками вверх и вниз или клавиши со стрелками ВВЕРХ и ВНИЗ, а также вводя число в поле. При нажатии клавиши со стрелкой ВВЕРХ значение увеличивается до максимума; при нажатии клавиши со стрелкой ВНИЗ число уменьшается до минимума. Числовые элементы управления с возможностью перемещения вверх и вниз по шкале используются в некоторых приложениях панели управления Windows.



***Рис. 3. Интерфейс элемента управления NumericUpDown***

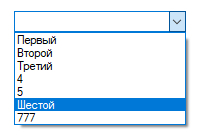
Минимальное и максимальное значение элемента управления NumericUpDown устанавливается с помощью свойств Minimum и Maximum. Текущее значение этого элемента может быть задано или считано через свойство Value и имеет тип данных decimal:

decimal number = numericUpDown1.Value

С помощью свойства DecimalPlaces можно задать количество отображаемых знаков после запятой.

**Элемент управления ComboBox**

Элемент управления Windows Forms ComboBox используется для вывода данных в раскрывающемся поле со списком (рис. 4). По умолчанию элемент управления ComboBox отображается в виде двух частей: верхняя часть представляет собой текстовое поле, в которое пользователь может ввести элемент списка. Вторая часть представляет собой список элементов, один из которых пользователь может выбрать.



***Рис. 4. Элемент управления ComboBox***

Свойство SelectedIndex возвращает целочисленное значение, соответствующее выбранному элементу списка. Выбранный элемент можно изменить программными средствами, изменив в коде значение SelectedIndex; соответствующий элемент списка появится в текстовом поле. Если выбранных элементов нет, значение SelectedIndex равно -1. Если в списке выбран первый элемент, значение SelectedIndex равно 0. Свойство SelectedItem аналогично свойству SelectedIndex, но возвращает сам элемент, обычно в виде строкового значения. Свойство Count отражает число элементов в списке, а значение свойства Count всегда на единицу больше максимально возможного значения свойства SelectedIndex, поскольку для свойства SelectedIndex индексация ведется от нуля.

Чтобы добавить или удалить элементы в элементе управления ComboBox, используйте метод Add, Insert, Clear или Remove. Кроме того, можно добавить элементы в список с помощью свойства Items во время разработки.

Для того чтобы в текстовое поле элемента управления нельзя было вписывать текст, свойству DropDownStyle необходимо задать значение DropDownList.

**Элемент управления TextBox**

Текстовые поля Windows Forms используются для приема данных, вводимых пользователем, или для отображения текста. Элемент управления TextBox обычно используется для редактируемого текста, хотя его можно также сделать доступным только для чтения. В текстовых полях можно выводить несколько строк текста (для этого необходимо задать свойству Multiline значение true), размещать текст в соответствии с размером элемента управления и применять основные элементы форматирования. В элементе управления TextBox можно вводить или отображать текст только в одном формате.

Текст, отображаемый в элементе управления, содержится в свойстве Text. По умолчанию в текстовом поле можно ввести до 2048 знаков. Если свойству Multiline присвоить значение true, это позволит вводить до 32 килобайт текста. Свойство Text может быть установлено в окне "Свойства" во время разработки, программными средствами во время выполнения или в результате ввода данных пользователем во время выполнения. Текущее содержимое текстового поля может быть получено во время выполнения путем считывания значения свойства Text.

Для доступа к строкам многострочного TextBox можно использовать свойство Lines (массив строк) или использовать переносы строки «\r\n» при заполнении свойства Text.

Свойство Text хранит значение типа string. Если есть необходимость считать значение как число, то можно использовать методы синтаксического разбора строк:

double d;

if (double.TryParse(textBox1.Text, out d))

{

//Переменная d хранит число, если преобразование удалось

}

или конвертирования:

d = Convert.ToDouble(textBox1.Text);

Для того чтобы в многострочном текстовом поле была возможность прокрутки, необходимо задать свойству ScrollBars значение Both.

**Задание на лабораторную работу**

1. Создайте Windows Forms приложения.
2. На основной форме разместите контейнер управления закладками (TabControl).
3. Добавьте дополнительные закладки в контейнер, чтобы их было 7. Дайте названия вкладкам по порядку: «Первая», «Вторая», «Третья», «Четвертая», «Пятая», «Шестая» и «Седьмая».
4. **Первая вкладка.** Разместите элемент управления NumericUpDown и настройте диапазон допустимых значений от 1 до 100.
5. Разместите на этой вкладке TextBox.
6. Добавьте обработчик на событие изменения значения в NumericUpDown. В этом обработчике выполните очистку TextBox и заполнение его числами от 1 до заданного в NumericUpDown в одну строку через пробел.
7. **Вторая вкладка.** Добавьте на вкладку ComboBox, два TextBox и два Button. Для ComboBox свойству DropDownStyle установите значение DropDownList.
8. Добавьте обработчик события Button №1 – при нажатии на него в список значений ComboBox должен добавляться текст из TextBox №1.
9. Добавьте обработчик события Button №2 – при нажатии на него список значений ComboBox должен очищаться.
10. Добавьте обработчик события ComboBox – при выборе значения в ComboBox оно должно также подставляться в TextBox №2.
11. **Третья вкладка.** Добавьте на вкладку четыре Button и три TextBox.
12. При нажатии на Button №1 (задайте ему текст «+») необходимо сложить вещественные значения из TextBox №1 и TextBox №2, а результат вывести в TextBox №3.
13. При нажатии на Button №2 (задайте ему текст «-») необходимо посчитать разницу вещественных значений из TextBox №1 и TextBox №2, а результат вывести в TextBox №3.
14. При нажатии на Button №3 (задайте ему текст «\*») необходимо перемножить вещественные значения из TextBox №1 и TextBox №2, а результат вывести в TextBox №3.
15. При нажатии на Button №4 (задайте ему текст «/») необходимо поделить вещественное значение из TextBox №1 на вещественное значение из TextBox №2, а результат вывести в TextBox №3.
16. **Четвертая вкладка**. Добавьте на вкладку TextBox, ComboBox и Button.
17. Задайте свойству Multiline для TextBox значение true, а свойству ScrollBars – Both.
18. Добавьте обработчик для Button – при нажатии на кнопку должна происходить построчная обработка TextBox. Для каждой строки должно проверяться – является ли она числом (вещественным). Если является – то добавить значение этого числа в список значений ComboBox. В начала обработки события Button необходимо очищать список значений ComboBox.
19. **Пятая вкладка.** Добавьте на вкладку TextBox, NumericUpDown и Button.
20. Задайте свойству Multiline для TextBox значение true, а свойству ScrollBars – Both.
21. Задайте ограничение диапазона возможных значений для NumericUpDown от 0.000001 до 0.01.
22. При нажатии на Button необходимо рассчитать сумму сходящегося ряда . Расчеты вести с точностью, задаваемой в NumericUpDown. Каждое промежуточное значение суммы выводить отдельной строкой в TextBox.
23. **Шестая** **вкладка.** Добавьте на вкладку два TextBox и Button.
24. Задайте свойству Multiline для обоих TextBox значение true, а свойству ScrollBars – Both.
25. Добавьте обработку события для Button – при нажатии на кнопку необходимо обработать все строки в TextBox №1. Если строка не представляет собой вещественное число, то она попадает в TextBox №2. Строки в TextBox №2 должны попадать в обратном порядке. Например, в TextBox №1 мы имеем следующий набор строк:

Test

Number

123

44FtFt

66.444

%gg5f

Тогда в TextBox №2 после нажатия на кнопку должно содержаться:

%gg5f

44FtFt

Number

Test

1. **Седьмая вкладка.** Добавьте на вкладку четыре TextBox и Button.
2. Необходимо протабулировать функцию *f*(*x*) = sin(*x*)/(|*x*|+1) на итервале [*a*, *b*] с шагом *h*.
3. Значение *a* берется из TextBox №1. Значение *b* берется из TextBox №2. Значение *h* берется из TextBox №3.
4. Результаты табуляции выводятся построчно в TextBox №4 после нажатия на кнопку Button в виде «x=…; f(x)=…».

**Контрольные вопросы**

1. Какие существуют стандартные элементы управления для Windows Forms?
2. Каким образом используется элемент управления вкладками?
3. Какие виды элементов управления относятся к контейнерам?
4. Какие основные свойства имеются у элемента управления TextBox?
5. Какие основные свойства имеются у элемента управления ComboBox?
6. Какие основные свойства имеются у элемента управления NumericUpDown?

# Список рекомендуемой литературы

1. Нейгел, К. C# 5.0 и платформа .NET 4.5 для профессионалов / К.Нейгел, Б.Ивьен, Дж.Глинн, М.Скиннер, К.Уотсон. – М.: Диалектика, 2013. – 1440 с.
2. Неш, Т. C# 2010: ускоренный курс для профессионалов / Т.Неш – М.: Вильямс, 2010. – 592 с.
3. Снелл, М. Microsoft Visual Studio 2008 / М.Снелл, Л.Пауэрс. – СПб: БХВ-Петербург, 2009. – 1200 с.
4. Троелсен, Э. Язык программирования С# 5.0 и платформа .NET 4.5 / Э.Троелсен. – 6-е изд. – М.: Вильямс, 2015. – 1312 с.
5. Рихтер, Дж. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C# / Дж.Рихтер. – СПб.: Питер, 2016. – 896 с.