

ЛАБОРАТОРНАЯ РАБОТА №10

Компонент ProgressBar

Предмет исследований

- Использование таймера и его режимов.
- Свойства компонента ProgressBar.
- Компонент statusStrip и его варианты вывода им информации.
- Работа с подкомпонентом StatusLabel компонента statusStrip.

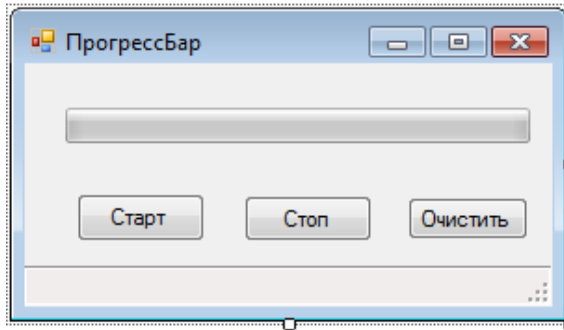
Контрольные вопросы

1. Варианты использования компонента ProgressBar в программах.
2. Задание параметров ProgressBar.
3. Работа с таймером.
4. Варианты использования Таймера в программах.
5. Компонент statusStrip и его подкомпоненты.
6. Задание параметров подкомпонентам statusStrip.
7. События onClick.

Задание. Создать программу, в которой будут использоваться следующие компоненты: ProgressBar, statusStrip (и один из его подкомпонентов). Для работы с ProgressBar, statusStrip используйте таймер.

Пример. Программа, демонстрирующая работу с компонентами ProgressBar, statusStrip (и один из его подкомпонентов). Их мы добавляем в форму из "Панели элементов", вкладка "Стандартные элементы управления". В программе также использованы компоненты Button для управления, они в "Панели элементов".

При нажатии кнопки "Старт" полоса прогресса в компоненте ProgressBar начинает заполняться слева направо, внизу в компоненте statusStrip показывается процент выполнения. При помощи кнопки "Стоп" полосу можно остановить. Кнопка "Очистка" используется для перевода ProgressBar и statusStrip в начальное состояние.



Листинг программы

```
using System;
using System.Windows.Forms;

namespace LabProgress
{
    public partial class Form1 : Form
    {
        Timer time = new Timer();
        public Form1()
        {
            InitializeComponent();
        }
        private void button1_Click(object sender, EventArgs e)
        {
            time.Interval = 250;
            time.Tick += new EventHandler(IncreaseProgressBar);
            time.Start();
        }
        private void IncreaseProgressBar(object sender, EventArgs e)
        {
            progressBar1.Increment(1);
            toolStripStatusLabel1.Text = progressBar1.Value.ToString() +
                "% Выполнено";
            if (progressBar1.Value == progressBar1.Maximum)
            {
                time.Stop();
                progressBar1.Value = 0;
            }
        }
    }
}
```

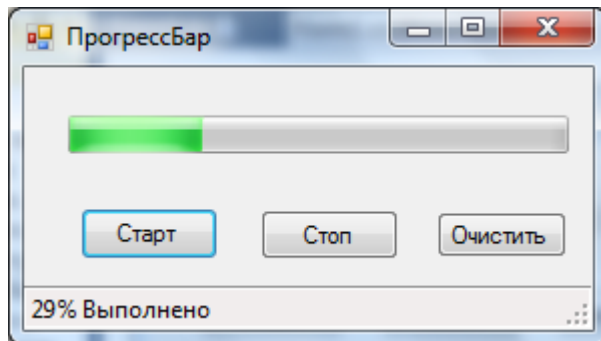
```

    }
    private void button2_Click(object sender, EventArgs e)
    {
        time.Stop();
    }

    private void button3_Click(object sender, EventArgs e)
    {
        progressBar1.Value = 0;
        toolStripStatusLabel1.Text = "0";
    }
}

```

При запуске нажимаем кнопку "Старт" и смотрим результат выполнения.



Списки

Предмет исследований

Приложение Windows с компонентами списков:

- Список listBox.
- Комбинированный список comboBox. Это комбинация простого списка с однострочным редактором textBox.

Контрольные вопросы

1. Списки, их назначение.
2. Компоненты списков listBox (список) и ComboBox (комбинированный спи-сок) в ИСП. Их назначение и сравнение.
3. Создание в форме компонентов списков.
4. Свойство Items (элементы). Нумерация элементов списков.
5. Метод SelectedIndex.
6. Предварительное редактирование списков.
7. Свойство Text (текст) компонента comboBox.
8. Программное редактирование строк компонентов списков.
9. Метод добавить элемент Add(Item) и его использование.
10. Метод ввести элемент Insert(Index,Item) и его использование.
11. Метод удалить элемент RemoveAt(Index) и его использование.

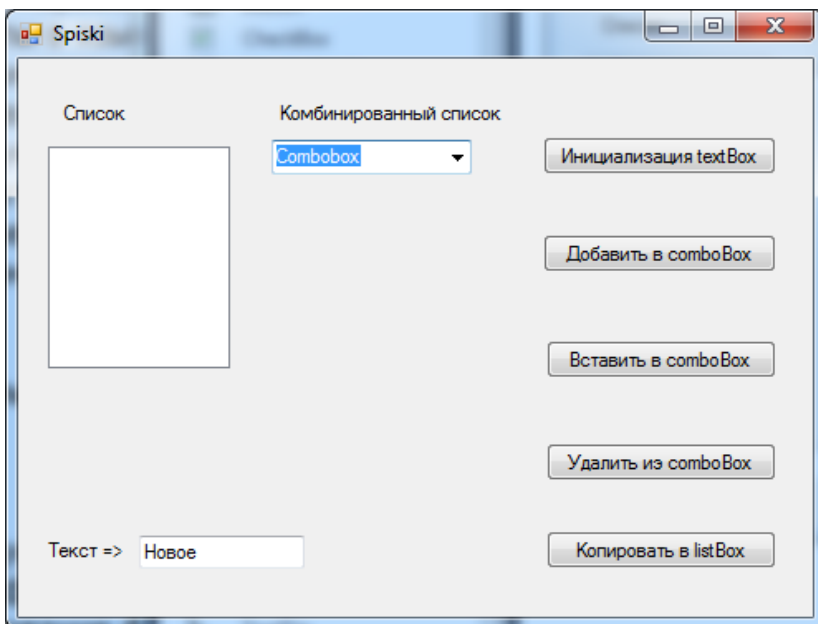
Задание. Создать проект работы со списками.

Пример. Создать проект работы со списками. В проекте

предусмотреть:

- Инициализацию поля ввода.
- Добавление строки из поля ввода в конец комбинированного списка.
- Вставку строки из поля ввода в комбинированный список в выделенное место.
- Удаление строки из комбинированного списка в выделенном месте.
- Копирование строки из комбинированного списка в выделенном месте в список

Рекомендуемая форма программы:



Форма содержит следующие компоненты:

- listBox1 - список.
- label1 – метка для заголовка (Список) компонента listBox1.
- comboBox1 - комбинированный список.
- label2 – метка для заголовка (Комбинированный список) компонента comboBox1.
- textBox1 - однострочный редактор для ввода текста. В него вводится стро-ка текста (по умолчанию - Новое).
- label3 – метка для заголовка (Текст = >) компонента textBox1.
- button1– кнопка "Инициализация textBox". Позволяет в textBox1 очистить поле Text и установить там курсор.
- button2 – кнопка "Добавить в comboBox" для заполнения компонента comboBox1.
- button3 – кнопка "Вставить в comboBox". Позволяет вставить строку в вы-деленное место в списке.
- Button4 – кнопка "Удалить из comboBox". Позволяет удаплить выделенную строку в списке.
- button5 – кнопка "Копировать в listBox". Позволяет копировать в listBox строку, выделяемую в comboBox.

Выполнение

Активируем ИСР. В главном меню исполняем команду **File=>New Proect**. Активируется окно выбора типа проекта. В нем нужно выбрать **WindowsFormApplication**. Отображается окно выбора скрытности кода, в котором выбираем обычной загрузки. Теперь отображаются окна проекта:

- Form1.cs [Design]. Конструктор формы, там пустая форма, которую нужно заполнить компонентами.
- Toolbox - компоненты.
- Properties - свойства.
- Form1.cs. Редактор кода с шаблоном кода модуля формы. Если окна нет, то правый щелчок по окну конструктора вызывает контекстное меню, в котором нужно исполнить команду **View Code**.
- Error List. Сообщения об ошибках.

Задание свойств формы. Выбираем объект **Form1** и подбираем его размеры и положение. В окне **Properties** назначаем свойства формы:

Свойство		Значение
English	Перевод	English
Name	Имя	Form1
Text	Текст	Списки

Заполнение формы. При заполнении формы в палитре компонент **Toolbox** выбирается нужный блок и переносится в форму. Там подбираются его положение и размеры. Затем в окне **Properties** назначаются свойства компонента. Большая часть свойств устанавливается автоматически, их редактировать не следует.

Для кнопок следует изменить текст, показав выполняемое действие. Также необходимо добавить функциональность кнопки, написав код обработчика события нажатия. При двойном щелчке по кнопке ИСР автоматически объявляет подпрограмму обработчика и в код автоматически заносит ее шаблон. Остается для каждого обработчика записать исполняемые команды.

Листинг программы

```
using System;  
using System.Windows.Forms;  
  
namespace WindowsFormsApp1Spiski  
{  
    public partial class Form1 : Form
```

```
{
    public Form1()
    {
        InitializeComponent();
    }

    private void Form1_Load(object sender, EventArgs e)
    {

    }

    private void button1_Click(object sender, EventArgs e)
    {
        textBox1.Text = "";
        textBox1.Focus();
    }

    private void button2_Click(object sender, EventArgs e)
    {
        comboBox1.Items.Add(textBox1.Text);
        textBox1.Text = "";
        textBox1.Focus();
    }

    private void button3_Click(object sender, EventArgs e)
    {
        int i = comboBox1.SelectedIndex;
        comboBox1.Items.Insert(i, textBox1.Text);
        textBox1.Text = "";
        textBox1.Focus();
    }

    private void button4_Click(object sender, EventArgs e)
    {
        int i = comboBox1.SelectedIndex;
        comboBox1.Items.RemoveAt(i);
    }

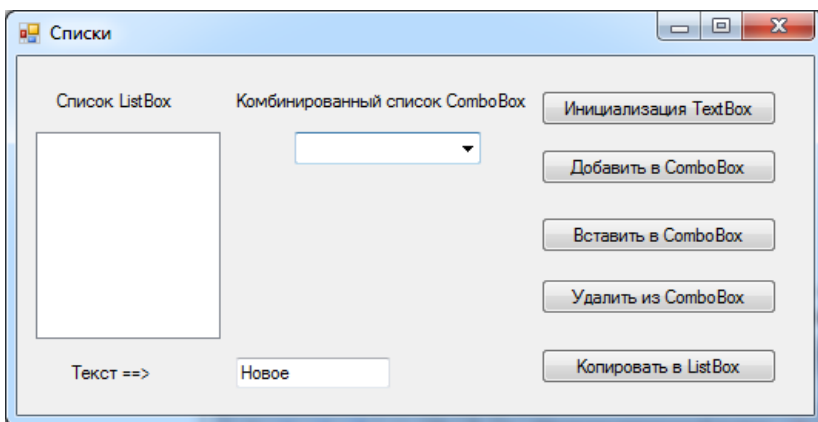
    private void button5_Click(object sender, EventArgs e)
    {
        int i = comboBox1.SelectedIndex;
```

```

        listBox1.Items.Add(comboBox1.Items[i]);
    }
}
}

```

Проверим программу в работе сначала без компиляции. Запускаем программу командой **Отладка=>Запуск без отладки**. Если допущены ошибки, то программа не исполняется, выводится окно с сообщением об этом. В окне ошибок все ошибки перечислены, их нужно исправить и посторонно запустить программу. Если ошибок нет, то программа запускается.

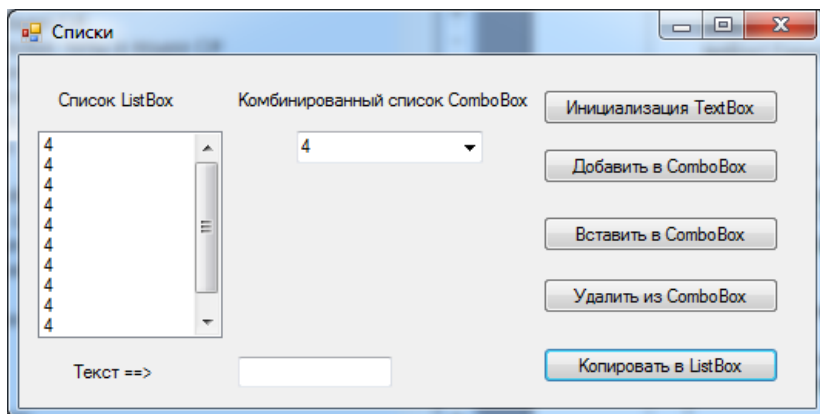


Теперь надо проверить правильность логики обработчиков.

- Инициализация textBox. Кнопка очищает textBox1 и передает ему фокус ввода. Курсор появляется в его поле.
- Добавить в comboBox. Для каждой строки в textBox1 заносим текст, кнопкой переносим его в конец списка. Повторяем операцию нужное число раз. Каждый раз проверяем список comboBox1, убеждаясь в появлении новой строки в конце.
- Вставить в comboBox. В textBox1 заносим строку текста для ввода в произвольно выбираемую позицию. В comboBox1 выделяем позицию для ввода. Кнопкой текст из textBox1 перемещается в выделенную строку comboBox1.
- Удалить из comboBox. В comboBox1 выделяем позицию для удаления. Кнопкой выделенную строку удаляется из comboBox1.
- Копировать в listBox. В comboBox выделяем позицию для копирования, ее текст перемещается в верхнюю строку. Кнопкой копируем ее в конец listBox. Каждый раз наблюдаем listBox, чтобы убедиться в появлении там

новой строки в очередной позиции. Обратите внимание на появление в listBox линейки прокрутки, когда его поле заполнено.

Приложение перед закрытием программы:



Если все правильно, то проект компилируется и собирается. Это можно сделать двумя способами:

- Команда Построение=>Построить решение генерирует исполняемый файл в отладочном режиме. Он размещается во вложенной папке проекта bin\debug.
- Команда Построение=>Перестроить решение генерирует исполняемый файл в выпускном режиме. Он размещается во вложенной папке проекта bin\release.