Федеральное агентство связи

Федеральное государственное бюджетное образовательное учреждение

высшее образования

«Сибирский Государственный Университет телекоммуникаций и информатики»

Лабораторная работа №1

Выполнил студент 2 курса

ИВТ, гр ИП-814

Якунин Андрей

Проверил

Ассистент ПМиК

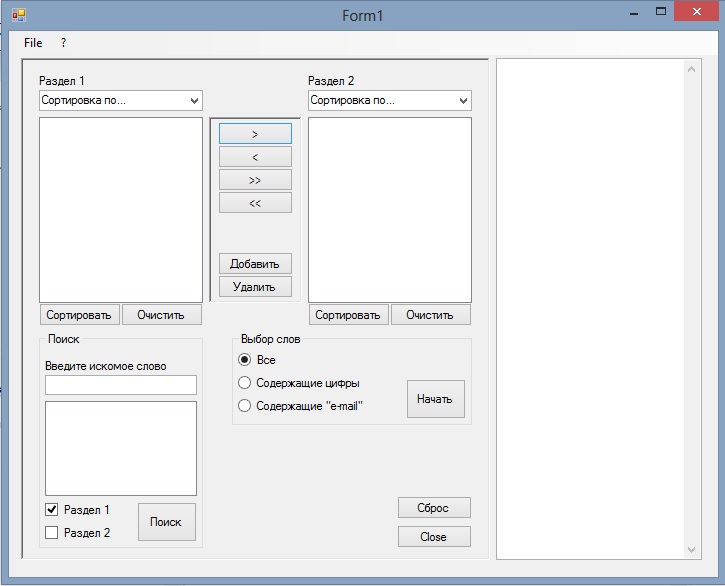
Суходоева Наталья Николаевна

Новосибирск 2020

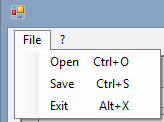
**Задание**

*1.* Создайте визуальную часть приложения, как показано на рисунке (или произвольным образом, сохранив в наличии все указанные компоненты). У компонентов Panel установите значения свойства BorderStyle (Fixed3D – выпуклая, утопленная). У компонентов RadioButton, CheckBox и ComboBox установите начальные значения свойств Checked, Checked и Text соответственно (см. рисунок).

ComboBox ListBox Panel



CheckBox GroupBox RadioButton TextBox

2. Создайте главное меню следующего вида:

По команде Open: должно появляться окно диалога открытия файла (используем OpenFileDialog), содержимое выбранного файл загружается в TextBox.

По команде Save: содержимое Раздела 2 сохраняется в выбранном файле (используем SaveFileDialog). Для этого надо последовательно перезаписать все строки из ListBox2 в файл. Для обращения к строкам ListBox используется: **ListBox1.Items**.

По команде Exit закрывается приложение.

По команде ? выдается информация об авторах данного программного продукта.

3. Создайте обработчики события “Нажатие на кнопку” для кнопок Сброс, Выход.

По кнопке *Сброс* – возврат формы в исходное состояние: очистка рабочей области (TextBox), области поиска, Раздела 1 и Раздела 2, а также установка CheckBox, RadioButton и ComboBox в начальное состояние.

По кнопке *Close*– выход из программы.

4. По кнопке *Начать*: содержимое TextBox разбивается на отдельные слова (признак окончания очередного слова - встреча пробела), которые последовательно помещаются в Раздел 1 в зависимости от значения радиокнопок. Один из способов реализации:

- из TextBox последовательно считываются строки;

- очередная считанная строка делится на слова;

- каждое слово проверяется на соответствие критерию выбора (определяется значениями радиокнопок (RadioButton): если соответствует -помещается в Раздел 1 (ListBox1). Чтобы поместить в ListBox данные (STRING) воспользуемся **ListBox1.Items.Add(**STRING**)**.

5. Реализовать:

- перемещение данных (полное или только выделенных строк) из одного Раздела в другой,

- удаление выделенных строк из Разделов 1 и 2,

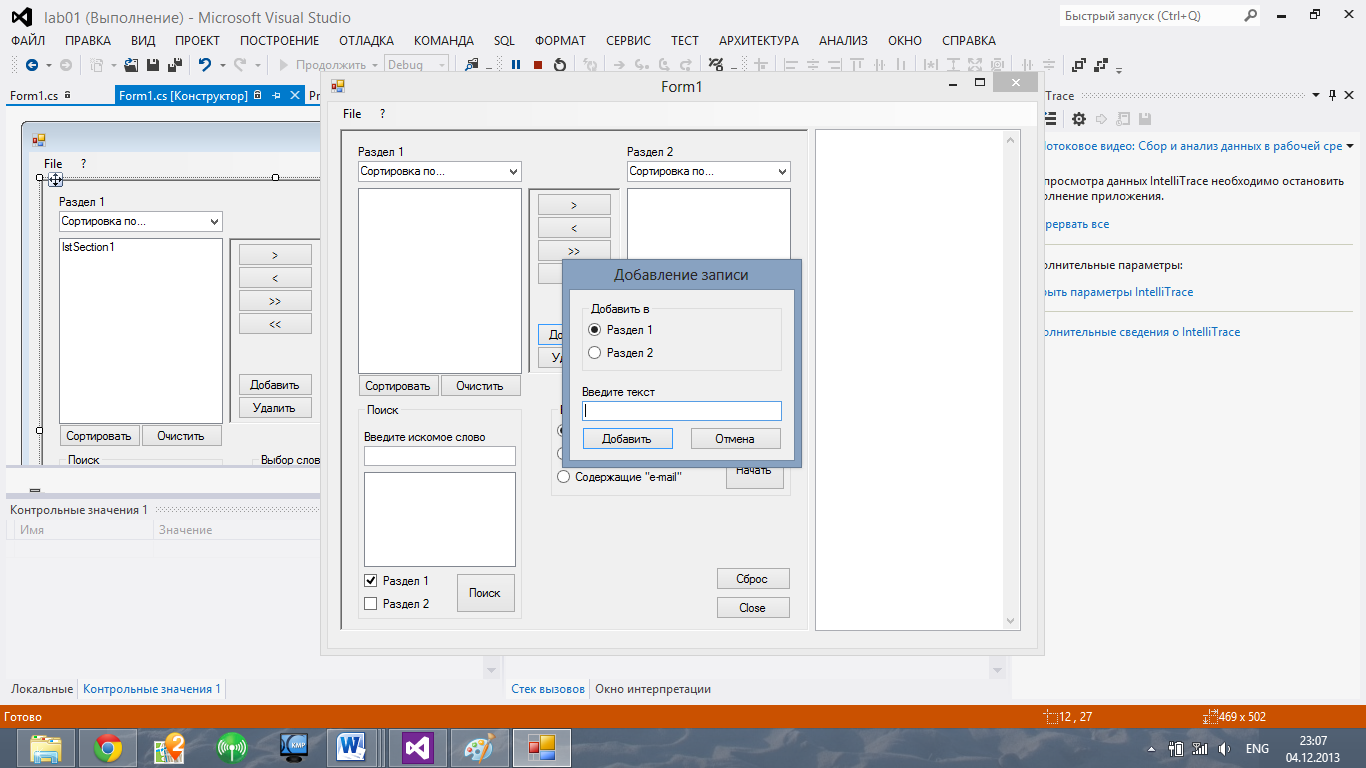
- добавление данных в Раздел 2.

**5.1.** Для перемещения строк из одного Раздела в другой (*полное*) необходимо вначале добавить все элементы из одного ListBox в другой (с помощью метода Add), а затем очистить ListBox (Clear), из которого мы производили перемещение. Чтобы узнать количество элементов в ListBox, можно воспользоваться ListBox1.Items.**Count**.

Для возможности выделять несколько строк в списке ListBox, установите его свойство SelectionMode = MultiExtended.

**5.2.** Для перемещения строк из одного Раздела в другой (частично), вначале необходимо найти выделенную строку (ListBox1.**SelectedItems[**номер выделенной строки**]** – true, если строка выделена). После того, как строка найдена, добавляем ее в другой раздел. Затем эту строку удаляем (используем ListBox1.Items.**RemoveAt(**номер удаляемой строки**)**).

**5.3.** Удаление выделенных строк из Раздела1(2) или одновременно из того и другого производится аналогично перемещению строк (частичное). Т.е. вначале мы ищем выделенные строки (Selected), затем удаляем найденную строку (RemoveAt). После этого переходим в другой раздел и проделываем то же самое. Предварительно (перед поиском) надо делать проверку наличия данных в Разделе (если пустой – проверка другого раздела …)

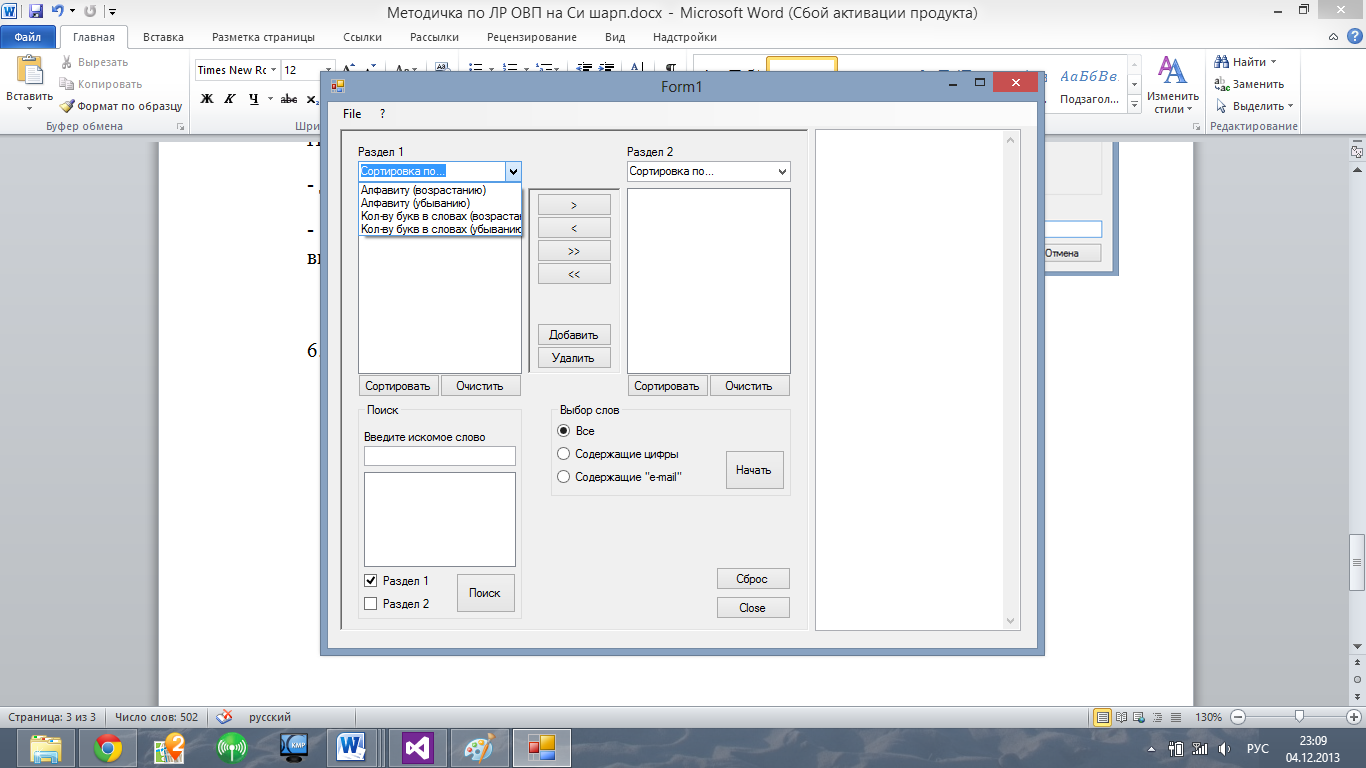
**5.4.** Для добавления данных в Раздел1(2) создадим модальную форму следующего вида.

По кнопке Добавить:

- данная форма закрывается;

- на главной форме в список выбранного Раздела помещается введенный текст.

1. Компонентам ComboBox следует присвоить значения:



Алфавиту (по возрастанию)

Алфавиту (по убыванию)

Длине слова (по возрастанию)

Длине слова (по убыванию)

1. Создайте обработчики для кнопок Сортировать, Очистить.

*Очистить* – очищает Раздел1(2) (т.е. ListBox1(2)). Чтобы произвести эту операцию используйте метод **Clear**.

*Сортировать* – производит сортировку данных, находящихся в Разделе1(2) по заданному критерию. Для организации сортировки необходимо сначала определить тип сортировки (т.е. по алфавиту (по возрастанию или убыванию) или по длине слов (по возрастанию или убыванию). Для этого используется проверка значения **ComboBox1.SelectedIndex**. Затем произвести сортировку данных любым известным способом сортировки. Для обращения к строкам ListBox используется: **ListBox1**.**Items.**.

Примечание: во время сортировки курсор мыши должен изменять свою форму (становиться песочными часами).

По окончании сортировки курсор должен восстанавливать свой исходный вид.

1. Реализовать поиск данных.

Для поиска необходимо вначале ввести ключ поиска (textBox1), очистить ListBox3, затем определить область поиска (т.е. Раздел1, Раздел2 или оба, для этого используйте свойство Checked компонента CheckBox). После этого начинаем последовательно просматривать элементы соответствующего раздела (ListBox….**Items**). При нахождение нужного элемента добавляем его в ListBox3.

**Листинг**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.IO;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Text.RegularExpressions;

using System.Collections;

namespace WindowsFormsApplication1

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void label3\_Click(object sender, EventArgs e)

{

}

private void groupBox1\_Enter(object sender, EventArgs e)

{

}

private void label2\_Click(object sender, EventArgs e)

{

}

private void openFileDialog1\_FileOk(object sender, CancelEventArgs e)

{

}

private void toolStripMenuItem1\_Click(object sender, EventArgs e)

{

}

private void openToolStripMenuItem\_Click(object sender, EventArgs e)

{

OpenFileDialog dlg = new OpenFileDialog();

if (dlg.ShowDialog() == DialogResult.OK)

{

StreamReader reader = new StreamReader(dlg.FileName, Encoding.Default);

txtFileContents.Text = reader.ReadToEnd();

reader.Close();

}

dlg.Dispose();

}

private void hotButton(object sender, KeyEventArgs e )

{

if ((e.KeyCode == Keys.O) && e.Control)

{

OpenFileDialog dlg = new OpenFileDialog();

if (dlg.ShowDialog() == DialogResult.OK)

{

StreamReader reader = new StreamReader(dlg.FileName, Encoding.Default);

txtFileContents.Text = reader.ReadToEnd();

reader.Close();

}

dlg.Dispose();

}

if (e.KeyCode == Keys.S && e.Control)

{

SaveFileDialog dlg = new SaveFileDialog();

if (dlg.ShowDialog() == DialogResult.OK)

{

StreamWriter writer = new StreamWriter(dlg.FileName);

for (int i = 0; i < lstSection2.Items.Count; i++)

{

writer.WriteLine((string)lstSection2.Items[i]);

}

writer.Close();

}

dlg.Dispose();

}

if (e.KeyCode == Keys.X && e.Alt)

{

this.Close();

}

}

private void saveToolStripMenuItem\_Click(object sender, EventArgs e)

{

SaveFileDialog dlg = new SaveFileDialog();

if (dlg.ShowDialog() == DialogResult.OK)

{

StreamWriter writer = new StreamWriter(dlg.FileName);

for (int i = 0; i < lstSection2.Items.Count; i++)

{

writer.WriteLine((string)lstSection2.Items[i]);

}

writer.Close();

}

dlg.Dispose();

}

private void exitToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Close();

}

private void btnClose\_Click(object sender, EventArgs e)

{

this.Close();

}

private void btnReset\_Click(object sender, EventArgs e)

{

lstSection1.Items.Clear();

lstSection2.Items.Clear();

lstSearchResults.Items.Clear();

txtFileContents.Clear();

chkSection1.Checked = true;

chkSection2.Checked = false;

rdoAll.Checked = true;

}

private void btnStart\_Click(object sender, EventArgs e)

{

lstSection1.Items.Clear();

lstSection2.Items.Clear();

lstSection1.BeginUpdate();

string[] strings = txtFileContents.Text.Split(new char[] { '\n', '\t', ' ' }, StringSplitOptions.RemoveEmptyEntries);

foreach (string s in strings)

{

string str = s.Trim();

if (str == String.Empty)

continue;

if (rdoAll.Checked)

{

lstSection1.Items.Add(str);

}

if (rdoDigits.Checked)

{

if (Regex.IsMatch(str, @"\d"))

lstSection1.Items.Add(str);

}

if (rdoEmails.Checked)

{

if (Regex.IsMatch(str, @"\w+@\w+\.\w+"))

lstSection1.Items.Add(str);

}

}

lstSection1.EndUpdate();

}

private void aboutToolStripMenuItem\_Click(object sender, EventArgs e)

{

MessageBox.Show("Якунин Андрей и Парфенцов Семен ИП-814");

}

private void DeleteSelectedItems(ListBox lstFrom)

{

lstFrom.Items.Remove(lstFrom.SelectedItem);

}

private void MoveSelectedItems(ListBox lstFrom, ListBox lstTo)

{

lstTo.BeginUpdate();

foreach (object item in lstFrom.SelectedItems)

{

lstTo.Items.Add(item);

}

lstTo.EndUpdate();

DeleteSelectedItems(lstFrom);

}

private void MoveAllItems(ListBox lstFrom, ListBox lstTo)

{

lstTo.Items.AddRange(lstFrom.Items);

lstFrom.Items.Clear();

}

private void btnDelete\_Click(object sender, EventArgs e)

{

DeleteSelectedItems(lstSection1);

DeleteSelectedItems(lstSection2);

}

private void btnMoveAllToSection1\_Click(object sender, EventArgs e)

{

MoveAllItems(lstSection2, lstSection1);

}

private void btnMoveAllToSection2\_Click(object sender, EventArgs e)

{

MoveAllItems(lstSection1, lstSection2);

}

private void btnMoveToSection1\_Click(object sender, EventArgs e)

{

MoveSelectedItems(lstSection2, lstSection1);

}

private void btnMoveToSection2\_Click(object sender, EventArgs e)

{

MoveSelectedItems(lstSection1, lstSection2);

}

private void SortListBox(ListBox list, int choose)

{

switch (choose)

{

case 0:

this.Cursor = Cursors.WaitCursor;

string[] stringsArray = new string[list.Items.Count];

list.Items.CopyTo(stringsArray, 0);

Array.Sort(stringsArray);

list.Items.Clear();

list.Items.AddRange(stringsArray);

this.Cursor = Cursors.Default;

break;

case 1:

this.Cursor = Cursors.WaitCursor;

string[] stringsArray1 = new string[list.Items.Count];

list.Items.CopyTo(stringsArray1, 0);

Array.Reverse(stringsArray1);

list.Items.Clear();

list.Items.AddRange(stringsArray1);

this.Cursor = Cursors.Default;

break;

case 2:

this.Cursor = Cursors.WaitCursor;

string[] stringsArray2 = new string[list.Items.Count];

list.Items.CopyTo(stringsArray2, 0);

Array.Sort(stringsArray2, (x, y) => x.Length.CompareTo(y.Length));

list.Items.Clear();

list.Items.AddRange(stringsArray2);

this.Cursor = Cursors.Default;

break;

case 3:

this.Cursor = Cursors.WaitCursor;

string[] stringsArray3 = new string[list.Items.Count];

list.Items.CopyTo(stringsArray3, 0);

Array.Sort(stringsArray3, (x, y) => x.Length.CompareTo(y.Length));

Array.Reverse(stringsArray3);

list.Items.Clear();

list.Items.AddRange(stringsArray3);

this.Cursor = Cursors.Default;

break;

default:

MessageBox.Show("Выберите критерий сортировки!", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

}

private void btnAdd\_Click\_1(object sender, EventArgs e)

{

Form2 dlg = new Form2();

dlg.Owner = this;

dlg.ShowDialog();

}

private void btnClearSection1\_Click\_1(object sender, EventArgs e)

{

lstSection1.Items.Clear();

}

private void btnClearSection2\_Click(object sender, EventArgs e)

{

lstSection2.Items.Clear();

}

private void btnFind\_Click(object sender, EventArgs e)

{

lstSearchResults.Items.Clear();

string textToFind = txtToFind.Text;

if (chkSection1.Checked)

{

foreach (string s in lstSection1.Items)

{

if (s.Contains(textToFind))

lstSearchResults.Items.Add(s);

}

}

if (chkSection2.Checked)

{

foreach (string s in lstSection2.Items)

{

if (s.Contains(textToFind))

lstSearchResults.Items.Add(s);

}

}

}

private void btnSortSection1\_Click\_1(object sender, EventArgs e)

{

SortListBox(lstSection1, SortOrderSection1.SelectedIndex);

}

private void btnSortSection2\_Click(object sender, EventArgs e)

{

SortListBox(lstSection2, SortOrderSection2.SelectedIndex);

}

private void menuStrip1\_ItemClicked(object sender, ToolStripItemClickedEventArgs e)

{

}

}

}

**Скриншоты**

