**Теоретические сведения**

Описание строк:

* string str = "abcdefg";

Вычисление длины строки:

* int length = str.Length;

Задаем пустую строку:

* str ="";

|  |  |
| --- | --- |
| **Метод** | **Назначение** |
| Contains() | Поиск в строке |
| StartsWith() | Начинается с … |
| EndsWith() | Заканчивается на … |
| IndexOf() | Находит первое вхождение заданной подстроки |
| Trim() | Удаление символов |
| Insert() | Вставка одной строки в другую |
| Remove() | Удаление части строки |
| Replace() | Замена части строки |
| ToUpper() | Делает заглавными все буквы в вызывающей строке |
| ToLower() | Делает строчными все буквы в вызывающей строке |
| Substring() | Получение подстроки из строки |

**Задание 1 Поиск абонента**

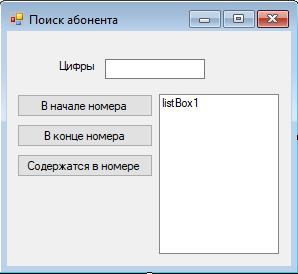
В задание необходимо найти пользователей телефонов по заданным первым цифрам телефона, последним цифрам или цифрам, имеющимся в середине номера.

Вначале объявите глобальные массивы (до описания процедур) номеров телефонов и соответствующих им фамилий абонентов:

string[] phone = new string[5];

string[] people = new string[5];

Разработайте форму вида:



В этой форме цифры, по которым мы будем искать информацию, вводятся в поле textBox1, а фамилии абонентов, чьи номера телефонов соответствуют маске поиска выводятся в списке listBox1.

При загрузке формы массивы заполняются значениями, например, такими:

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

{

phone[0] = "896045678";

phone[1] = "896545678";

phone[2] = "891255511";

phone[3] = "890855511";

phone[4] = "890322511";

people[0] = "Иванов";

people[1] = "Петров";

people[2] = "Сидоров";

people[3] = "Путилов";

people[4] = "Минин";

}

Код обработки первой кнопки:

private void button1\_Click(object sender, EventArgs e)

{

String a = textBox1.Text; // читаем маску поиска

listBox1.Items.Clear(); // обнуляем список

for (int i = 0; i < 5; i++) // перебираем элементы массива

{

if (phone[i].StartsWith(a) == true) listBox1.Items.Add(people[i]);

}

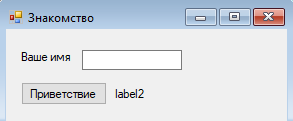
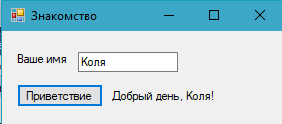
// если очередной номер начинается с указанных символов, выводим фамилию в список

}

**Самостоятельно** напишите коды обработки второй и третьей кнопки.

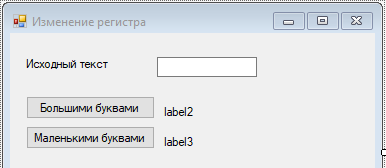
**Задание 2 Приветствие**

Разработайте форму, с которой считывается имя пользователя и по кнопке выводится приветствие для этого пользователя.

**Задание 3 Изменение регистра**

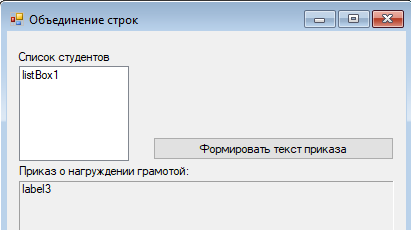
Разработайте форму, на которой введенный текст выводится либо большими прописными буквами, либо маленькими.

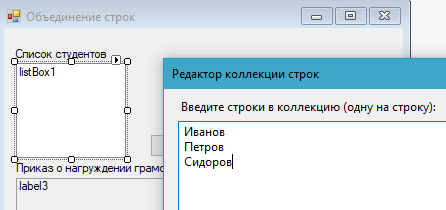


Подсказка: label2.Text = textBox1.Text.ToUpper();

**Задание 4 Объединение слов в предложение**

Разработайте форму, на которой данные из списка объединяются в единое предложение.

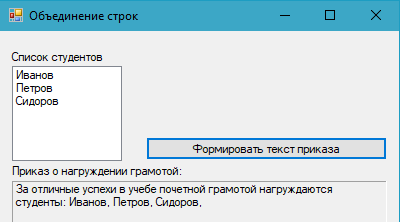
  
listBox1 заполняйте, используя свойство items (коллекцию).



Формат поля label3:



В результате работы программы должно получиться:



Начало кода обработки кнопки:

string SS;

string S = "За отличные успехи в учебе почетной грамотой награждаются студенты: ";

int k = listBox1.Items.Count;

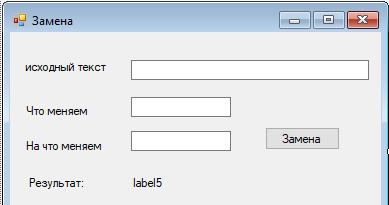
for (int i = 0; i < k; i++)

{

SS = (string)listBox1.Items[i];

…

**Задание 5 Замена слова в предложении**



Подсказка: label5.Text = (string)a.Replace(b, c);