Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное

учреждение высшего образования

ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Институт/Факультет - ИИТиАД

**Название работы -GUI интерфейс**

Отчет по лабораторной работе № 2

по дисциплине

Технология программирования

Выполнил cтудент ИСТб-20-2 М.А. Пигарев

номер группы подпись И. О. Фамилия

21.09.21

дата

Принял Доцент З.А. Бахвалова

21.09.21 Должность подпись И. О. Фамилия

дата

Иркутск –2021

Оглавление

[Задача на For 2](#_Toc84765633)

[Постановка задачи 2](#_Toc84765634)

[Словесное описание алгоритма 2](#_Toc84765635)

[Таблица спецификаций 3](#_Toc84765636)

[Таблица тестов 3](#_Toc84765637)

[Код программы 3](#_Toc84765638)

[Результаты тестов 6](#_Toc84765639)

[Задача на строки 7](#_Toc84765640)

[Постановка задачи 7](#_Toc84765641)

[Словесное описание алгоритма 7](#_Toc84765642)

[Таблица спецификаций 8](#_Toc84765643)

[Таблица тестов 8](#_Toc84765644)

[Код программы 8](#_Toc84765645)

[Результаты тестов 10](#_Toc84765646)

[Ссылка на Git 12](#_Toc84765647)

## Задача на For

### Постановка задачи

**Задача №14.** Дана последовательность натуральных чисел. Определить, есть ли в последовательности хотя бы одна n-ка одинаковых “соседних” чисел (n и элементы последовательности вводятся с клавиатуры). В случае положительного ответа определить порядковые номера чисел первой из таких пар.

### Словесное описание алгоритма

1. Заводим переменную строкового типа под набор изначальных чисел и массив строк, также заводим переменную под n (необходимое количество рядом стоящих чисел). Создаем целочисленным массив размером n (он необходим для хранения номеров рядом стоящих чисел)
2. Изначальную строчку с числами разбиваем на числа и записываем в массив строк
3. Устраиваем перебор всех элементов строкового массива через два цикла for,если удаётся найти n–ку одинаковых рядом стоящих чисел, то записываем их индексы в целочисленный массив
4. Выводим результат на экран

### Таблица спецификаций

|  |  |  |
| --- | --- | --- |
| Имя | Назначение | Тип/Диапазон |
| Входные величины | | |
| n | n-ка одинаковых “соседних” чисел | Действительное положительное число |
| s | Хранение элементов последовательности | Действительные положительные числа |
| Выходные величины | | |
| arr | Хранение порядковых номеров чисел | Действительные положительные числа |

### Таблица тестов

| Номер теста | Что проверяем | Входные данные | Выходные данные |
| --- | --- | --- | --- |
| 1 | Случай, когда есть пары | s=14 14 56 78 95  n=2 | arr={0 1} |
| 2 | Случай, когда нет пар | s=14 1 56 78 95  n=3 | Возможны пар не обнаружено |
| 3 | Случай, когда есть пары | s=3 4 6 7 7 7 8  n=3 | arr={3 4 5} |

### Код программы

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace \_2\_laba

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

textBox1.Text = Properties.Settings.Default.s.ToString();

textBox3.Text = Properties.Settings.Default.n.ToString();

textBox2.Text = Properties.Settings.Default.outText.ToString();

}

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

{

int n,a;

string s,outText;

try

{

n = int.Parse(textBox3.Text);

s = textBox1.Text;

string[] subs = s.Split(' ');

for(int i = 0; i < subs.Length; i++)

{

a = int.Parse(subs[i]);

}

}

catch (FormatException)

{

// сообщение об ошибке

MessageBox.Show("Некорректный ввод. В первое поле можно вводить только числа и пробелы между ними (в конце и в начале пробелы запрещены). Во второе поле можно вводить только одно число.", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

textBox1.Text = "";

textBox2.Text = "";

textBox3.Text = "";

return; // а затем прерываем обработчик

}

outText= Logic.Compare(n,s);

textBox2.Text = outText;

//----

Properties.Settings.Default.s = s;

Properties.Settings.Default.n = n.ToString();

Properties.Settings.Default.outText = outText;

Properties.Settings.Default.Save();

//----

}

private void button2\_Click(object sender, EventArgs e)

{

textBox1.Text ="";

textBox2.Text ="";

textBox3.Text ="";

}

private void textBox1\_KeyPress(object sender, KeyPressEventArgs e)

{

if (e.KeyChar == (char)Keys.Enter)

{

textBox3.Focus();

// button1\_Click( sender, e);

}

}

private void textBox3\_KeyPress(object sender, KeyPressEventArgs e)

{

if(e.KeyChar == (char)Keys.Enter)

{

button1\_Click( sender, e);

}

}

}

public class Logic

{

public static string Compare(int n,string s)

{

string outText="";

string[] subs = s.Split(' ');

int count;

//создаём массив для хранения индексов элементов

int[] arr = new int[n];

/\*перебираем массив цикл идёт до subs.Length-n+1 поскольку в дальнейшем будет

\* ещё один цикл for в котором будет веститсь перебор дальше. subs.Length-n+1

\* позволяет избежать выхода за пределы массива

\* \*/

for (int i = 0; i < subs.Length - n + 1; i++)

{

count = 0;

for (int j = i; j < n + i - 1; j++)

{

if (subs[j] == subs[j + 1])

{

arr[count] = j;

count++;

if (j == n + i - 2)

{

arr[count] = j + 1;

}

}

}

if (count + 1 == n)

{

for( i = 0; i < arr.Length; i++)

{

if (i == arr.Length - 1)

{

outText = outText + arr[i];

}

else

{

outText = outText + arr[i] + " ";

}

}

break;

}

}

if (arr[n - 1] == 0)

{

outText="Возможных пар не обнаружено";

}

return outText;

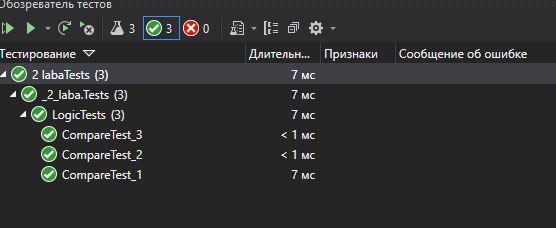
}

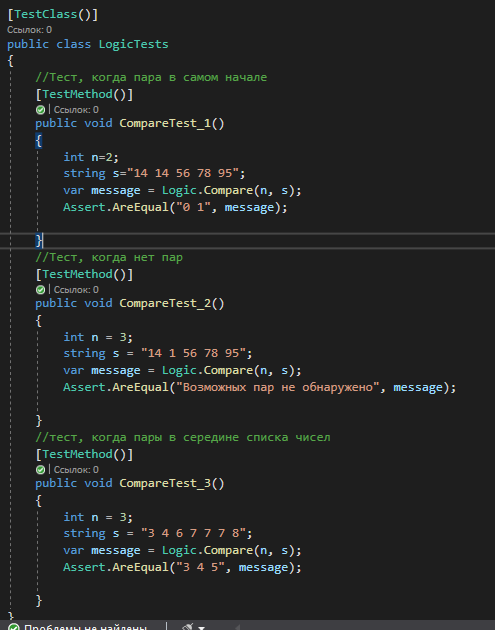
}

}

## Результаты тестов

|  |  |
| --- | --- |
| №Теста | Скриншот результата |
| 1 |  |
| 2 |  |
| 3 |  |





## Задача на строки

### Постановка задачи

**Задача №11.** Даны два слова. Для каждой буквы первого слова определить, входит ли она во второе слово. Повторяющиеся буквы первого слова не рассматривать. Например, если заданные слова процессор и информация, то для букв первого из них ответом должно быть: нет да да да нет нет.

### Словесное описание алгоритма

1. Заводим переменные для двух слов. Создаём пустую строчку, в которую мы будем записывать не повторяющиеся буквы первого слова
2. Берём по очереди буквы первого слова и проверяем есть ли они в созданной нами строке, если буквы там нет, то записываем её туда, после чего проверяем есть ли эта буква во втором слове, если буква там есть, то пропускаем итерацию цикла и переходим к следующей букве
3. Выводим результат на экран

### Таблица спецификаций

|  |  |  |
| --- | --- | --- |
| Имя | Назначение | Тип/Диапазон |
| Входные величины | | |
| s1 | Первое слово | Текст |
| s2 | Второе слово | Текст |
| Выходные величины | | |
| - | - | - |

### Таблица тестов

| Номер теста | Что проверяем | Входные данные | Выходные данные |
| --- | --- | --- | --- |
| 1 | Случай, когда есть все вхождения букв | рука  курица | Да Да Да Да |
| 2 | Случай, когда есть совпадения, а также повторяющиеся буквы | клопп  лес | Нет Да Нет Нет |
| 3 | Случай, когда нет совпадений | огт  рак | Нет Нет Нет |

### Код программы

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace \_2\_3

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

textBox1.Text = Properties.Settings.Default.s1.ToString();

textBox3.Text = Properties.Settings.Default.s2.ToString();

textBox2.Text = Properties.Settings.Default.outText.ToString();

}

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

{

int a,i=0;

bool trigger = false;

string s1 = textBox1.Text,outText;

string s2 = textBox3.Text,check="1234567890-=\_+()!@#$%^&\*/[]{}?<>.,` ";

try

{

while (trigger == false)

{

if (check.IndexOf(s1[i]) != -1)

{

a = int.Parse(" ");

trigger = true;

}

i++;

if (i == s1.Length)

{

trigger = true;

}

}

trigger = false;

i = 0;

while (trigger == false)

{

if (check.IndexOf(s2[i]) != -1)

{

a = int.Parse(" ");

trigger = true;

}

i++;

if (i == s2.Length)

{

trigger = true;

}

}

}

catch (FormatException)

{

// сообщение об ошибке

MessageBox.Show("Некорректный ввод. В оба поля можно вводить только буквы", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

textBox1.Text = "";

textBox2.Text = "";

textBox3.Text = "";

return; // а затем прерываем обработчик

}

outText = Logic.Compare(s1, s2);

textBox2.Text = outText;

Properties.Settings.Default.s1 = s1;

Properties.Settings.Default.s2 = s2;

Properties.Settings.Default.outText = outText;

Properties.Settings.Default.Save();

}

private void button2\_Click(object sender, EventArgs e)

{

textBox1.Text = "";

textBox2.Text = "";

textBox3.Text = "";

}

private void textBox1\_KeyPress(object sender, KeyPressEventArgs e)

{

if (e.KeyChar == (char)Keys.Enter)

{

textBox3.Focus();

// button1\_Click( sender, e);

}

}

private void textBox3\_KeyPress(object sender, KeyPressEventArgs e)

{

if (e.KeyChar == (char)Keys.Enter)

{

button1\_Click(sender, e);

}

}

}

public class Logic

{

public static string Compare(string s1, string s2)

{

string check = "",outText="";

for (int i = 0; i < s1.Length; i++)

{

//Проверяем есть ли данная буква в строке check, если есть, значит буква повторяющиеся

// поэтому пропускаем итерацию и переходим к следующему элементу, если её нет, значит она

//не повторяющиеся, а значит необходимо проверить её на вхождение во втором слове

if (check.IndexOf(s1[i]) == -1)

{

//проверяем есть ли эта буква во втором слове

if (s2.IndexOf(s1[i]) != -1)

{

outText = outText + "Да ";

}

else

{

outText = outText + "Нет ";

}

}

//добавляем букву к строке check

check = check + s1[i];

}

return outText;

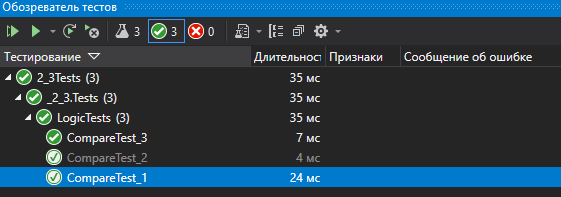
}

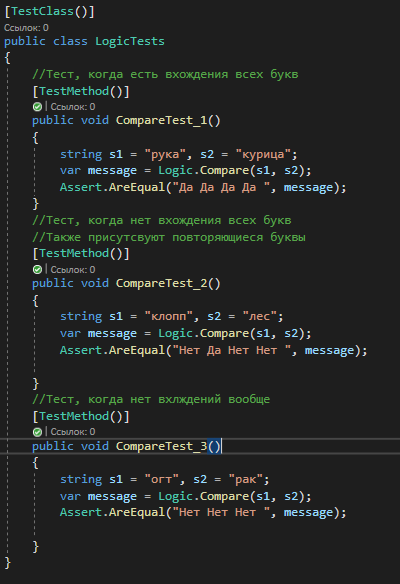
}

}

## Результаты тестов

|  |  |
| --- | --- |
| №Теста | Скриншот результата |
| 1 |  |
| 2 |  |
| 3 |  |





## Ссылка на Git

Задача на for: https://github.com/Bert-10/2\_2-laba.git

Задача на str: https://github.com/Bert-10/2\_3.git