МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное

учреждение высшего образования

«Южно-Уральский государственный университет

(национальный исследовательский университет)»

Высшая школа электроники и компьютерных наук

Кафедра системного программирования

**ОТЧЕТ**

о выполнении практического задания №1 (строки)

по дисциплине

«Структуры и алгоритмы обработки данных»

|  |  |
| --- | --- |
|  | Выполнил:  студент группы КЭ-204  Тельнов В.А.  Проверил:  ст. преподаватель кафедры СП  Петрова Л.Н. |

Челябинск - 2021

Оглавление

[Словестная постановка задачи 2](#_Toc64801074)

[Листинг программы 2](#_Toc64801075)

[Program.cs 2](#_Toc64801076)

[Form1.cs 3](#_Toc64801077)

[FormStringReplace.cs [Конструктор] 6](#_Toc64801078)

[Контрольный тест 6](#_Toc64801079)

[Выводы по практической работе 7](#_Toc64801080)

# Словестная постановка задачи

Вариант 25.

Функция заменяет символы строки из одного заданного алфавита на символы другого алфавита.

На вход подается 2 последовательности символов, разделенных пробелами, (алфавиты) и строка, которую требуется преобразовать.

Задача имеет решение если

1. Введены не пустые алфавиты

2. Каждому символу первого алфавита соответствует символ второго алфавита,

3. Символы первого алфавита не повторяются

4. Символы алфавитов состоят из одного знака

Ожидаемый результат – стока, в которой символы первого алфавита изменены на символы второго.

# Листинг программы

С#

## Program.cs

using System;

using System.Windows.Forms;

namespace lab1\_strings

{

static class Program

{

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new FormStringReplace());

}

}

}

## Form1.cs

using System;

using System.Windows.Forms;

namespace lab1\_strings

{

public partial class FormStringReplace : Form

{

private char[] \_firstAlphabet, \_secondAlphabet;

private string \_modifiedString;

public FormStringReplace()

{

InitializeComponent();

}

// Вызывается событием нажатия кнопки "Заменить"

// и преобразует строку

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

{

if (!ParseAlphabets())

return;

// Преобразование строки

char[] inputSting = textBoxInput.Text.ToCharArray();

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

{

for(int j = 0; j < \_firstAlphabet.Length; j++)

{

if(inputSting[i] == \_firstAlphabet[j])

{

inputSting[i] = \_secondAlphabet[j];

break;

}

}

}

\_modifiedString = new string(inputSting);

textBoxOutput.Text = \_modifiedString;

MessageBox.Show("Строка преобразована успешно\n", "Успех");

}

// Преобразует строки, введенные в первые 2 поля в алфавиты

// Возвращает true при успешном преобразовании и false при ошибке

private bool ParseAlphabets()

{

string[] firstAlphabetUnparsed, secondAlphabetUnparsed;

firstAlphabetUnparsed = textBoxFirstAlphabet.Text.Trim().Split(' ');

secondAlphabetUnparsed = textBoxSecondAlphabet.Text.Trim().Split(' ');

if(string.IsNullOrWhiteSpace(textBoxFirstAlphabet.Text))

{

MessageBox.Show("Введен пустой алфавит", "Ошибка");

return false;

}

if (firstAlphabetUnparsed.Length != secondAlphabetUnparsed.Length)

{

MessageBox.Show("Введены алфавиты разной длины,\nлибо введен двойной символ пробела", "Ошибка");

return false;

}

\_firstAlphabet = new char[firstAlphabetUnparsed.Length];

\_secondAlphabet = new char[secondAlphabetUnparsed.Length];

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

{

if (string.IsNullOrWhiteSpace(firstAlphabetUnparsed[i]) || string.IsNullOrWhiteSpace(secondAlphabetUnparsed[i]))

{

MessageBox.Show("Введен двойной символ пробела", "Ошибка");

return false;

}

if (firstAlphabetUnparsed[i].Length > 1 || secondAlphabetUnparsed[i].Length > 1)

{

MessageBox.Show("Элемент алфавита содержит больше одного символа", "Ошибка");

return false;

}

\_firstAlphabet[i] = firstAlphabetUnparsed[i][0];

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

{

if(\_firstAlphabet[j] == \_firstAlphabet[i])

{

MessageBox.Show("Обнаружен повторяющийся символ в первом алфавите", "Ошибка");

return false;

}

}

\_secondAlphabet[i] = secondAlphabetUnparsed[i][0];

}

return true;

}

}

}

## FormStringReplace.cs [Конструктор]

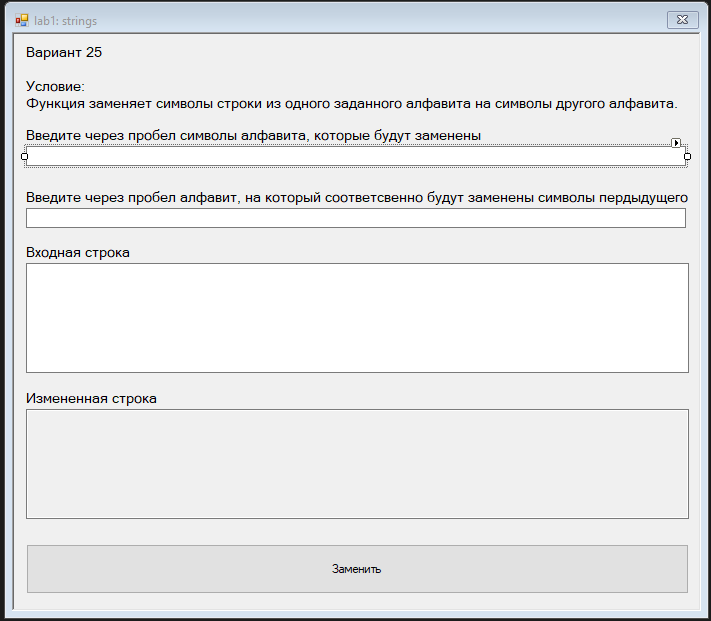


Рис. 1. Внешний вид конструктора

# Контрольный тест

|  |  |  |
| --- | --- | --- |
|  | Данные | Результат |
| Тест 1 | | |
| Алфавит 1 | e t | d~xd ~xampl~ |
| Алфавит 2 | ~ d |
| Входная строка | text example |
| Тест 2 | | |
| Алфавит 1 | о с я , | Не мызлю г\_рдый звет забавить!  Вниманье дружбы в\_злюбю!  Х\_тел бы ю тебе предзтавить  Зал\_г д\_зт\_йнее тебю! |
| Алфавит 2 | \_ з ю ! |
| Входная строка | Не мысля гордый свет забавить,  Вниманье дружбы возлюбя,  Хотел бы я тебе представить  Залог достойнее тебя, |
| Тест 3 | | |
| Алфавит 1 | e t t | Ошибка: обнаружен повторяющийся символ в первом алфавите |
| Алфавит 2 | ~ d m |
| Входная строка | text example |
| Тест 4 | | |
| Алфавит 1 | e t | Ошибка: элемент алфавита содержит больше одного символа |
| Алфавит 2 | ~ da |
| Входная строка | text example |
| Тест 5 | | |
| Алфавит 1 | e t p | Ошибка: алфавиты разной длины |

Табл. 1. Контрольные тесты

# Выводы по практической работе

Структура данных «Строка» может быть использована для работы с символьными последовательностями – их хранением, редактированием, вводом и выводом.