Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное

учреждение высшего образования

ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Институт/Факультет - ИИТиАД

Отчет по лабораторной работе № 2

по дисциплине

Технология программирования

Выполнил cтудент ИСТб-21-2 Н.И. Криницкий

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 2023

# Оглавление

[Оглавление 2](#_Toc84720986)

[Задача на if 3](#_Toc84720987)

[Постановка задачи 3](#_Toc84720988)

[Словесное описание алгоритма 3](#_Toc84720989)

[Таблица спецификаций 3](#_Toc84720990)

[Таблица тестов 3](#_Toc84720991)

[Код программы 5](#_Toc84720992)

[Задача на for 6](#_Toc84720993)

[Постановка задачи 6](#_Toc84720994)

[Словесное описание алгоритма 6](#_Toc84720995)

[Таблица спецификаций 7](#_Toc84720996)

[Таблица тестов 7](#_Toc84720997)

[Код программы 8](#_Toc84720998)

# Задача на if

## Постановка задачи

Задача № 9. Составить программу нахождения произведения двух наименьших из трех различных чисел.

## Словесное описание алгоритма

1. Вводим три различных числа a, b, c.
2. Проверяем числа на корректный ввод.
   1. Если одно из значений не является числом, то выводим «Введено не целое число».
   2. Если хоть одно число равно другому, выдается ошибка «Введите различные числа»
3. Если а максимальное, то произведение всех чисел делим на число а
4. Если b максимальное, то произведение всех чисел делим на число b
5. Если c максимальное, то произведение всех чисел делим на число c

## Таблица спецификаций

|  |  |  |
| --- | --- | --- |
| Имя | Назначение | Тип/Диапазон |
| Входные величины | | |
| a | Одно из трех различных чисел | Целое число |
| b | Одно из трех различных чисел | Целое число |
| c | Одно из трех различных чисел | Целое число |
| Выходные величины | | |
| max | Максимальное число | Целое число |
| proiz | Произведение 2 наименьших чисел | Целое число |

### Таблица тестов

| Номер теста | Что проверяем | Входные данные | Выходные данные |
| --- | --- | --- | --- |
| 1 | Произведение двух наименьших a\*c | a = 5, b = 17, c = 3 | Произведение двух наименьших a и c: 15 |
| 2 | Произведение двух наименьших b\*c | a = 33, b = 15, c = 5 | Произведение двух наименьших b и c: 75 |
| 3 | Произведение двух наименьших a\*b | a = 5, b = 99, c = 9 | Произведение двух наименьших a и b: 45 |
| 4 | Введено не целое число | a = ц, b = 99, c = 9 | «Введено не целое число» |
| 5 | Введены одинаковые числа. | a = 22, b = 15, c = 22 | «Введите различные числа |

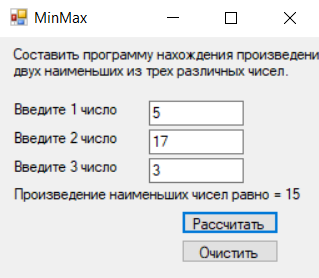


Рисунок 1- тест 1.

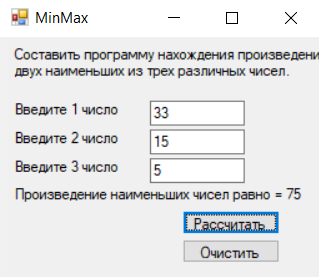


Рисунок 2- тест 2.

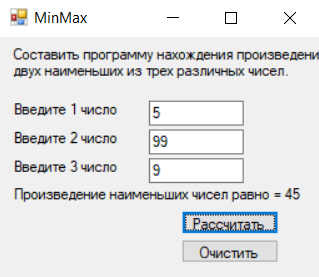


Рисунок 3- тест 3.

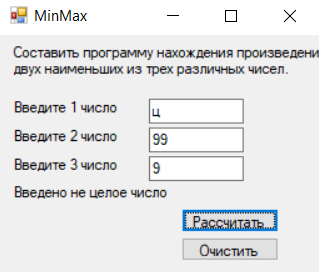


Рисунок 4- тест 4.

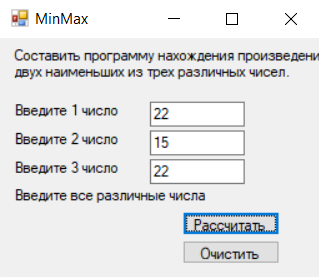


Рисунок 5- тест 5.

## 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 GUI1

{

public partial class MinMax : Form

{

public MinMax()

{

InitializeComponent();

this.KeyPreview = true;

this.KeyUp += new System.Windows.Forms.KeyEventHandler(this.Form1\_KeyUp);

this.button1.Click += new System.EventHandler(this.button1\_Click);

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

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

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

}

private void Form1\_KeyUp(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter) button1.PerformClick();

}

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

{

int a = 0, b = 0, c = 0;

string Out;

try

{

a = Convert.ToInt32(this.textBox1.Text);

b = Convert.ToInt32(this.textBox2.Text);

c = Convert.ToInt32(this.textBox3.Text);

Out = Logic.MinMax(a, b, c);

Properties.Settings.Default.a = a;

Properties.Settings.Default.b = b;

Properties.Settings.Default.c = c;

Properties.Settings.Default.Save();

}

catch (FormatException)

{

Out = "Введено не целое число";

}

label2.Text = Out;

}

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

{

textBox1.Clear();

textBox2.Clear();

textBox3.Clear();

label2.Text = "";

}

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

{

}

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

{

}

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

{

}

}

public class Logic

{

public static string MinMax(int a, int b, int c)

{

string Out2;

int max, proiz=0;

if (a > b)

{

max = a;

}

else

{

max = b;

}

if (max > c)

{

}

else

{

max = c;

}

if ((a == b) && (a == c)||(a==b)||(a==c)||(b==c))

{

Out2 = "Введите все различные числа";

}

else {

proiz = (a \* b \* c) / max;

Out2 =("Произведение наименьших чисел равно = " + proiz.ToString());

}

return (Out2);

}

}

}

# Задача на строки

## Постановка задачи

**Задача №7.** Дано предложение. Определить, сколько в нем одинаковых соседних букв. Пробелы не учитывать.

## Словесное описание алгоритма

1. Вводим предложение
2. Находим соседние одинаковые буквы и выводим.

## Таблица спецификаций

|  |  |  |
| --- | --- | --- |
| Имя | Назначение | Тип/Диапазон |
| Входные величины | | |
| str | Вводимое пользователем предложение | Строка символов |
| Выходные величины | | |
| sum | Кол-во одинаковых соседних букв | Целое число |

### Таблица тестов

| Номер теста | Что проверяем | Входные данные | Выходные данные |
| --- | --- | --- | --- |
| 1 | Ни одна пара соседних букв не будет одинаковой | qwerty | Количество одинаковых соседних букв: 0 |
| 2 | В предложении будут присутствовать | programmirovanie | Количество одинаковых соседних букв: 2 |
| 3 | В предложении будут присутствовать пробелы | Sumku ukrali | Количество одинаковых соседних букв: 0 |

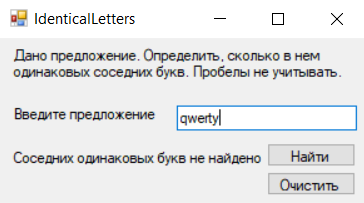


Рисунок 1- тест 1.

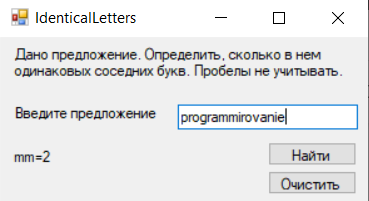


Рисунок 2- тест 2.

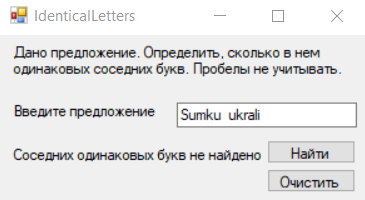


Рисунок 3- тест 3.

## 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 GUI21

{

public partial class Identicalletters : Form

{

public Identicalletters()

{

InitializeComponent();

this.KeyPreview = true;

this.KeyUp += new System.Windows.Forms.KeyEventHandler(this.Form1\_KeyUp);

this.button1.Click += new System.EventHandler(this.button1\_Click);

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

}

private void Form1\_KeyUp(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter) button1.PerformClick();

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

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

{

string str;

str = this.textBox1.Text;

string Out = Logic.LongStr(str);

Properties.Settings.Default.str = str;

Properties.Settings.Default.Save();

label3.Text = Out;

}

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

{

textBox1.Clear();

label3.Text = "";

}

}

public class Logic

{

public static string LongStr(string str)

{

str += " ";

int two = 0;

string Out2 = "";

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

{

if (str[i] == str[i + 1] && str[i]!=' ')

{

two++;

int j = i;

Out2 += str[i];

while (str[j] == str[j + 1])

{

j++;

Out2 += str[i];

if (str[j] != str[j + 1])

{

Out2 += ("=" + (j + 1 - i) + "\n");

i = j;

break;

}

}

}

}

if (two == 0)

{

Out2 = "Соседних одинаковых букв не найдено";

}

return Out2;

}

}

}