Лабораторная работа №2

«Запись арифметических выражений. Использование проверки корректности данных»

**Задание**

Выполнить расчёт следующих арифметических выражений на основе исходных данных. Кроме того, использовать проверку корректности ввода данных, которая заключается в том, что пользователь в поля ввода может вводить только цифры, знак «минус», запятую в качестве разделителя, а также разрешена клавиша «Backspace».

Выражение:

**Запись арифметических выражений на языке программирования.**

Вычисление арифметического выражения и сохранение в переменную представлено следующим образом:

double lab2\_result\_z = ( Math.Sin(x) / Math.Sqrt(1+Math.Pow(m,2) \* Math.Pow(Math.Sin(x),2))) - c \* m \* Math.Log(m \* x);

double lab2\_result\_s = Math.Exp(-a\*x) \* Math.Sqrt(x+1) + Math.Exp(-b\*x) \* Math.Sqrt(x+1.5);

* 1. **Алгоритм решения арифметического выражения.**

Входными данными для алгоритма являются пять значений: "m", "c", "x", "a" и "b". После ввода этих значений, они обрабатываются в соответствии с описанными ранее формулами, и результаты записываются в переменные "s" и "z". Алгоритм решения задачи представлен на рисунке 1.

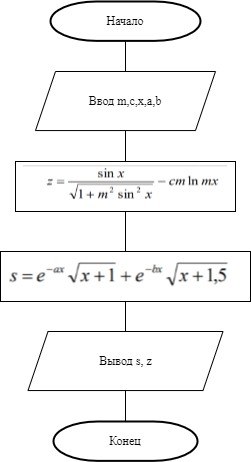


Рисунок 1 – Алгоритм решения арифметического выражения

* 1. **Код на языке С# для решения поставленной задачи**

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 Лабораторная\_работа\_\_2

{

public partial class CalcLab2 : Form

{

public CalcLab2()

{

InitializeComponent();

}

private void Input\_m\_lab2\_KeyPress(object sender, KeyPressEventArgs e)

{

string inputText = Input\_m\_lab2.Text;

// Разрешить обработку команды Backspace

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

{

return;

}

// Разрешить ввод цифр

if (char.IsDigit(e.KeyChar))

{

return;

}

// Разрешить ввод одного знака минус

if (e.KeyChar == '-' && !inputText.Contains("-") && !inputText.EndsWith(","))

{

return;

}

// Разрешить ввод одного знака запятой

if (e.KeyChar == ',' && !inputText.Contains(",") && !inputText.EndsWith("-"))

{

return;

}

// Запретить ввод, если символ является минусом или запятой и в поле уже присутствует соответствующий символ

if ((e.KeyChar == '-' && (inputText.Contains("-") || inputText.EndsWith(","))) ||

(e.KeyChar == ',' && (inputText.Contains(",") || inputText.EndsWith("-"))))

{

e.Handled = true;

return;

}

e.Handled = true;

}

private void Input\_c\_lab2\_KeyPress(object sender, KeyPressEventArgs e)

{

string inputText = Input\_c\_lab2.Text;

// Разрешить обработку команды Backspace

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

{

return;

}

// Разрешить ввод цифр

if (char.IsDigit(e.KeyChar))

{

return;

}

// Разрешить ввод одного знака минус

if (e.KeyChar == '-' && !inputText.Contains("-") && !inputText.EndsWith(","))

{

return;

}

// Разрешить ввод одного знака запятой

if (e.KeyChar == ',' && !inputText.Contains(",") && !inputText.EndsWith("-"))

{

return;

}

// Запретить ввод, если символ является минусом или запятой и в поле уже присутствует соответствующий символ

if ((e.KeyChar == '-' && (inputText.Contains("-") || inputText.EndsWith(","))) ||

(e.KeyChar == ',' && (inputText.Contains(",") || inputText.EndsWith("-"))))

{

e.Handled = true;

return;

}

e.Handled = true;

}

private void Input\_x\_lab2\_KeyPress(object sender, KeyPressEventArgs e)

{

string inputText = Input\_x\_lab2.Text;

// Разрешить обработку команды Backspace

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

{

return;

}

// Разрешить ввод цифр

if (char.IsDigit(e.KeyChar))

{

return;

}

// Разрешить ввод одного знака минус

if (e.KeyChar == '-' && !inputText.Contains("-") && !inputText.EndsWith(","))

{

return;

}

// Разрешить ввод одного знака запятой

if (e.KeyChar == ',' && !inputText.Contains(",") && !inputText.EndsWith("-"))

{

return;

}

// Запретить ввод, если символ является минусом или запятой и в поле уже присутствует соответствующий символ

if ((e.KeyChar == '-' && (inputText.Contains("-") || inputText.EndsWith(","))) ||

(e.KeyChar == ',' && (inputText.Contains(",") || inputText.EndsWith("-"))))

{

e.Handled = true;

return;

}

e.Handled = true;

}

private void Input\_a\_lab2\_KeyPress(object sender, KeyPressEventArgs e)

{

string inputText = Input\_a\_lab2.Text;

// Разрешить обработку команды Backspace

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

{

return;

}

// Разрешить ввод цифр

if (char.IsDigit(e.KeyChar))

{

return;

}

// Разрешить ввод одного знака минус

if (e.KeyChar == '-' && !inputText.Contains("-") && !inputText.EndsWith(","))

{

return;

}

// Разрешить ввод одного знака запятой

if (e.KeyChar == ',' && !inputText.Contains(",") && !inputText.EndsWith("-"))

{

return;

}

// Запретить ввод, если символ является минусом или запятой и в поле уже присутствует соответствующий символ

if ((e.KeyChar == '-' && (inputText.Contains("-") || inputText.EndsWith(","))) ||

(e.KeyChar == ',' && (inputText.Contains(",") || inputText.EndsWith("-"))))

{

e.Handled = true;

return;

}

e.Handled = true;

}

private void Input\_b\_lab2\_KeyPress(object sender, KeyPressEventArgs e)

{

string inputText = Input\_b\_lab2.Text;

// Разрешить обработку команды Backspace

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

{

return;

}

// Разрешить ввод цифр

if (char.IsDigit(e.KeyChar))

{

return;

}

// Разрешить ввод одного знака минус

if (e.KeyChar == '-' && !inputText.Contains("-") && !inputText.EndsWith(","))

{

return;

}

// Разрешить ввод одного знака запятой

if (e.KeyChar == ',' && !inputText.Contains(",") && !inputText.EndsWith("-"))

{

return;

}

// Запретить ввод, если символ является минусом или запятой и в поле уже присутствует соответствующий символ

if ((e.KeyChar == '-' && (inputText.Contains("-") || inputText.EndsWith(","))) ||

(e.KeyChar == ',' && (inputText.Contains(",") || inputText.EndsWith("-"))))

{

e.Handled = true;

return;

}

e.Handled = true;

}

private void Make\_result\_lab2\_Click(object sender, EventArgs e)

{

// Получаем значения из полей ввода

string input\_m = Input\_m\_lab2.Text;

string input\_c = Input\_c\_lab2.Text;

string input\_x = Input\_x\_lab2.Text;

string input\_a = Input\_a\_lab2.Text;

string input\_b = Input\_b\_lab2.Text;

// Переменная для проверки успешности ввода

bool validInput = true;

// Проверка и парсинг значения m

if (!double.TryParse(input\_m, out double m))

{

validInput = false;

MessageBox.Show("Неверный формат данных для m.", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

// Проверка и парсинг значения c

if (!double.TryParse(input\_c, out double c))

{

validInput = false;

MessageBox.Show("Неверный формат данных для c.", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

// Проверка и парсинг значения x

if (!double.TryParse(input\_x, out double x))

{

validInput = false;

MessageBox.Show("Неверный формат данных для x.", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

// Проверка и парсинг значения a

if (!double.TryParse(input\_a, out double a))

{

validInput = false;

MessageBox.Show("Неверный формат данных для a.", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

// Проверка и парсинг значения b

if (!double.TryParse(input\_b, out double b))

{

validInput = false;

MessageBox.Show("Неверный формат данных для b.", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

// Выполнение вычислений, если ввод данных был успешным //

if (validInput)

{

//Вычесление z

double lab2\_result\_z = ( Math.Sin(x) / Math.Sqrt(1+Math.Pow(m,2) \* Math.Pow(Math.Sin(x),2))) - c \* m \* Math.Log(m \* x);

lab2\_Textbox\_result\_z.Text = lab2\_result\_z.ToString();

//Вычесление s

double lab2\_result\_s = Math.Exp(-a\*x) \* Math.Sqrt(x+1) + Math.Exp(-b\*x) \* Math.Sqrt(x+1.5);

lab2\_Textbox\_result\_s.Text = lab2\_result\_s.ToString();

}

}

private void Lab2\_exit\_Click(object sender, EventArgs e)

{

this.Close(); // Закрывает последнюю страничку

}

}

}

* 1. **Результаты выполнения программы**

Вся необходимая информация об учащемся и проверяющем указана на титульной форме (см. рисунок 2)

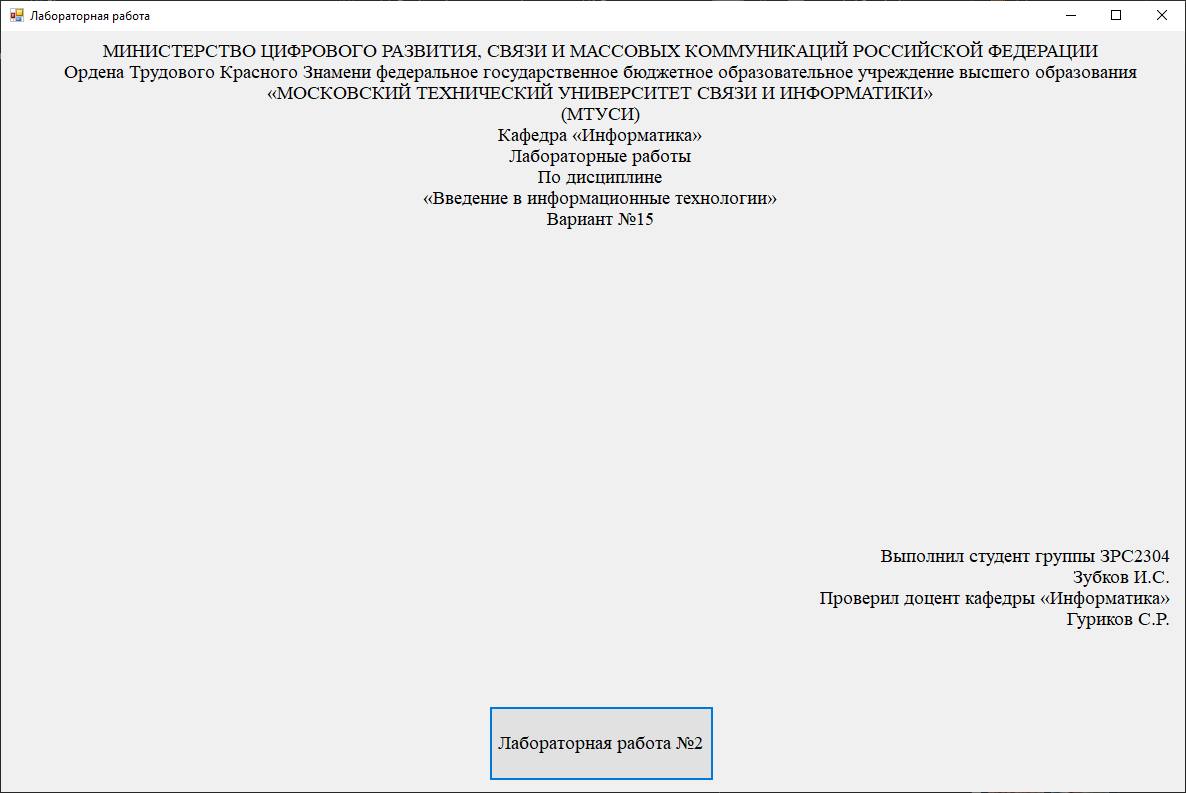


Рисунок 2 – Титульное окно

На второй форме предоставлены краткая информация о работе и формулы. (см. рисунок 3)

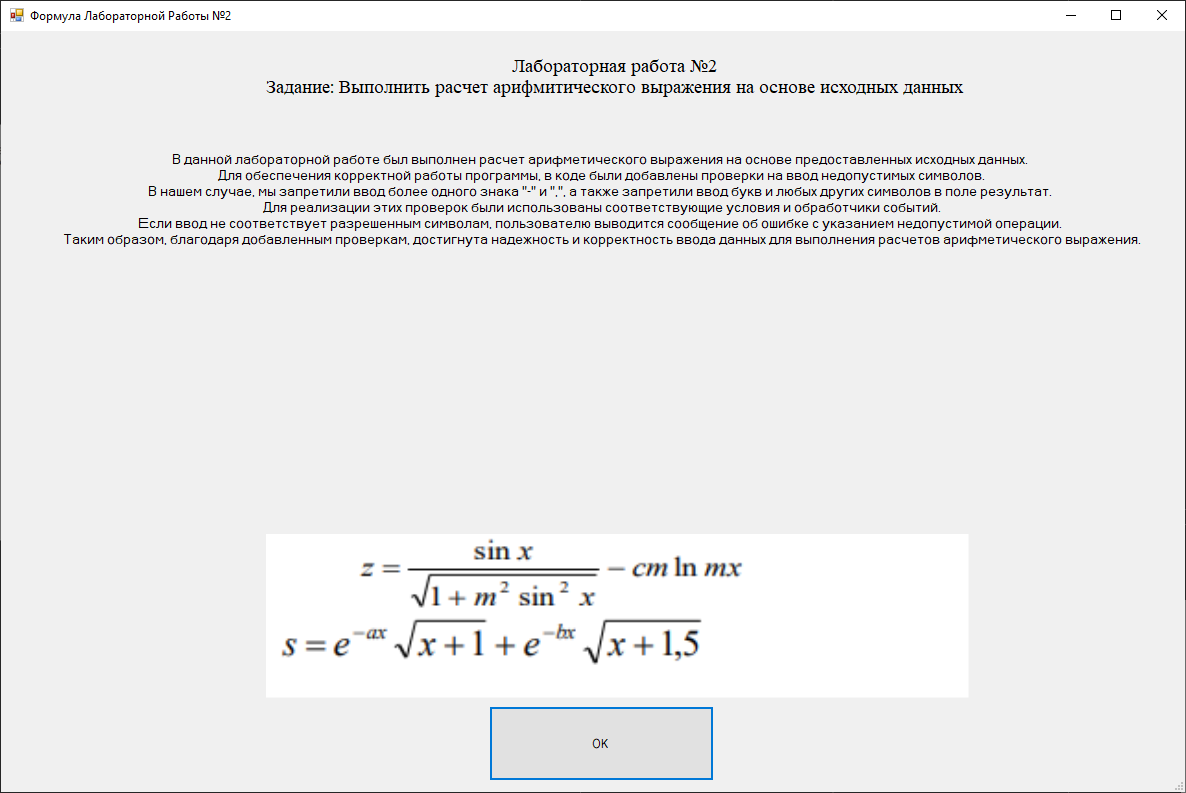


Рисунок 3 – Формула и краткая информация

Визуальное представление решения задачи с указанными данными представлено на рисунке 4.

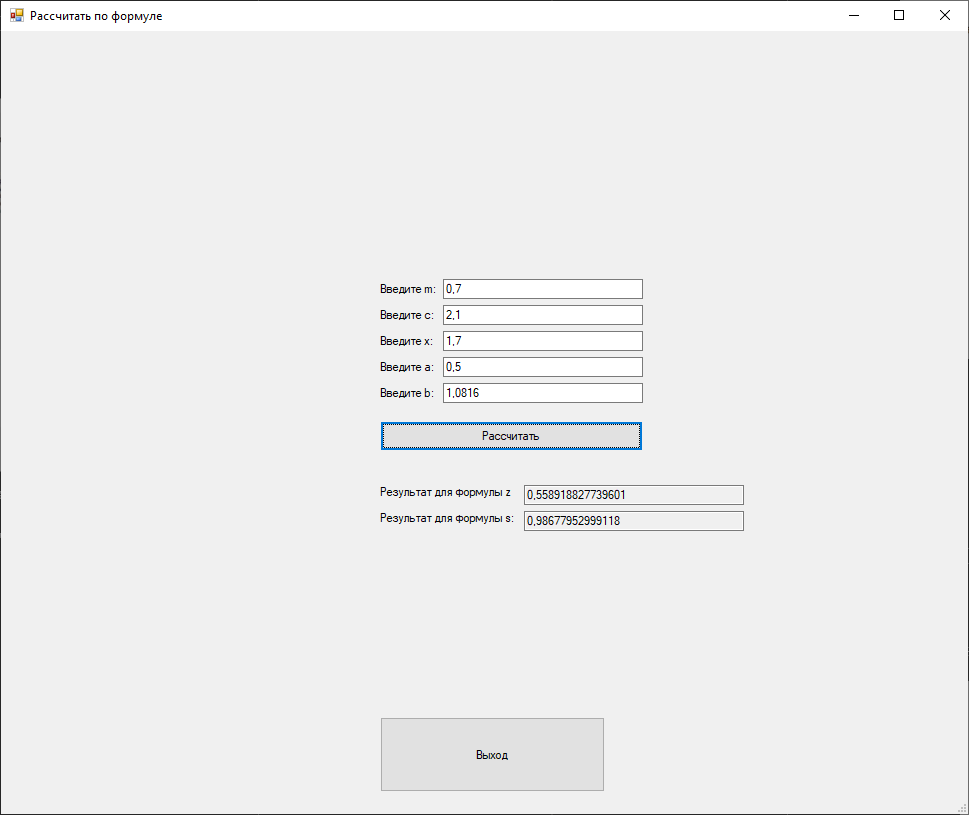


Рисунок 4 – Решение задачи

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Гуриков, С. Р. Введение в программирование на языке Visual C#: учебное пособие / С. Р. Гуриков. — Москва: ФОРУМ: ИНФРА-М, 2019. — 447 с. — (Среднее профессиональное образование). - ISBN 978-5-00091-540-0. - Текст: электронный. - URL: <https://znanium.com/catalog/product/1012397> (дата обращения: 10.10.2023).

2. ГОСТ 7.32-2017. Межгосударственный стандарт. Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления" (введен в действие Приказом Росстандарта от 24.10.2017 N 1494-ст) (дата обращений 10.10.2023).

3. ГОСТ 19.701-90 (ИСО 5807-85) “Единая система программной документации СХЕМЫ АЛГОРИТМОВ, ПРОГРАММ, ДАННЫХ И СИСТЕМ Обозначения условные и правила выполнения”. (дата обращения 10.10.2023).