Министерство Цифрового Развития, Связи и Массовых Коммуникаций Российской Федерации

ордена Трудового Красного Знамени

федеральное бюджетное образовательное учреждение высшего образования «Московский технический университет связи и информатики»

Факультет информационных технологий

Кафедра «Информатика»

Лабораторная работа №2

по дисциплине «Технологии программирования»

Выполнил:

Студент гр. БСТ2102

Засс М.С.

Научный руководитель:

Заведующий кафедры «Информатика» Волков А.И.

Москва – 2022

Оглавление

[1 Постановка задачи 3](#_Toc115213861)

[2 Схемы алгоритмов 3](#_Toc115213862)

[3 Текст программы 7](#_Toc115213863)

[4 Результаты тестирования программы 9](#_Toc115213864)

[Выводы 12](#_Toc115213865)

# 1 Постановка задачи

Разработать программу Калькулятор с однострочным полем для отображения данных. Убедиться, что разработанный калькулятор правильно выполняет все допустимые операции с различными исходными данными, включая «цепочки» операций.

Разработать калькулятор, который будет запрещать деление на нуль, выводя сообщение об этом. Также который будет различать вещественные числа и будет производить с ними те же операции, что и с простыми. Кроме этого необходимо сделать так, чтобы калькулятор не разрешал ввод букв или символов, не являющихся запятой или минусом. Убедиться, что данные символы калькулятор не разрешает вводить по несколько раз. Вместе с этим калькулятор должен выводить сообщение о запрете действия, если поля ввода или одно из них пустые.

# 2 Схемы алгоритмов

На рисунке 1 представлена схема алгоритма функции сложения.

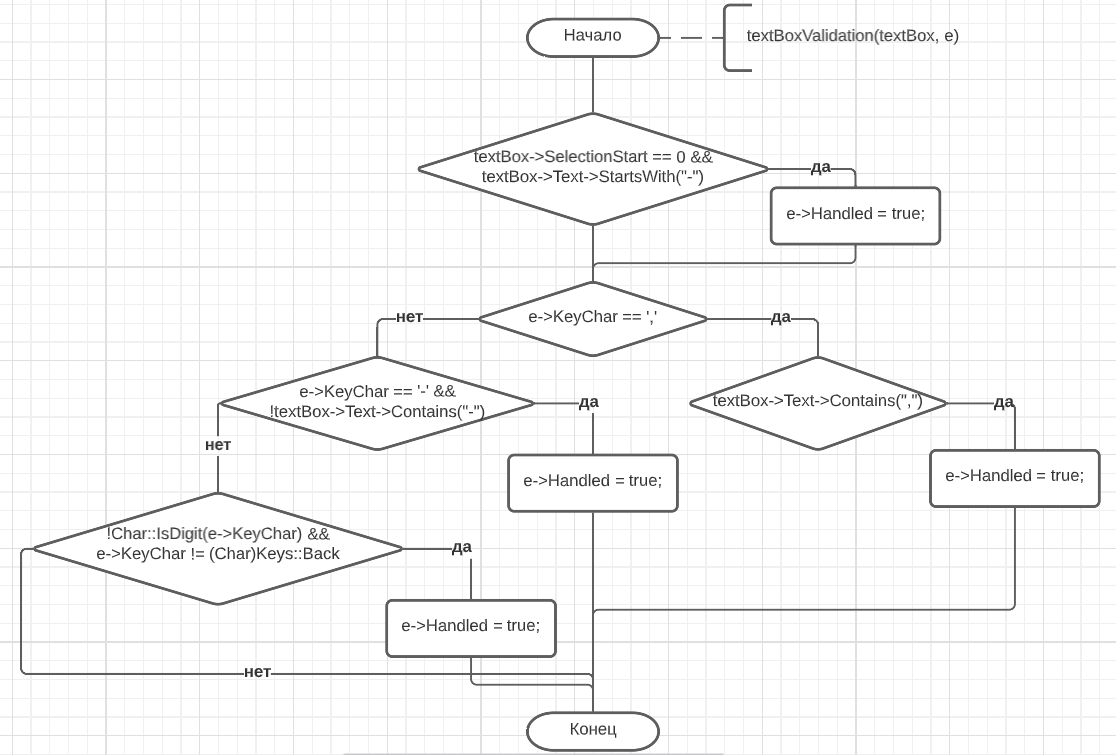


Рисунок 1 – Схема алгоритма функции сложения

На рисунке 2 представлена схема алгоритма функции вычитания.

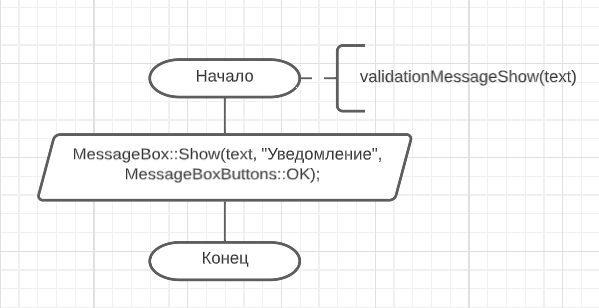


Рисунок 2 – Схема алгоритма функции вычитания

На рисунке 3 представлена схема алгоритма функции умножения.

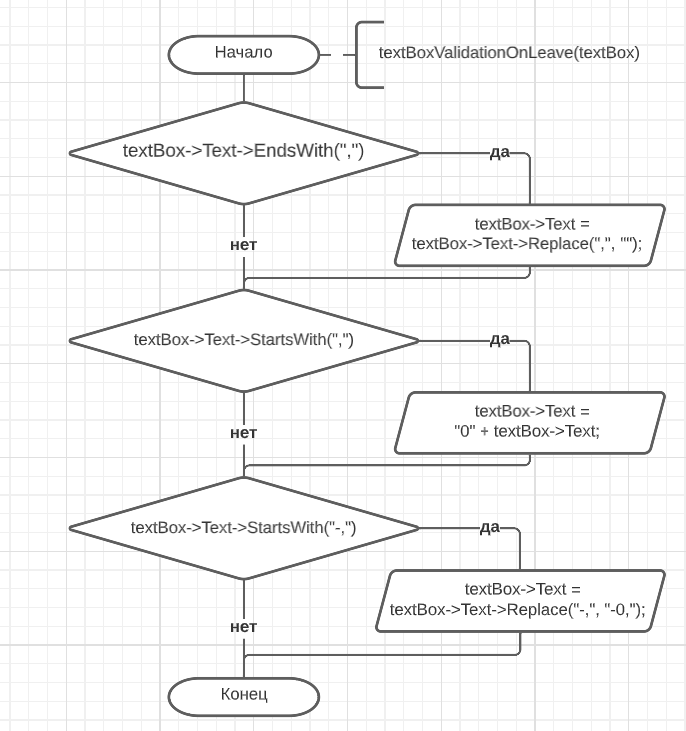


Рисунок 3 – Схема алгоритма функции умножения

На рисунке 4 представлена схема алгоритма функции деления.

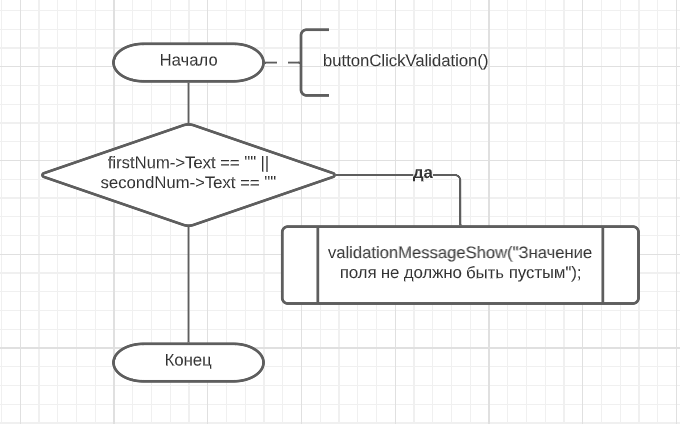


Рисунок 4 – Схема алгоритма функции деления

На рисунке 5 представлена схема алгоритма для первого поля ввода функции, запрещающей вводить какие-либо символы помимо запятой, минуса и цифр, а также несколько данных символов.

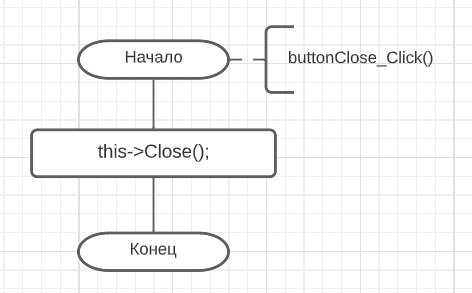


Рисунок 5 – Схема алгоритма функции сложения

На рисунке 6 представлена схема алгоритма для второго поля ввода функции, запрещающей вводить какие-либо символы помимо запятой, минуса и цифр, а также несколько данных символов.

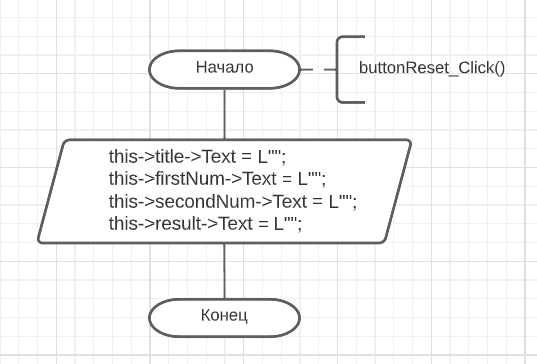
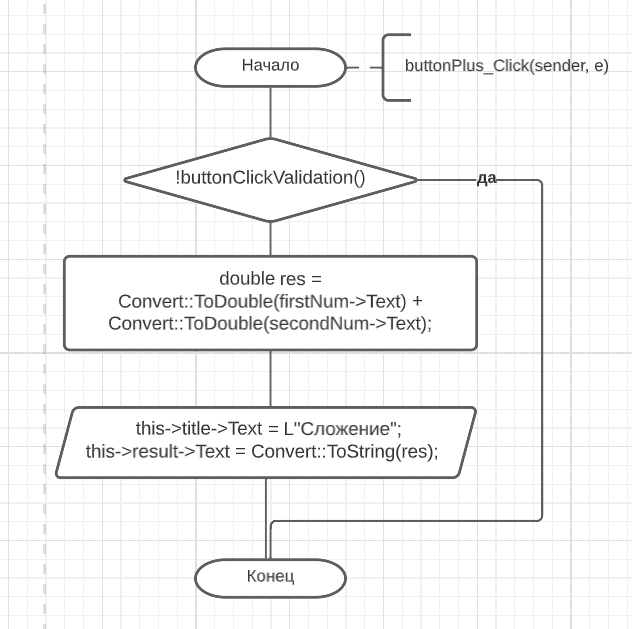
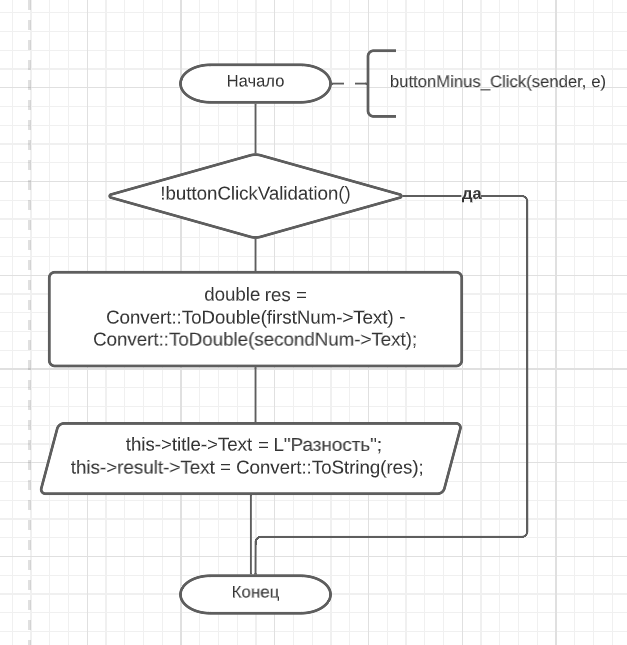
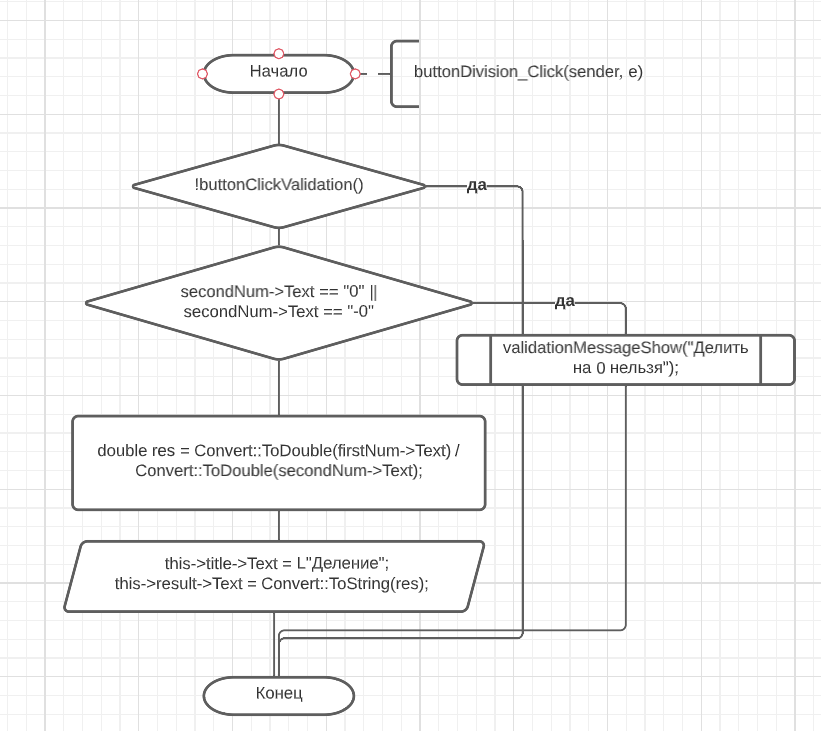
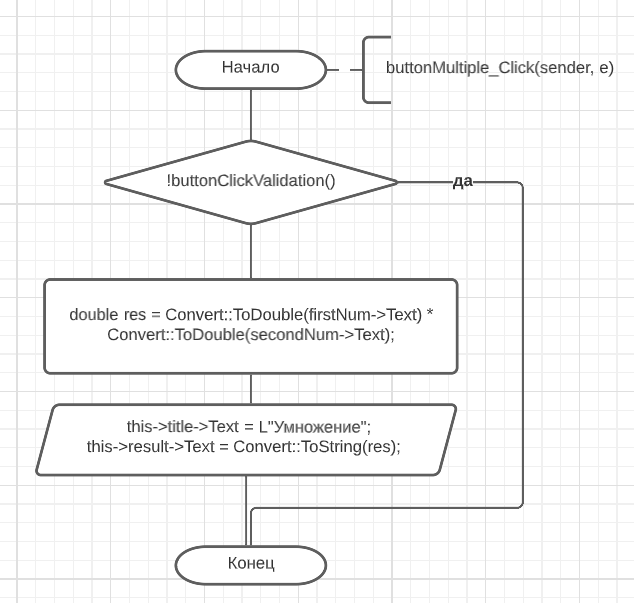


Рисунок 6 – Схема алгоритма функции деления









# 3 Текст программы

#pragma once

namespace Calc {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

#pragma endregion

// Helper functions

private: void validationMessageShow(String^ text) {

MessageBox::Show(text, "Уведомление", MessageBoxButtons::OK);

}

private: void textBoxValidation(Windows::Forms::TextBox^ textBox, Windows::Forms::KeyPressEventArgs^ e) {

// Disable typing before "-"

if (textBox->SelectionStart == 0 && textBox->Text->StartsWith("-")) {

e->Handled = true;

}

// Access ","

if (e->KeyChar == ',') {

if (textBox->Text->Contains(",")) {

e->Handled = true;

}

}

// Access "-"

else if (e->KeyChar == '-' && !textBox->Text->Contains("-")) {

e->Handled = true;

textBox->Text = "-" + textBox->Text;

textBox->SelectionStart = textBox->Text->Length;

}

// Access digits and backspace

else if (!Char::IsDigit(e->KeyChar) && e->KeyChar != (Char)Keys::Back) {

e->Handled = true;

}

}

private: void textBoxValidationOnLeave(Windows::Forms::TextBox^ textBox) {

if (textBox->Text->EndsWith(",")) {

textBox->Text = textBox->Text->Replace(",", "");

}

if (textBox->Text->StartsWith(",")) {

textBox->Text = "0" + textBox->Text;

}

if (textBox->Text->StartsWith("-,")) {

textBox->Text = textBox->Text->Replace("-,", "-0,");

}

}

private: bool buttonClickValidation() {

if (firstNum->Text == "" || secondNum->Text == "") {

validationMessageShow("Значение поля не должно быть пустым");

return false;

}

return true;

}

// Event Listeners

private: System::Void buttonClose\_Click(System::Object^ sender, System::EventArgs^ e) {

this->Close();

}

private: System::Void buttonReset\_Click(System::Object^ sender, System::EventArgs^ e) {

this->title->Text = L"";

this->firstNum->Text = L"";

this->secondNum->Text = L"";

this->result->Text = L"";

}

private: System::Void buttonPlus\_Click(System::Object^ sender, System::EventArgs^ e) {

if (!buttonClickValidation()) {

return;

}

this->title->Text = L"Сложение";

double res = Convert::ToDouble(firstNum->Text) + Convert::ToDouble(secondNum->Text);

this->result->Text = Convert::ToString(res);

}

private: System::Void buttonMinus\_Click(System::Object^ sender, System::EventArgs^ e) {

if (!buttonClickValidation()) {

return;

}

this->title->Text = L"Разность";

double res = Convert::ToDouble(firstNum->Text) - Convert::ToDouble(secondNum->Text);

this->result->Text = Convert::ToString(res);

}

private: System::Void buttonDivision\_Click(System::Object^ sender, System::EventArgs^ e) {

if (!buttonClickValidation()) {

return;

}

if (secondNum->Text == "0" || secondNum->Text == "-0") {

validationMessageShow("Делить на 0 нельзя");

return;

}

this->title->Text = L"Деление";

double res = Convert::ToDouble(firstNum->Text) / Convert::ToDouble(secondNum->Text);

this->result->Text = Convert::ToString(res);

}

private: System::Void buttonMultiple\_Click(System::Object^ sender, System::EventArgs^ e) {

if (!buttonClickValidation()) {

return;

}

this->title->Text = L"Умножение";

double res = Convert::ToDouble(firstNum->Text) \* Convert::ToDouble(secondNum->Text);

this->result->Text = Convert::ToString(res);

}

private: System::Void firstNum\_KeyPress(System::Object^ sender, System::Windows::Forms::KeyPressEventArgs^ e) {

this->textBoxValidation(this->firstNum, e);

}

private: System::Void secondNum\_KeyPress(System::Object^ sender, System::Windows::Forms::KeyPressEventArgs^ e) {

this->textBoxValidation(this->secondNum, e);

}

private: System::Void firstNum\_Leave(System::Object^ sender, System::EventArgs^ e) {

textBoxValidationOnLeave(this->firstNum);

}

private: System::Void secondNum\_Leave(System::Object^ sender, System::EventArgs^ e) {

textBoxValidationOnLeave(this->secondNum);

}

};

}

# 4 Результаты тестирования программы

В таблице 1 представлены наборы текстовых исходных данных, вводимых разными способами, и соответствующих результатов.

Таблица 1 – результаты тестирования программы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Операция | Исходные данные первой переменной | Исходные данные второй переменной | Вывод | Результат |
| Умножение | 0 | 56 | 0 | При умножении числа на нуль калькулятор правильно выполняет действие и выводит результат. |
| 0, 45 | 23 | 10, 35 | При умножении вещественного числа на целое калькулятор правильно выполняет действие и выводит результат |
| А | % | – | При попытке ввести не исключенные символы в первую или во вторую строку калькулятор не дает этого сделать |
| 10 | -3, 56 | -35,6 | При умножении числа на отрицательное число калькулятор правильно выполняет действие и выводит результат |
| Деление | 67 | 0 | – | При попытке разделить на нуль калькулятор выводит сообщение о запрете деления на нуль и оставляет пустой поле вывода |
| № | Й | – | При попытке ввести не исключенные символы в первую или во вторую строку калькулятор не дает этого сделать |
| 0, 98 | 0,007 | 140 | При делении вещественного числа на другое вещественное число калькулятор выполняет действие и выводит результат |
| -876 | 50 | -17,52 | При делении отрицательного числа на положительное калькулятор выполняет действие и выводит результат |
| Сложение | Ф | \* | – | При попытке ввести не исключенные символы в первую или во вторую строку калькулятор не дает этого сделать |
| 0, 58 | -89 | -88, 42 | При сложении вещественного числа и отрицательного числа калькулятор правильно выполняет действие и выводит результат |
| Вычитание | -67 | 34 | -101 | При вычитании одного числа из отрицательного числа калькулятор правильно выполняет действие и выводит результат |
| 0 | 56 | -56 | При вычитании числа из нуля калькулятор правильно выполняет действие и выводит результат |

Результаты работы программы при выборе деления на рисунке 7.

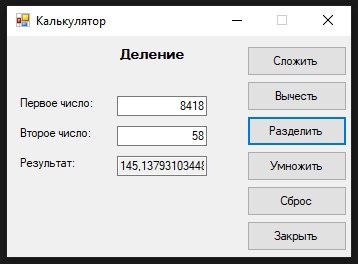


Рисунок 7 – Результат работы программы при выборе деления

Результаты работы программы при делении на нуль на рисунке 8

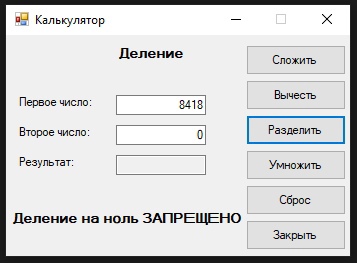


Рисунок 8 – Результат работы программы при делении на нуль

Результаты работы программы при делении вещественных чисел на рисунке 9.

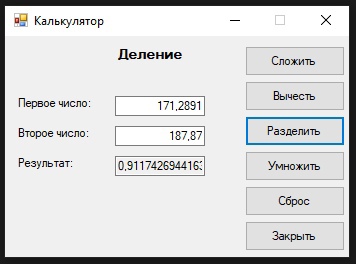


Рисунок 9– Результат работы программы при делении вещественных чисел

Результаты работы программы при умножении на нуль на рисунке 10.

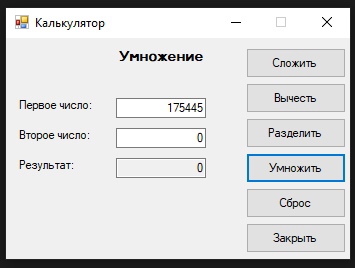


Рисунок 10 – Результат работы программы при умножении на нуль

Результаты работы программы при умножении отрицательных чисел на рисунке 11.

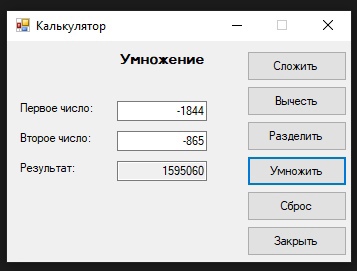


Рисунок 11 – Результат работы программы при умножении отрицательных чисел

# Выводы

Мы разработали программу Калькулятор с однострочным полем для отображения данных. И убедились, что разработанный калькулятор правильно выполняет все допустимые операции с различными исходными данными, включая «цепочки» операций.

Мы разработали калькулятор, который запрещает деление на нуль, выводя сообщение об этом. Также различает вещественные числа и выполняет с ними те же операции, что и с простыми. Кроме этого мы сделали так, что калькулятор не разрешает ввод букв или символов, не являющихся запятой или минусом. Убедились, что данные символы калькулятор не разрешает вводить по несколько раз. Вместе с этим калькулятор выводит сообщение о запрете действия, если поля ввода или одно из них пустые.