**мыФедеральное агентство связи**

**Государственное бюджетное образовательное учреждение высшего**

**образование**

**Ордена Трудового Красного Знамени**

**«Московский технический университет связи и информатики»**

Кафедра информатики

**Отчет по лабораторной работе №2**

«Разработка линейной программы “Простой калькулятор”»

по дисциплине «Информатика»

Подготовил студент

группы БВТ1903: Щитов В.М.

Проверил: Волков А.И.

Москва 2019

Оглавление

[1 Постановка задачи 3](#_Toc26811961)

[1.1 Общее задание на разработку программы 3](#_Toc26811962)

[1.2 Требования к элементам пользовательского интерфейса 4](#_Toc26811963)

[1.3 Требования к организации ввода-вывода данных в программе 9](#_Toc26811964)

[1.4 Требование к реакции программы на нестандартные, в том числе критические, действия пользователя 9](#_Toc26811965)

[1.5 Прочие требования к программе 10](#_Toc26811966)

[2 Схемы алгоритмов 11](#_Toc26811967)

[3 Код программы 17](#_Toc26811968)

[4 Результаты тестирования программы 21](#_Toc26811969)

[Вывод 25](#_Toc26811970)

# **1 Постановка задачи**

## **Общее задание на разработку программы**

Разработать на Visual C++ приложение, простой калькулятор. Условие задачи предполагает следующие пункты его реализации:

1) Создать форму, имеющую вид, представленный на рисунке 1, приблизительный результат работы которой также представлен на рисунке 1;

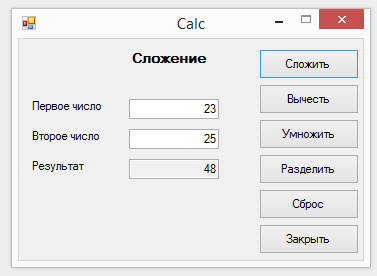


Рисунок 1 – Окно приложения «Простой калькулятор».

2) Реализовать обработчик события нажатия кнопок «Сложить», «Вычесть», «Умножить», «Разделить». Обработчик должен учитывать возможность отсутствия какой-либо информации, возможность ввода некорректных символов, реагировать на попытку деления на 0, правильно работать с действительными (вещественными) числами, предотвращать ошибки, связанные с этими событиями, и выводить уведомления об ошибке.

3) При нажатии на кнопку «Сброс» текстовые поля с вводимыми числами и результатом должны «очищаться», то есть становиться пустыми.

4) Нажатие кнопки «Закрыть» должно осуществлять корректное завершение работы приложения.

## **1.2 Требования к элементам пользовательского интерфейса**

Для данной программы необходимо на форме расположить 6 элементов типа Button, необходимых для арифметических вычислений, 3 элемента типа Label, описывающие информацию из соответствующего каждому из них TextBox , 3 элемента типа TextBox, двое из которых отвечают за ввод информации, а третий за вывод информации. Начнем с описания свойств разработанной формы, которые представлены в таблице 1.

Таблица 1 – Свойства кнопки и их значения

|  |  |  |
| --- | --- | --- |
| **Свойство** | **Описание** | **Значение** |
| Text | Заголовок формы | Calc |
| FormBorderStyle | Стиль обрамления  формы | FixedSingle (неизменяемая рамка, не допускающая изменения размеров окна) |
| MaximizeBox | Доступность кнопки «Развернутость» в заголовке окна | False (кнопка «Развернуть» не доступна) |
| Size.Width | Ширина формы | 360 |
| Size.Height | Высота формы | 260 |

Свойства элементов управления Button, в котором расположены такие элементы управления, как btnAdd, btnDeduct, btnMul, btnDiv, btnResult, btnClose отвечающие за арифметический операции, закрытие и сброс данных текстовых полей программы представлены в таблицах 2 - 4.

Таблица 2 – Свойства кнопки «Закрыть» и их значения

|  |  |  |
| --- | --- | --- |
| **Свойство** | **Описание** | **Значение** |
| (Name) | Имя кнопки | btnClose |
| Text | Заголовок кнопки | Закрыть |
| Location X | Расстояние от левого края | 240 |
| Location Y | Расстояние от верхнего края | 185 |
| Size.Width | Ширина кнопки | 100 |
| Size.Height | Высота кнопки | 30 |

Таблица 3 – Свойства кнопок «Сложить» и «Сброс» и их значения

|  |  |  |
| --- | --- | --- |
| **Свойство** | **Описание** | **Значение** |
| (Name) | Имя кнопки | btnAdd, btnReset |
| Text | Заголовок кнопки | Сложить, Сброс |
| Location X | Расстояние от левого края | 240 |
| Location Y | Расстояние от верхнего края | 10,150 |
| Size.Width | Ширина кнопки | 100 |
| Size.Height | Высота кнопки | 30 |

Таблица 4 – Свойства кнопок «Вычесть», «Умножить» и «Разделить» и их значения

|  |  |  |
| --- | --- | --- |
| **Свойство** | **Описание** | **Значение** |
| (Name) | Имя кнопки | btnDeduct, btnMul, btnDiv |
| Text | Заголовок кнопки | Вычесть, Умножить, Разделить |
| Location X | Расстояние от левого края | 240 |
| Location Y | Расстояние от верхнего края | 45,80,115 |
| Size.Width | Ширина кнопки | 100 |
| Size.Height | Высота кнопки | 30 |

Свойства элементов управления типа Label представлены в таблицах 5-7.

Таблица 5 – Свойства метки lblOper и их значения

|  |  |  |
| --- | --- | --- |
| **Свойство** | **Описание** | **Значение** |
| (Name) | Имя метки | lblOper |
| Text | Текст метки |  |
| TextAlign | Способ выравнивания текста | MiddleCenter |
| Location | Координаты (x; y) | 110; 10 |
| Font | Шрифт | Шрифт – Ms Sans Serif  Начертание – Жирный  Размер – 10 pt |

Таблица 6 – Свойства метки lbl1 и их значения

|  |  |  |
| --- | --- | --- |
| **Свойство** | **Описание** | **Значение** |
| (Name) | Название | lbl1 |
| Location | Координаты (x; y) | 10; 60 |
| TextAlign | Задает выравнивание текста в полях редактирования | MiddleLeft |
| Tеxt | Текст | Первое число |
| Font | Шрифт | Шрифт – Ms Sans Serif  Начертание – обычный  Размер – 8 pt |

Таблица 7 – Свойства lbl2 и их значения

|  |  |  |
| --- | --- | --- |
| **Свойство** | **Описание** | **Значение** |
| (Name) | Название | lbl2 |
| Location | Координаты (x; y) | 10; 90 |
| TextAlign | Задает выравнивание текста в полях редактирования | MiddleLeft |
| Text | Текст | Второе число |
| Font | Шрифт | Шрифт – Ms Sans Serif  Начертание – обычный  Размер – 8 pt |

Свойства метки lblResult предназначенной для обозначения первого поля ввода, представлены в таблице 8.

Таблица 8 – Свойства метки lblResult и их значения

|  |  |  |
| --- | --- | --- |
| **Свойство** | **Описание** | **Значение** |
| (Namе) | Название | lblResult |
| Location | Координаты (x; y) | 10; 120 |
| TextAlign | Задает выравнивание текста в полях редактирования | MiddleLeft |
| Text | Текст | Сумма |
| Font | Шрифт | Шрифт-Ms Sans Serif  Начертание – обычный  Размер – 8 pt |

Свойства текстового поля txt1, предназначенного для ввода первого числа, представлены в таблице 9.

Таблица 9 – Свойства txt1 и их значения

|  |  |  |
| --- | --- | --- |
| **Свойство** | **Описание** | **Значение** |
| (Namе) | Название | txt1 |
| Sizе | Размер: ширина и длина | 90; 22 |
| Location | Координаты (x; y) | 110; 60 |
| TabIndex | Порядок переключения TAB | 8 |
| TextAlign | Задает выравнивание текста в полях редактирования | Right |
| ReadOnly | Задает, может ли изменяться текст в поле редактировании | Falsе |

Свойства txt2, предназначенной для ввода второго числа, представлены в таблице 10.

Таблица 10 – Свойства txt1

|  |  |  |
| --- | --- | --- |
| **Свойство** | **Описание** | **Значение** |
| (Namе) | Название | txt2 |
| Sizе | Размер: ширина и длина | 90; 22 |
| Location | Координаты (x; y) | 110; 90 |
| TabIndex | Порядок переключения TAB | 9 |
| TextAlign | Задает выравнивание текста в полях редактирования | Right |
| RеadOnly | Задает, может ли изменяться текст в поле редактировании | Falsе |

Свойства txtResult, предназначенной для ввода второго числа, представлены в таблице 11.

Таблица 11 – Свойства txtResult

|  |  |  |
| --- | --- | --- |
| **Свойство** | **Описание** | **Значение** |
| (Namе) | Название | txtResult |
| Sizе | Размер: ширина и длина | 90; 22 |
| Location | Координаты (x; y) | 110; 120 |
| TabIndex | Порядок переключения TAB | 10 |
| TextAlign | Задает выравнивание текста в полях редактирования | Right |
| RеadOnly | Задает, может ли изменяться текст в поле редактировании | True |

## **1.3 Требования к организации ввода-вывода данных в программе**

Ввод информации осуществляется с клавиатуры, а вывод представлен на экране компьютера.

## **1.4 Требование к реакции программы на нестандартные, в том числе критические, действия пользователя**

При вводе некорректной информации пользователем, программа выдает уведомление об ошибке, в котором информирует пользователя о возможных вариантах вводимых, и блокирует его действия на нажатии не тех клавиш клавиатуры.

## **Прочие требования к программе**

Программа должна работать с вещественными числами, правильно реагировать на попытку деления на 0, корректно обрабатывать ошибки ввода данных. Нажатие кнопки «Закрыть» должно осуществлять корректное завершение работы приложения.

# **2 Схемы алгоритмов**

Схема алгоритма btnAdd\_Click представлена на рисунке 3.



Рисунок 3 – Схема алгоритма функции btnAdd\_Click.

Схема алгоритма btnDeduct\_Click представлена на рисунке 4.



Рисунок 4 – Схема алгоритма функции btnDeduct\_Click

Схема алгоритма btnReset\_Click представлена на рисунке 5.



Рисунок 5 – Схема алгоритма функции btnReset\_Click

Схема алгоритма btnMul\_Click представлена на рисунке 6.



Рисунок 6 – Схема алгоритма функции btnMul\_Click

Схема алгоритма btnClose\_Click представлена на рисунке 7.



Рисунок 7 – Схема алгоритма функции btnClose\_Click

Схема алгоритма btnDiv\_Click представлена на рисунке 8.



Рисунок 8 – Схема алгоритма функции btnDiv\_Click

Схема алгоритма TextChanged представлена на рисунке 9.



Рисунок 9 – Схема алгоритма функции TextChanged

Схема алгоритма TextChanged2 представлена на рисунке 10.



Рисунок 10 – Схема алгоритма функции TextChanged2

# **3** **Код программы**

Исходный код программы представлен в ниже приведённом листинге.

//Кнопка «Закрыть»

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

System::Windows::Forms::DialogResult result = MessageBox::Show(this, "Вы уверены, что хотите выйти?", "Выход", MessageBoxButtons::YesNo, MessageBoxIcon::Exclamation, MessageBoxDefaultButton::Button2);

if (result == System::Windows::Forms::DialogResult::Yes) {

this->Close();

}

}

//Кнопка «Сложить»

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

double i1, i2, i3;

this->txtResult->Text = "";

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

if (txt1->Text == "" || txt2->Text == "" || txt1->Text == "," || txt2->Text == "," || txt1->Text == "-" || txt2->Text == "-") {

System::Windows::Forms::DialogResult result = MessageBox::Show(this, "Введите числа!", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error, MessageBoxDefaultButton::Button1);

if (result == System::Windows::Forms::DialogResult::OK) {

this->txtResult->Text = "";

}

}

else {

this->lblOper->Text = L"Сложение";

i1 = Convert::ToDouble(txt1->Text);

i2 = Convert::ToDouble(txt2->Text);

i3 = i1 + i2;

this->txtResult->Text = Convert::ToString(i3);

}

}

//Кнопка «Сброс»

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

System::Windows::Forms::DialogResult result = MessageBox::Show(this, "Вы уверены, что хотите сбросить поля?", "Сброс", MessageBoxButtons::YesNo, MessageBoxIcon::Exclamation, MessageBoxDefaultButton::Button2);

if (result == System::Windows::Forms::DialogResult::Yes) {

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

this->txt1->Text = L"";

this->txt2->Text = L"";

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

}

}

//Кнопка «Вычесть»

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

double i1, i2, i3;

this->txtResult->Text = "";

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

if (txt1->Text == "" || txt2->Text == "" || txt1->Text == "," || txt2->Text == "," || txt1->Text == "-" || txt2->Text == "-") {

System::Windows::Forms::DialogResult result = MessageBox::Show(this, "Введите числа!", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error, MessageBoxDefaultButton::Button1);

if (result == System::Windows::Forms::DialogResult::OK) {

this->txtResult->Text = "";

}

}

else {

this->lblOper->Text = L"Вычитание";

i1 = Convert::ToDouble(txt1->Text);

i2 = Convert::ToDouble(txt2->Text);

i3 = i1 - i2;

this->txtResult->Text = Convert::ToString(i3);

}

}

//Кнопка «Умножить»

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

double i1, i2, i3;

this->txtResult->Text = "";

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

if (txt1->Text == "" || txt2->Text == "" || txt1->Text == "," || txt2->Text == "," || txt1->Text == "-" || txt2->Text == "-") {

System::Windows::Forms::DialogResult result = MessageBox::Show(this, "Введите числа!", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error, MessageBoxDefaultButton::Button1);

if (result == System::Windows::Forms::DialogResult::OK) {

this->txtResult->Text = "";

}

}

else {

this->lblOper->Text = L"Умножение";

i1 = Convert::ToDouble(txt1->Text);

i2 = Convert::ToDouble(txt2->Text);

i3 = i1 \* i2;

this->txtResult->Text = Convert::ToString(i3);

}

}

//Кнопка «Разделить»

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

double i1, i2, i3;

this->txtResult->Text = "";

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

if (txt1->Text == "" || txt2->Text == "" || txt1->Text == "," || txt2->Text == "," || txt1->Text == "-" || txt2->Text == "-") {

System::Windows::Forms::DialogResult result = MessageBox::Show(this, "Введите числа!", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error , MessageBoxDefaultButton::Button1);

if (result == System::Windows::Forms::DialogResult::OK) {

this->txtResult->Text = "";

}

}

else {

this->lblOper->Text = L"Деление";

i1 = Convert::ToDouble(txt1->Text);

i2 = Convert::ToDouble(txt2->Text);

if (i2 == 0) {

System::Windows::Forms::DialogResult result = MessageBox::Show(this, "На 0 нельзя делить!", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error, MessageBoxDefaultButton::Button1);

if (result == System::Windows::Forms::DialogResult::OK) {

this->txtResult->Text = "";

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

}

}

else {

i3 = i1 / i2;

this->txtResult->Text = Convert::ToString(i3);

}

}

}

int x1;

double x3;

//Обработчик ввода первого текстового поля

private: System::Void TextChanged(System::Object^ sender, System::Windows::Forms::KeyPressEventArgs^ e) {

if ((!Char::IsDigit(e->KeyChar)) && (e->KeyChar != '-' || txt1->Text->IndexOf('-') != -1) && (e->KeyChar != ',' || txt1->Text->IndexOf(',') != -1) && (e->KeyChar != (char)Keys::Back)) e->Handled = true;

if ((e->KeyChar == (char)Keys::D0 || e->KeyChar == (char)Keys::NumPad0) && (txt1->Text->Length >= 1) && (txt1->Text[0] == '0' && txt1->SelectionStart < 2)) e->Handled = true;

if (txt1->Text->IndexOf('-') == 0) {

if (txt1->SelectionStart == 0) {

e->Handled = true;

}

if ((txt1->SelectionStart == 2) && (e->KeyChar == '0')) {

e->Handled = true;

}

if ((txt1->SelectionStart == 1) && (e->KeyChar == ',')) {

e->Handled = true;

}

}

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

x1 = 0;

if (txt1->Text != "" && txt1->Text[0] == '-') x1 = 1;

if (txt1->Text->IndexOf(',') == -1 && txt1->SelectionStart == x1) {

txt1->Text = txt1->Text->Insert(x1, "0");

txt1->SelectionStart = x1 + 1;

}

}

if (e->KeyChar == '-' && txt1->Text->IndexOf('-') == -1) {

txt1->SelectionStart = 0;

}

}

//Обработчик ввода второго текстового поля

private: System::Void TextChanged2(System::Object^ sender, System::Windows::Forms::KeyPressEventArgs^ e) {

if ((!Char::IsDigit(e->KeyChar)) && (e->KeyChar != '-' || txt2->Text->IndexOf('-') != -1) && (e->KeyChar != ',' || txt2->Text->IndexOf(',') != -1) && (e->KeyChar != (char)Keys::Back)) e->Handled = true;

if ((e->KeyChar == (char)Keys::D0 || e->KeyChar == (char)Keys::NumPad0) && (txt2->Text->Length >= 1) && (txt2->Text[0] == '0' && txt2->SelectionStart < 2)) e->Handled = true;

if (txt2->Text->IndexOf('-') == 0) {

if (txt2->SelectionStart == 0) {

e->Handled = true;

}

if ((txt2->SelectionStart == 2) && (e->KeyChar == '0')) {

e->Handled = true;

}

if ((txt2->SelectionStart == 1) && (e->KeyChar == ',')) {

e->Handled = true;

}

}

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

x1 = 0;

if (txt2->Text != "" && txt2->Text[0] == '-') x1 = 1;

if (txt2->Text->IndexOf(',') == -1 && txt2->SelectionStart == x1) {

txt2->Text = txt2->Text->Insert(x1, "0");

txt2->SelectionStart = x1 + 1;

}

}

if (e->KeyChar == '-' && txt2->Text->IndexOf('-') == -1) {

txt2->SelectionStart = 0;

}

}

# **4 Результаты тестирования программы**

Скриншот разработанной программы представлен на рисунке 11.

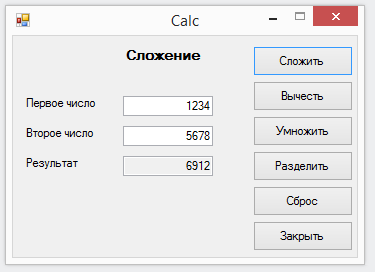


Рисунок 11 – Внешний вид разработанной программы

Чтобы удостовериться в корректности работы проведем тестирование программы при различных наборах данных, способных вызвать ошибку. Результаты тестирования программы приведены в таблице 12.

Таблица 12 – Результаты тестирования программы при некорректных данных.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Первое число | Второе число | Операция | Результат | Комментарий | Номер рисунка |
| - | - | Сложить | - | Проверка программы на корректную работу, в случае если пользователь не ввел оба числа | 12 |
| , | 122 | Сложить | - | Проверка программы на корректную работу, в случае если пользователь ввел только десятичный разделитель вместо первого числа | 13 |
| 122 | 0 | Разделить | - | Проверка программы на корректную работу при попытке пользователя разделить на 0 | 14 |
| 0,28 | 56 | Разделить | 0,005 | Проверка программы на корректную работу с вещественными числами | 15 |
| 25 | 25,005 | Сложение | 50,005 | Проверка программы на корректную работу с вещественными числами | 16 |
| 56 | 0,0001 | Вычитание | 55,9999 | Проверка программы на корректную работу с числами | 17 |
| 46,22 | 21,9 | Умножение | 1012,218 | Проверка программы на корректную работу с числами | 18 |

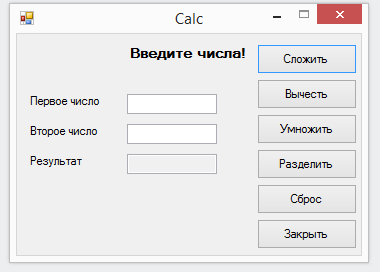


Рисунок 12 – Некорректный ввод двух чисел

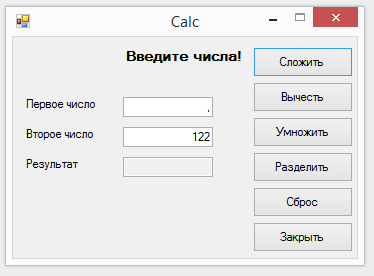


Рисунок 13 – Некорректный ввод первого числа

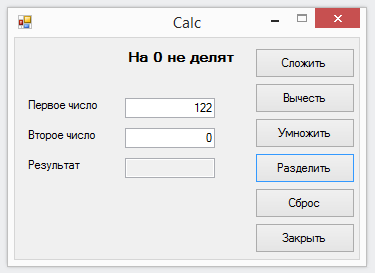


Рисунок 14 – Деление на ноль

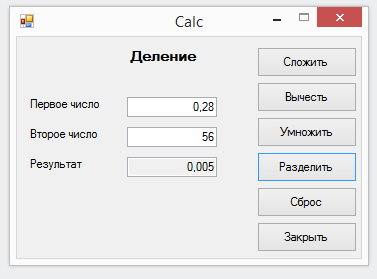


Рисунок 15 – Операция деления вещественных чисел

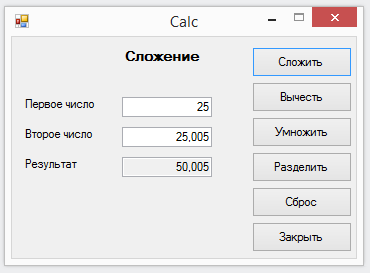


Рисунок 16 – Операция сложения вещественных чисел

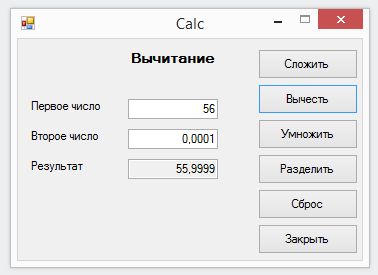


Рисунок 17 – Операция вычитания вещественных чисел

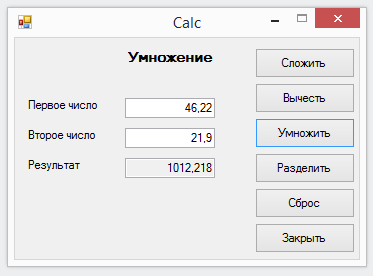


Рисунок 18 – Операция умножения вещественных чисел

# **Вывод**

В результате проделанной работы мы смогли создать простой калькулятор, а также научились предотвращать ошибки при вводе пользователем некорректных и критических данных.