**Федеральное агентство связи**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

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

Кафедра «Информатика»

**Отчет по Практической работе**

по дисциплине «Информатика»

на тему:

«Разработка линейной программы

«Простой калькулятор»»

Выполнил: студент группы

БВТ1905

Cемко Елена Константиновна

Руководитель:

Волков Андрей Иванович

Москва 2019

**Оглавление**

1 [Постановка Задачи 3](#_Toc534655555)

[2 Схемы алгоритмов 4](#_Toc534655556)

[3 Текст программы 10](#_Toc534655565)

[4 Результаты тестирования 14](#_Toc534655567)

Вывод…………………………………………………………………………….16

**1 Постановка Задачи**

В среде разработки Visual Studio требуется создать простой калькулятор. Возможности программы должны включать в себя операции сложения, вычитания, деления и умножения чисел. Кроме того, должен быть реализован функционал сброса всех введенных данных (используя кнопку сброса), закрытия программы (используя кнопку закрытия), а также отображение типа операции в текстовом окне. Программа должна правильно работать с действительными числами, реагировать на возможные ошибки ввода некорректных данных (выдавая соответствующие сообщения) и на попытку деления на ноль.

**2 Схемы алгоритмов**

Схема алгоритма для кнопки «Закрыть» представлена на рисунке 1

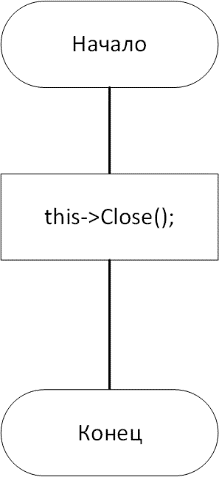
**

Рисунок 1 – Функция обработчика события нажатия на кнопку «Закрыть»

Схема алгоритма для кнопки «Сброс» представлена на рисунке 2

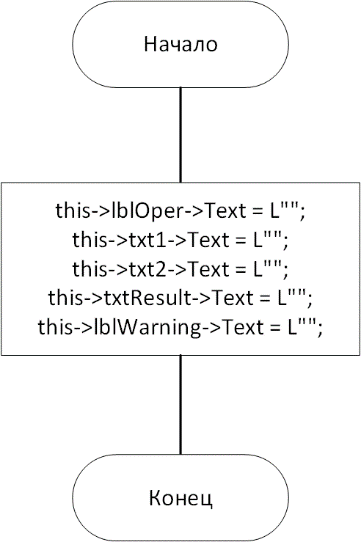
**

Рисунок 2 – Функция обработчика события нажатия на кнопку «Сброс»

Схема алгоритма для кнопки «Сложение» представлена на рисунке 3

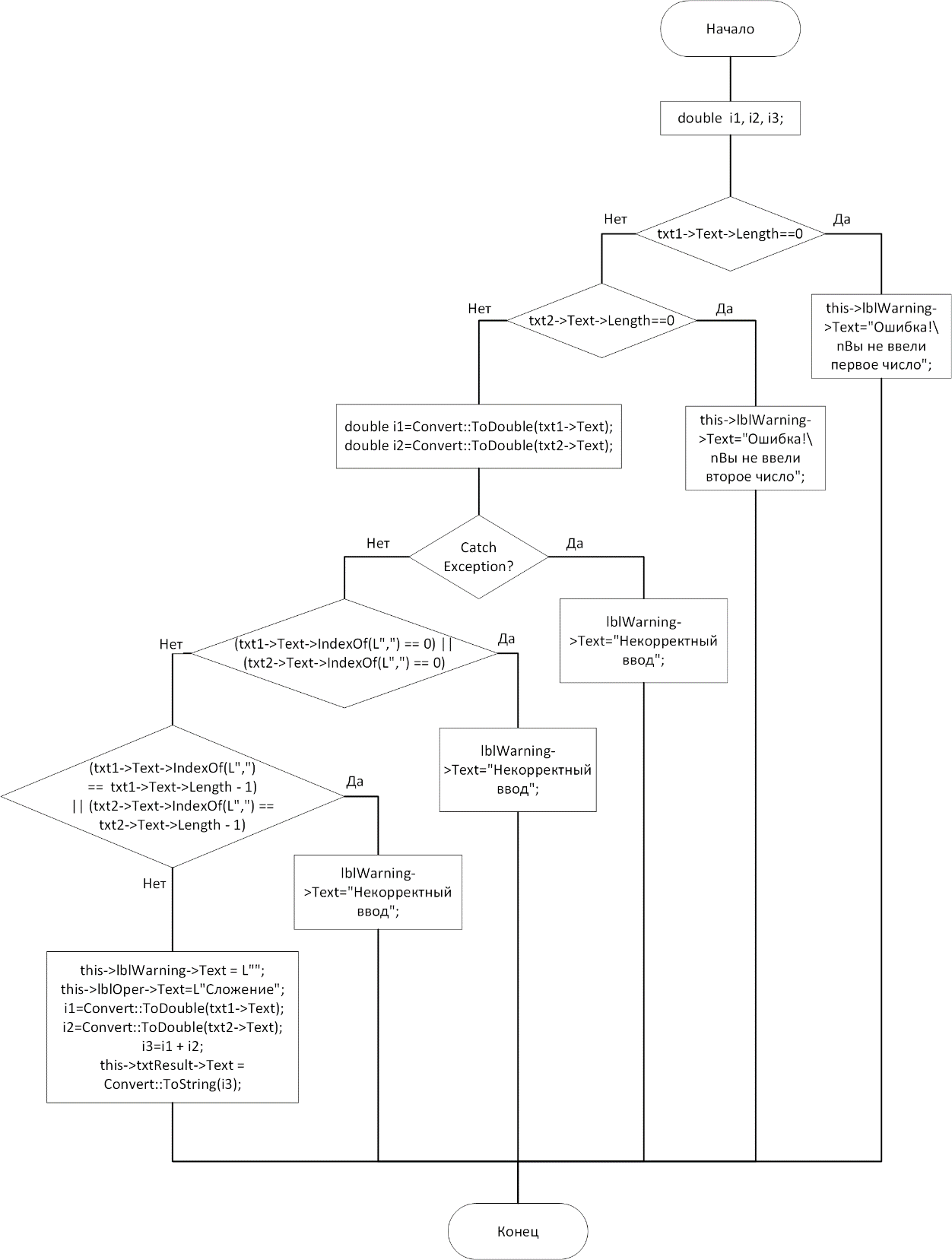


Рисунок 3 – Функция обработчика события нажатия на кнопку «Сложение»

Схема алгоритма для кнопки «Вычитание» представлена на рисунке 4

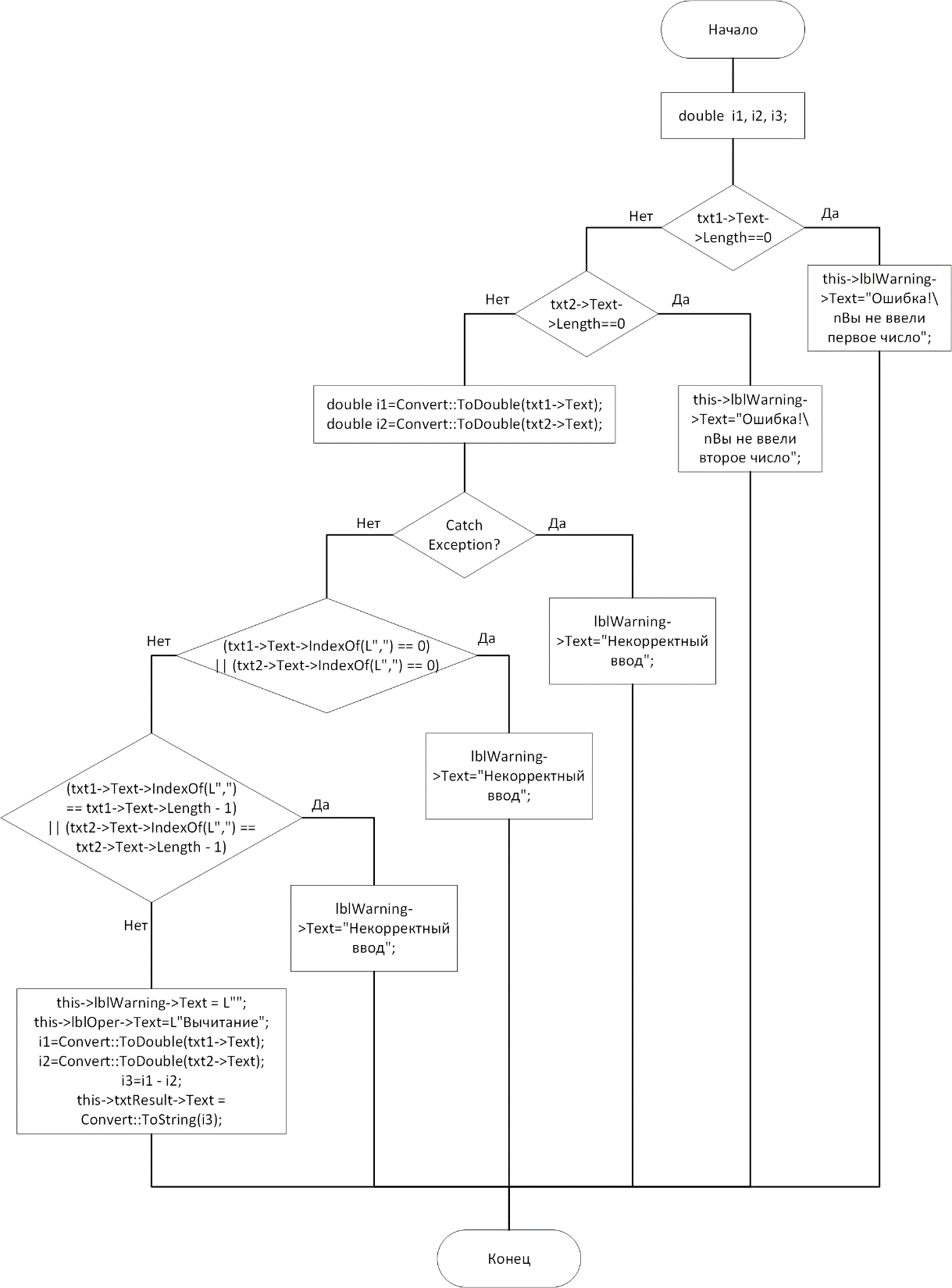


Рисунок 4 – Функция обработчика события нажатия на кнопку «Вычитание»

Схема алгоритма для кнопки «Умножение» представлена на рисунке 5

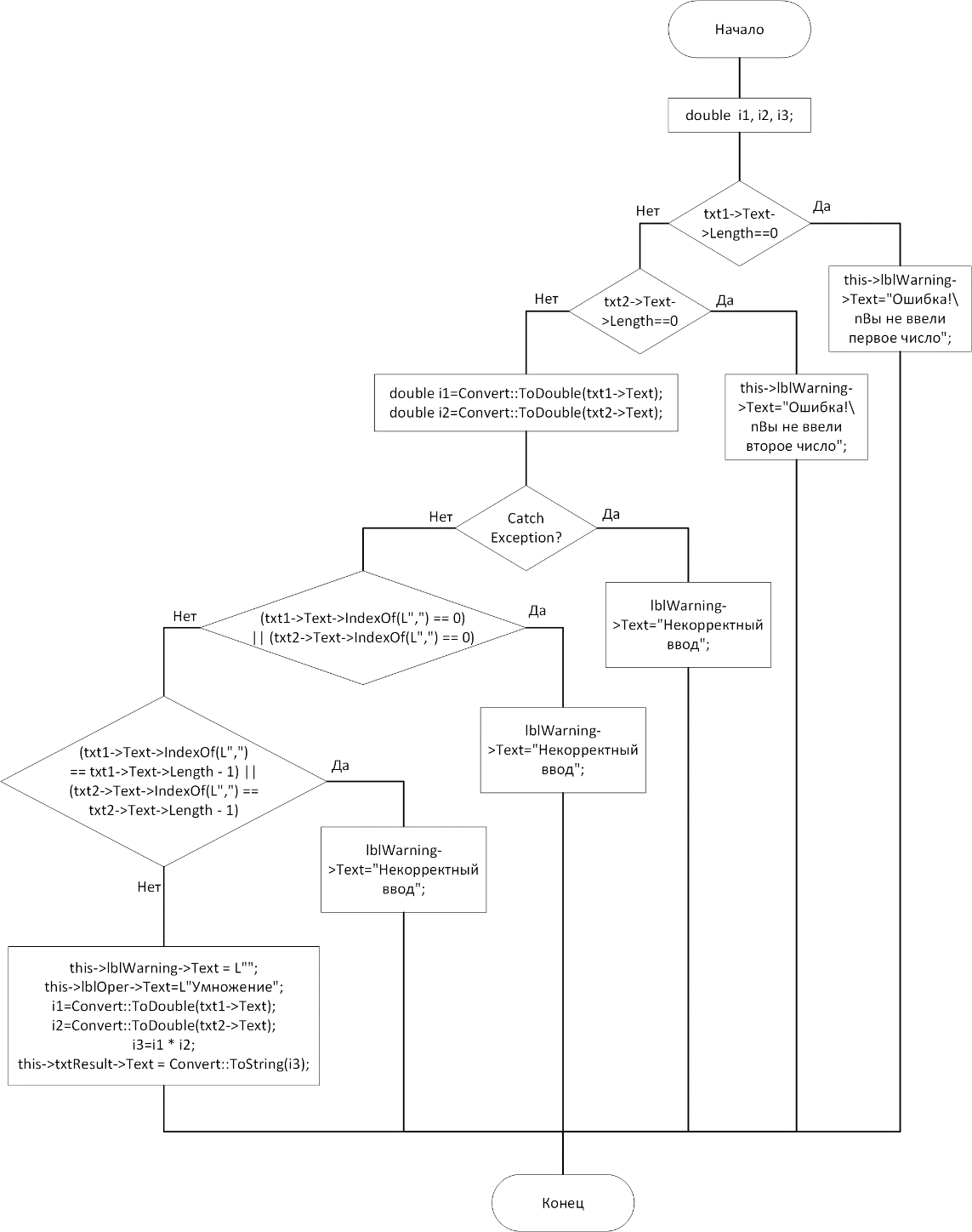


Рисунок 5 – Функция обработчика события нажатия на кнопку «Умножение»

Схема алгоритма для кнопки «Деление» представлена на рисунке 6

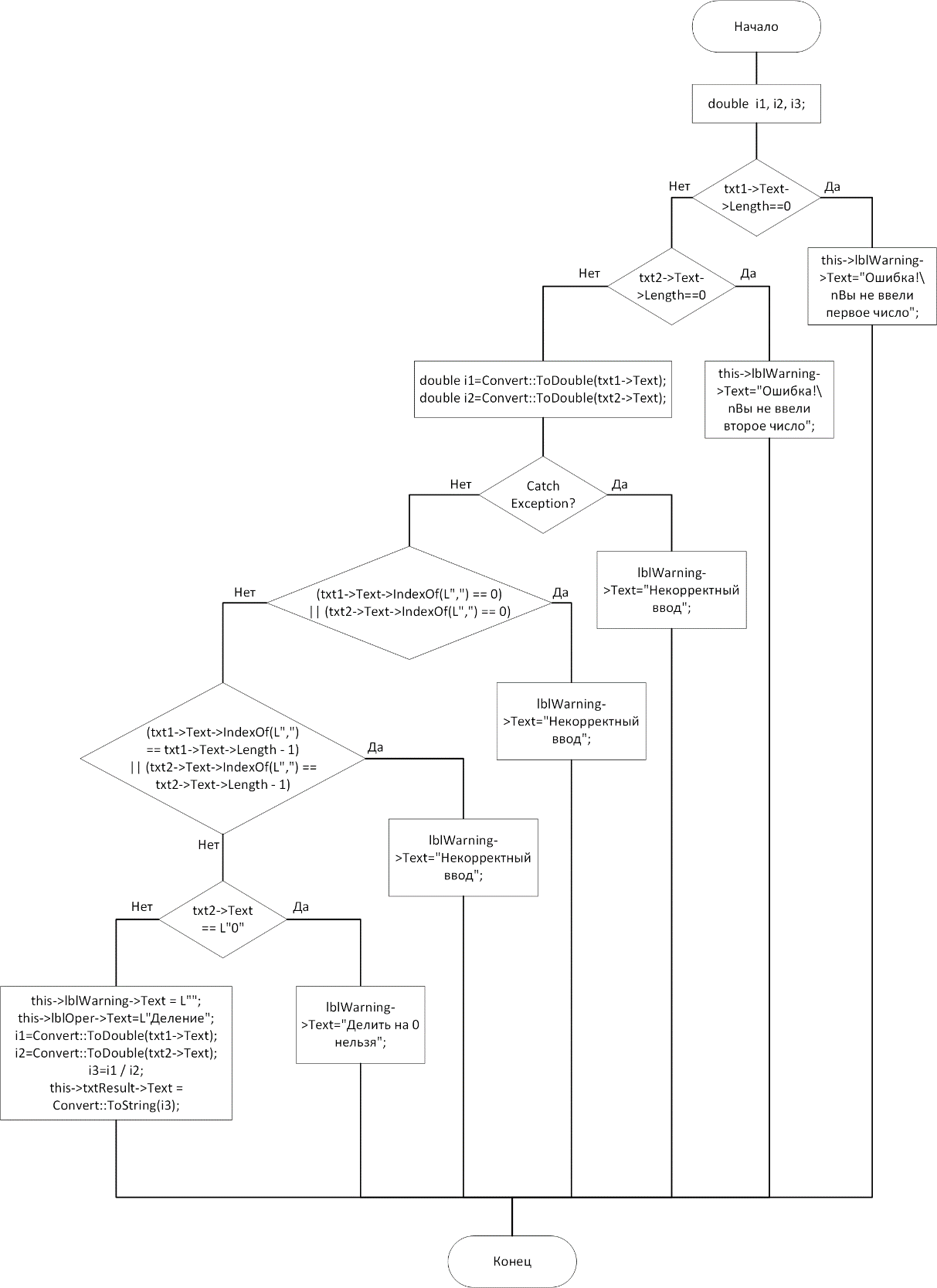


Рисунок 6 – Функция обработчика события нажатия на кнопку «Деление»

**3 Текст программы**

Текст программы представлен ниже.

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

**this**->Close();

}

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

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

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

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

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

**this**->lblWarning->Text = L"";

}

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

**double** i1, i2, i3;

**if** (txt1->Text->Length==**0**)

{**this**->lblWarning->Text=" Ошибка!\nВы не ввели первое число";

**return** ;}

**else** {**if** (txt2->Text->Length==**0**)

{**this**->lblWarning->Text="Ошибка!\nВы не ввели второе число";

**return** ;}};

try {

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

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

**catch**(...){lblWarning->Text="Некорректный ввод";

**return**;};

**if** ((txt1->Text->IndexOf(L",") == **0**) || (txt2->Text->IndexOf(L",") == **0**)) {lblWarning->Text="Некорректный ввод";

**return**;}

**if** (((txt1->Text->IndexOf(L",") == txt1->Text->Length - **1**))||(txt2->Text->IndexOf(L",") == txt2->Text->Length - **1**)) {lblWarning->Text="Некорректный ввод";

**return**;}

**this**->lblWarning->Text = L"";

**this**->lblOper->Text=L"Сложение";

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

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

i3=i1 + i2;

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

}

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

**double** i1, i2, i3;

**if** (txt1->Text->Length==**0**)

{**this**->lblWarning->Text=" Ошибка!\nВы не ввели первое число";

**return** ;}

**else** {**if** (txt2->Text->Length==**0**)

{**this**->lblWarning->Text="Ошибка!\nВы не ввели второе число";

**return** ;}};

try {

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

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

**catch**(...){lblWarning->Text="Некорректный ввод";

**return**;};

**if** ((txt1->Text->IndexOf(L",") == **0**) || (txt2->Text->IndexOf(L",") == **0**)) {lblWarning->Text="Некорректный ввод";

**return**;}

**if** (((txt1->Text->IndexOf(L",") == txt1->Text->Length - **1**))||(txt2->Text->IndexOf(L",") == txt2->Text->Length - **1**)) {lblWarning->Text="Некорректный ввод";

**return**;}

**this**->lblWarning->Text = L"";

**this**->lblOper->Text=L"Вычитание";

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

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

i3=i1 - i2;

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

}

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

**double** i1, i2, i3;

**if** (txt1->Text->Length==**0**)

{**this**->lblWarning->Text=" Ошибка!\nВы не ввели первое число";

**return** ;}

**else** {**if** (txt2->Text->Length==**0**)

{**this**->lblWarning->Text="Ошибка!\nВы не ввели второе число";

**return** ;}};

try {

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

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

**catch**(...){lblWarning->Text="Некорректный ввод";

**return**;};

**if** ((txt1->Text->IndexOf(L",") == **0**) || (txt2->Text->IndexOf(L",") == **0**)) {lblWarning->Text="Некорректный ввод";

**return**;}

**if** (((txt1->Text->IndexOf(L",") == txt1->Text->Length - **1**))||(txt2->Text->IndexOf(L",") == txt2->Text->Length - **1**)) {lblWarning->Text="Некорректный ввод";

**return**;}

**this**->lblWarning->Text = L"";

**this**->lblOper->Text=L"Умножение";

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

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

i3=i1 \* i2;

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

}

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

**double** i1, i2, i3;

**if** (txt1->Text->Length==**0**)

{**this**->lblWarning->Text=" Ошибка!\nВы не ввели первое число";

**return** ;}

**else** {**if** (txt2->Text->Length==**0**)

{**this**->lblWarning->Text="Ошибка!\nВы не ввели второе число";

**return** ;}};

try {

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

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

**catch**(...){lblWarning->Text="Некорректный ввод";

**return**;};

**if** ((txt1->Text->IndexOf(L",") == **0**) || (txt2->Text->IndexOf(L",") == **0**)) {lblWarning->Text="Некорректный ввод";

**return**;}

**if** (((txt1->Text->IndexOf(L",") == txt1->Text->Length - **1**))||(txt2->Text->IndexOf(L",") == txt2->Text->Length - **1**)) {lblWarning->Text="Некорректный ввод";

**return**;}

**if** (txt2->Text == L"0") {

lblWarning->Text="Делить на 0 нельзя";

**return**;

}

**this**->lblWarning->Text = L"";

**this**->lblOper->Text=L"Деление";

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

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

i3=i1 / i2;

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

}

**private**: System::Void Form1\_Load(System::Object^ sender, System::EventArgs^ e) {

}

**private**: System::Void txt1\_TextChanged(System::Object^ sender, System::EventArgs^ e) {

**int** i1 = txt1->Text->Length;

**if** (txt1->Text->IndexOf(L".") > -**1**) {

txt1->Text=txt1->Text->Replace(L".",L",");

txt1->SelectionStart = i1;

}

}

**private**: System::Void txt2\_TextChanged(System::Object^ sender, System::EventArgs^ e) {

**int** i1 = txt2->Text->Length;

**if** (txt2->Text->IndexOf(L".") > -**1**) {

txt2->Text=txt2->Text->Replace(L".",L",");

txt2->SelectionStart = i1;

}

}

**4 Результаты тестирования**

Результаты тестов представленные ниже на таблице 1

Таблица 1 - Тесты

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Первое число** | **Второе число** | **Результат** | **Операция** | **Комментарий** |
| 3 | 4 | 7 | Сложение |  |
| 8 | 2 | 6 | Вычитание |  |
| 2 | 0 |  | Деление | Делить на 0 нельзя. |
|  | 7 |  | Умножение | Ошибка! Вы не ввели первое число. |
| 0 | 0 | 0 | Умножение |  |
| 4,5 | 6 | 11,5 | Сложение |  |
| 3 | 1,5 | 2 | Деление |  |
| гг | вр |  | Сложение | Некорректный ввод |

На рисунке 7 представлена корректная работа программы при попытке пользователя разделить число на 0.

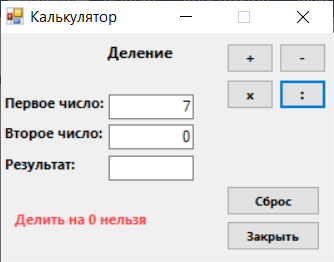


Рисунок 7 – Деление на ноль

На рисунке 8 представлена работа программы при попытке пользователя получить результат с пустым полем.

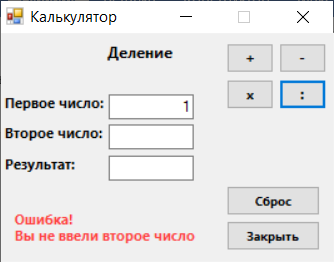


Рисунок 8 – Пустое поле

**Вывод**

В ходе данной лабораторной работы была создана программа модели простого калькулятора, которая умеет выполнять действия с вещественными числами, обрабатывать некорректный ввод данных, а также грамотно реагировать на попытку деления на ноль.