**Федеральное 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**

«Калькулятор»

по дисциплине «Информатика»

Подготовил студент

группы БВТ1903: Щитов В.М.

Проверил: Волков А.И.

Москва 2019

Оглавление

[1 Постановка задачи 3](#_Toc27929848)

[1.1 Общее задание на разработку программы 3](#_Toc27929849)

[1.2 Требования к элементам пользовательского интерфейса 4](#_Toc27929850)

[1.3 Требования к организации ввода-вывода данных в программе 7](#_Toc27929851)

[1.4 Требования к реакции программы на нестандартные, в том числе критические, действия пользователя 7](#_Toc27929852)

[1.5 Прочие требования к программе 7](#_Toc27929853)

[2 Схемы алгоритмов 8](#_Toc27929854)

[3 Код программы 13](#_Toc27929855)

[4 Результаты тестирования программы 16](#_Toc27929856)

[Вывод 18](#_Toc27929857)

# **1 Постановка задачи**

## Общее задание на разработку программы

Разработать на Visual C++ приложение «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;

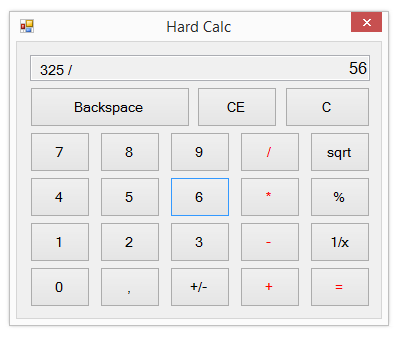


Рисунок 1 – Окно приложения «Калькулятор».

2) Реализовать введение числовых значений в поле textBox по нажатию на соответствующие кнопки на форме, а также десятичного разделителя. Реализовать корректное выполнение операций сложения, вычитания, умножения, деления, извлечения квадратного корня, вычисления заданного процента от числа, деления единицы на заданное число, а также изменения положительности/отрицательности числа.

3) Реализовать удаление последнего введённого символа из textBox по нажатию на кнопку Backspace, очистку значений обеих операнд по нажатию на кнопку C и очистку значения текущего операнда по нажатию на кнопку CE.

## 1.2 Требования к элементам пользовательского интерфейса

Для данной программы необходимо на форме расположить 23 элемента типа Button, выполняющих ввод данных в TextBox, арифметические операции и очистку формы, 1 элемент типа Label, содержащий в себе значение первого операнда , 1 элемент типа TextBox для ввода значения текущего операнда. Описание свойств формы представлено в таблице 1.

Таблица 1 – Свойства формы и её значения

|  |  |  |
| --- | --- | --- |
| **Свойство** | **Описание** | **Значение** |
| Text | Заголовок формы | Hard Calc |
| FormBorderStyle | Стиль обрамления  формы | FixedSingle (неизменяемый размер окна) |
| MaximizeBox | Доступность кнопки “Развернутость” в заголовке окна | False (кнопка ”Развернуть” недоступна) |
| Size.Width | Ширина формы | 380 |
| Size.Height | Высота формы | 315 |

Всего форма насчитывает 23 элемента управления типа Button, к которым относятся btnBackspace, btnCE, btnC, btn0, btn1, btn2, btn3, btn4, btn5, btn6, btn7, btn8, btn9, btnSeparator, btnPlusMinus, btnDiv, btnSqrt, btnMul, btnMod, btnDeduct, btnInverse, btnSum, btnResult. Они отвечают за ввод цифр, десятичного разделителя, арифметические операции, очистку полей и вывод результата в элемент textbox. Свойства этих элементов представлены в таблицах 2, 3, 4.

Таблица 2 – Свойства кнопок «Backspace», «CE» и «C» и их значения

|  |  |  |
| --- | --- | --- |
| **Свойство** | **Описание** | **Значение** |
| (Name) | Имя кнопки | btnBackspace, btnCE, btnC |
| Text | Заголовок кнопки | «Backspace», «CE», «С» |
| Location X | Расстояние от левого края | 13, 180, 268 |
| Location Y | Расстояние от верхнего края | 45 |
| Size.Width | Ширина кнопки | 160, 80, 85 |
| Size.Height | Высота кнопки | 40 |
| Font | Шрифт | Шрифт – Ms Sans Serif  Начертание – обычный  Размер – 10 pt |

Таблица 3 – Свойства кнопок чисел и арифметических выражений и их значения

|  |  |  |
| --- | --- | --- |
| **Свойство** | **Описание** | **Значение** |
| (Name) | Имя кнопки | btn7, btn8, btn9, btnDiv, btnSqrt, btn4, btn5, btn6, btnMul, btnMod, btn1, btn2, btn3, btnDeduct, btnInverse, btn0,btnSeparator, btnInverse, btnSum, btnResult |
| Text | Заголовок кнопки | «7», «8», «9», «/», «sqrt», «4», «5», «6», «\*», «%», «1», «2», «3», «-», «1/x», «0», «,»,  «+/-», «+», «=» |
| Location X | Расстояние от левого края | 13, 83, 153, 223, 293 |
| Location Y | Расстояние от верхнего края | 90, 135, 180, 225 |
| Size.Width | Ширина кнопки | 60 |
| Size.Height | Высота кнопки | 40 |
| Font | Шрифт | Шрифт – Ms Sans Serif  Начертание – обычный  Размер – 10 pt |

Свойства элемента управления типа Label, к которому относится lblTemp, отвечающий за хранение в поле Text значения первого операнда, представлены в таблице 4.

Таблица 4 – Свойства метки lblTemp и их значения

|  |  |  |
| --- | --- | --- |
| **Свойство** | **Описание** | **Значение** |
| (Name) | Имя метки | lblTemp |
| Text | Текст метки |  |
| TextAlign | Способ выравнивания текста | TopLeft |
| Location X | Расстояние от левого края | 20 |
| Location Y | Расстояние от верхнего края | 19 |
| Font | Шрифт | Шрифт – Ms Sans Serif  Начертание – обычный  Размер – 10 pt |

Свойства элемента управления типа TextBox, к которому относится txtBox, предназначенный для ввода значения текущего операнда и вывода результата вычислений, представлены в таблице 5.

Таблица 5 – Свойства текстового поля txtBox

|  |  |  |
| --- | --- | --- |
| **Свойство** | **Описание** | **Значение** |
| (Namе) | Название | txtBox |
| Sizе | Размер: длина и высота | 340;26 |
| Location | Координаты (x; y) | 13; 13 |
| TextAlign | Задает выравнивание текста в полях редактирования | Right |
| Font | Шрифт | Шрифт – Ms Sans Serif  Начертание – обычный  Размер – 12 pt |
| RеadOnly | Задает, может ли вводиться значение с клавиатуры | True |

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

Ввод информации осуществляется посредством нажатия на кнопки на форме приложения, а вывод представлен на экране компьютера.

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

Должен быть реализован запрет на ввод любых данных с клавиатуры.

## **Прочие требования к программе**

Программа должна работать с вещественными числами.

# **2 Схемы алгоритмов**

На рисунке 2 изображена блок-схема события txtBox\_TextChanged, отвечающего за изменение текстового поля



Рисунок 2 – Схема алгоритма функции txtBox\_TextChanged

На рисунке 3 изображена блок-схема события btnBackspace\_Click, отвечающего за удаление последнего введённого символа



Рисунок 3 – Схема алгоритма функции btnBackspace\_Click

На рисунке 4 изображена блок-схема события btnCE\_Click, отвечающего за очистку текстового поля txtBox.



Рисунок 4 – Схема алгоритма функции btnCE\_Click

На рисунке 5 изображена блок-схема события btnC\_Click, отвечающего за очистку текстового поля txtBox и метки lblTemp.



Рисунок 5 – Схема алгоритма функции btnC\_Click

На рисунке 6 изображена блок-схема события btnSeparator\_Click, отвечающего за ввод запятой в текстовое поле txtBox.



Рисунок 6 – Схема алгоритма функции btnSeparator\_Click

На рисунке 7 изображена блок-схема события btnPlusMinus\_Click, отвечающего за изменение знака введённого числа.



Рисунок 7 – Схема алгоритма функции btnPlusMinus\_Click

На рисунке 8 изображена блок-схема события btnNum\_Click, отвечающего за ввод цифр в текстовое поле txtBox.



Рисунок 8 – Схема алгоритма функции btnNum\_Click

На рисунке 9 изображена блок-схема события btnMath\_Click, отвечающего за ввод первого числа и выбор арифметического действия.



Рисунок 9 – Схема алгоритма функции btnMath\_Click

На рисунке 10 изображена блок-схема события btnResult\_Click, отвечающего за арифметические расчёты и вывод результатов расчётов.



Рисунок 10 – Схема алгоритма функции btnResult\_Click

# **3 Код программы**

Исходный код программы представлен в ниже приведённом листинге.

double firstnum, secondnum, result;

String^ temp;

//Обработчик изменения текстового поля

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

if (txtBox->Text == "") {

txtBox->Text = "0";

}

}

//Обработчик нажатия на кнопку «Backspace»

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

if (txtBox->Text->Length >0) {

txtBox->Text = txtBox->Text->Remove(txtBox->Text->Length -1 ,1);

}

}

//Обработчик нажатия на кнопку «С»

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

txtBox->Text = "0";

lblTemp->Text = "0";

}

//Обработчик нажатия на кнопку «СЕ»

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

txtBox->Text = "0";

}

//Обработчик нажатия на кнопку «,»

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

if (!txtBox->Text->Contains(",")) {

txtBox->Text = txtBox->Text + ",";

}

}

//Обработчик нажатия на кнопку «+/-»

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

if (txtBox->Text->Contains("-")){

txtBox->Text = txtBox->Text-> Remove(0,1);

}

else {

txtBox->Text = "-" + txtBox->Text;

}

}

//Обработчик нажатия на кнопку «=»

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

secondnum= Double::Parse(txtBox->Text);

lblTemp->Text = L"";

if (temp == "+") {

result = firstnum + secondnum;

txtBox->Text = System::Convert::ToString(result);

}

else if (temp == "-") {

result = firstnum - secondnum;

txtBox->Text = System::Convert::ToString(result);

}

else if (temp == "\*") {

result = firstnum \* secondnum;

txtBox->Text = System::Convert::ToString(result);

}

else if (temp == "/") {

result = firstnum / secondnum;

txtBox->Text = System::Convert::ToString(result);

}

else if (temp == "%") {

result = firstnum \*0.01\* secondnum;

txtBox->Text = System::Convert::ToString(result);

}

}

//Обработчик нажатия кнопок с цифрами

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

Button ^ Numbers = safe\_cast<Button^>(sender);

if (txtBox->Text =="0") {

txtBox->Text = Numbers->Text;

}

else txtBox->Text = txtBox->Text + Numbers->Text;

}

//Обработчик нажатия одной из кнопок с математическими действиями

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

Button ^ op = safe\_cast<Button^>(sender);

firstnum = Double::Parse(txtBox->Text);

temp = op->Text;

if (temp == "sqrt") {

result = sqrt(firstnum);

txtBox->Text = System::Convert::ToString(result);

lblTemp->Text = L"";

}

else if (temp == "1/x") {

result = 1 / firstnum;

txtBox->Text = System::Convert::ToString(result);

lblTemp->Text = L"";

}

else {

txtBox->Text = "";

lblTemp->Text = System::Convert::ToString(firstnum) + " " + temp;

}

}

# **4 Результаты тестирования программы**

На рисунке 11 представлен интерфейс рабочей программы.

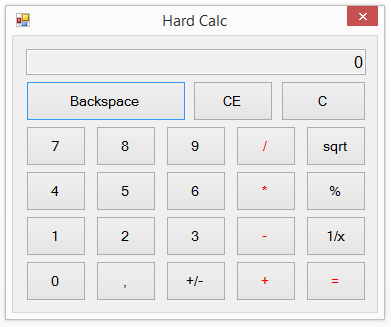


Рисунок 11 – Интерфейс программы

На рисунке 12 представлен результат сложения двух вещественных чисел.

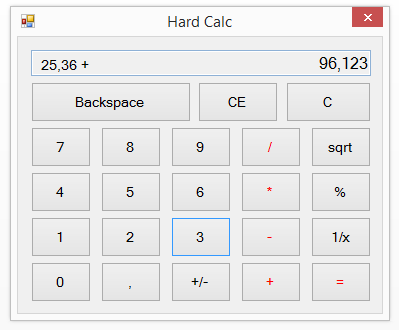
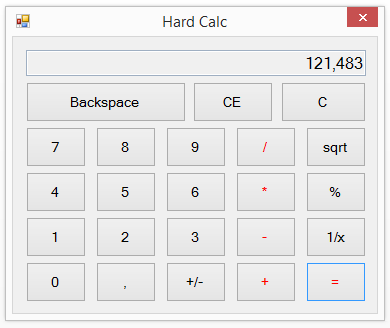
 

Рисунок 12 – Результат сложения двух вещественных чисел

На рисунке 13 представлен результат вычисления квадратного корня программой.

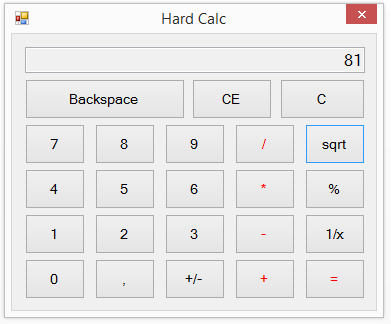
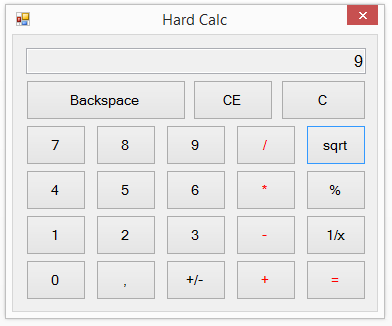
 

Рисунок 13 – Результат вычисления квадратного корня программой

На рисунке 14 представлен результат работы программы над вычислением заданного процента от числа.

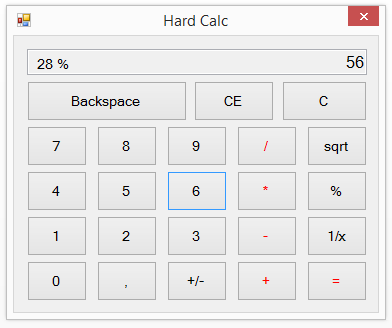
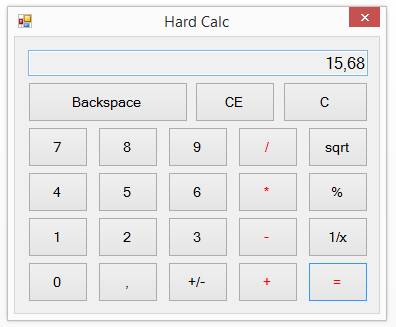
 

Рисунок 14 – Результат вычисления заданного процента от числа

## **Вывод**

В ходе данной лабораторной работы были получены практические знания, необходимые для работы с различными элементами Visual C++, необходимыми для создания аналога калькулятора, встроенного в Windows, со всеми необходимыми функциями.