**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 №3**

«Разработка программы с разветвлениями

“Решение квадратного уравнения”»

по дисциплине «Информатика»

Подготовил студент

группы БВТ1903: Щитов В.М.

Проверил: Волков А.И.

Москва 2019

Оглавление

[1 Постановка задачи 3](#_Toc27957749)

[1.1 Общее задание на разработку программы 3](#_Toc27957750)

[1.2 Требования к элементам пользовательского интерфейса 4](#_Toc27957751)

[1.3 Требования к организации ввода-вывода данных в программе 6](#_Toc27957752)

[1.4 Требования к реакции программы на нестандартные, в том числе критические, действия пользователя 7](#_Toc27957753)

[1.5 Прочие требования к программе 7](#_Toc27957754)

[2 Схемы алгоритмов 8](#_Toc27957755)

[3 Код программы 14](#_Toc27957756)

[4 Результаты тестирования программы 18](#_Toc27957757)

[Вывод 21](#_Toc27957758)

# **1 Постановка задачи**

## **Общее задание на разработку программы**

Разработать на Visual C++ приложение «Решение квадратного уравнения». 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.

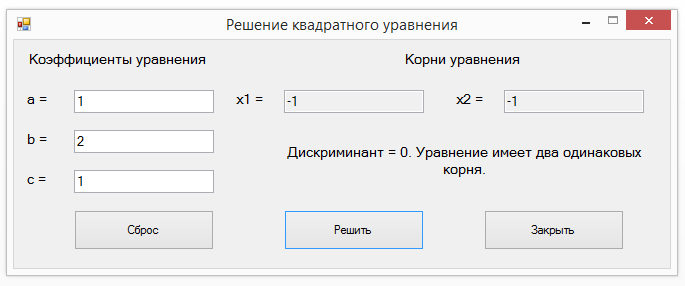


Рисунок 1 – Окно приложения «Решение квадратного уравнения».

2) Реализовать введение числовых значений в поля типа TextBox, . ограничить ввод букв и специальных символов, при этом должны корректно вводиться знак минуса и десятичного разделителя, чтобы могли вводиться вещественные и отрицательные числа.

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 3 элемента типа Button, выполняющих очистку полей, решение уравнения и закрытие программы, 8 элементов типа Label, 7 из которых поясняют содержание текстовых полей TextBox для ввода данных и найденных корней, 1 предназначен для уведомления пользователя в случаях нахождения корней, их отсутствия или неверно введённых данных, 5 элементов типа TextBox, 3 из которых предназначены для ввода значений коэффициентов, а 2 для вывода значений корней. Описание свойств формы представлено в таблице 1.

Таблица 1 – Свойства формы и её значения

|  |  |  |
| --- | --- | --- |
| **Свойство** | **Описание** | **Значение** |
| Text | Заголовок формы | Решение квадратного уравнения |
| FormBorderStyle | Стиль обрамления  формы | FixedDialog (неизменяемый размер окна) |
| MaximizeBox | Доступность кнопки “Развернутость” в заголовке окна | False (кнопка ”Развернуть” недоступна) |
| Size.Width | Ширина формы | 680 |
| Size.Height | Высота формы | 270 |

Свойства элементов управления типа Button, к которым относятся btnRes, btnAction, btnCls, отвечающие за очистку полей ввода-вывода, решение уравнения и закрытие приложения представлены в таблице 2.

Таблица 2 – Свойства кнопок и их значения

|  |  |  |
| --- | --- | --- |
| **Свойство** | **Описание** | **Значение** |
| (Name) | Имя кнопки | btnRes, btnAction, btnCls |
| Text | Заголовок кнопки | «Сброс», «Решить», «Закрыть» |
| Location X | Расстояние от левого края | 60, 270, 470 |
| Location Y | Расстояние от верхнего края | 170 |
| Size.Width | Ширина кнопки | 140 |
| Size.Height | Высота кнопки | 40 |

Свойства элементов управления типа Label, к которым относятся label1, label2, label3, label4, label5, label6, label7, lblRes, предназначенные для пояснения назначения элементов типа TextBox, представлены в таблице 3.

Таблица 3 – Свойства меток типа Label и их значения

|  |  |  |
| --- | --- | --- |
| **Свойство** | **Описание** | **Значение** |
| (Name) | Имя метки | label1, label2, label3, label4, label5, label6, label7, lblRes |
| Text | Текст метки | «Коэффициенты уравнения», «а=», «b=», «c=», «x1=», «x2=», «Корни уравнения», «» |
| TextAlign | Способ выравнивания текста | TopLeft |
| Location X | Расстояние от левого края | 10, 220, 440, 270 |
| Location Y | Расстояние от верхнего края | 10, 50, 90, 130 |
| Font | Шрифт | Шрифт – Ms Sans Serif  Начертание – обычный  Размер – 10 pt. |

Свойства элементов управления типа TextBox, к которому относятся txt1, txt2, txt3, txtRes1, txtRes2, предназначенные для ввода значений коэффициентов и вывода значений корней, представлены в таблице 4.

Таблица 4 – Свойства элементов типа TextBox и их значения

|  |  |  |
| --- | --- | --- |
| **Свойство** | **Описание** | **Значение** |
| (Namе) | Название | txt1, txt2, txt3, txtRes1, txtRes2 |
| Size | Размер: длина и высота | 140; 23 |
| Location | Координаты (x; y) | (60; 50, 90, 130)  (270, 490; 50) |
| TextAlign | Задает выравнивание текста в полях редактирования | Right |
| ReadOnly | Задает, может ли вводиться значение с клавиатуры | False, True |

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

Ввод данных должен осуществляться с клавиатуры, вывод должен производиться на экран компьютера.

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

Недопустим ввод любых данных кроме числовых, десятичного разделителя и знака минус. Для любых значений коэффициентов (в том числе и нулевых) выводимые комментарии должны быть математически корректными. В созданном приложении должны корректно обрабатываться возможные ошибки ввода исходных данных.

## **Прочие требования к программе**

Программа должна работать с вещественными числами, корректно реагировать на неверное введение требуемых для расчётов коэффициентов пользователем. Нажатие на кнопку «Закрыть» должно корректно завершать работу программы.

# **2** **Схемы алгоритмов**

На рисунке 2 изображена блок-схема события btnReset, отвечающего за очистку текстовых полей программы.



Рисунок 2 – Схема алгоритма функции btnReset

На рисунках 3, 4 и 5 изображены блок-схемы событий txt1\_TextChanged, txt2\_TextChanged и txt3\_TextChanged соответственно, отвечающих за ввод корректных символов в текстовые поля коэффициентов уравнений.



Рисунок 3 – Схема алгоритма функции txt1\_TextChanged



Рисунок 4 – Схема алгоритма функции txt2\_TextChanged



Рисунок 5 – Схема алгоритма функции txt3\_TextChanged

На рисунке 6 изображена блок-схема события btnClose, отвечающего за корректное закрытие приложения.



Рисунок 6 – Схема алгоритма функции btnClose

На рисунке 7 изображена блок-схема события btnAction\_Click, отвечающего квадратного уравнения или вывод корректного математического ответа.



Рисунок 7 – Схема алгоритма функции btnAction\_Click

# **3 Код программы**

Исходный код программы представлен в ниже приведённом листинге.

private:

int x1;

double x3;

//Обработчик ввода текста для первого текстового поля (а)

private: System::Void txt1\_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 текста для первого текстового поля (b)

private: System::Void txt2\_TextChanged(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;

}

}

//Обработчик ввода текста для первого текстового поля (с)

private: System::Void txt3\_TextChanged(System::Object^ sender, System::Windows::Forms::KeyPressEventArgs^ e) {

if ((!Char::IsDigit(e->KeyChar)) && (e->KeyChar != '-' || txt3->Text->IndexOf('-') != -1) && (e->KeyChar != ',' || txt3->Text->IndexOf(',') != -1) && (e->KeyChar != (char)Keys::Back)) e->Handled = true;

if ((e->KeyChar == (char)Keys::D0 || e->KeyChar == (char)Keys::NumPad0) && (txt3->Text->Length >= 1) && (txt3->Text[0] == '0' && txt3->SelectionStart < 2)) e->Handled = true;

if (txt3->Text->IndexOf('-') == 0) {

if (txt3->SelectionStart == 0) {

e->Handled = true;

}

if ((txt3->SelectionStart == 2) && (e->KeyChar == '0')) {

e->Handled = true;

}

if ((txt3->SelectionStart == 1) && (e->KeyChar == ',')) {

e->Handled = true;

}

}

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

x1 = 0;

if (txt3->Text != "" && txt3->Text[0] == '-') x1 = 1;

if (txt3->Text->IndexOf(',') == -1 && txt3->SelectionStart == x1) {

txt3->Text = txt3->Text->Insert(x1, "0");

txt3->SelectionStart = x1 + 1;

}

}

if (e->KeyChar == '-' && txt3->Text->IndexOf('-') == -1) {

txt3->SelectionStart = 0;

}

}

//Обработчик нажатия кнопки «Сбросить»

private: System::Void btnReset(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->lblRes->Text = L"";

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

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

this->txt3->Text = L"";

this->txtRes1->Text = L"";

this->txtRes2->Text = L"";

}

}

//Обработчик нажатия кнопки «Закрыть»

private: System::Void btnClose(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();

}

}

//Обработчик нажатия кнопки «Решить»

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

if (txt1->Text == "" || txt2->Text == "" || txt3->Text == "" || txt1->Text == "," || txt2->Text == "," || txt3->Text == "," || txt1->Text == "-" || txt2->Text == "-" || txt3->Text == "-") {

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

}

else {

double a, b, c, D, x1, x2;

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

this->txtRes1->Text = "";

this->txtRes2->Text = "";

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

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

c = Convert::ToDouble(txt3->Text);

D = (b\*b) - (4 \* a\*c);

if (a == 0 && b == 0 && c != 0) {

this->lblRes->Text = L"Уравнение не имеет действительных корней.";

}

else if ((a == 0 || a == -0) && (b == 0 || b == -0) && (c == 0 || c == -0)) {

this->lblRes->Text = L"Выражение тождественно (0 = 0).";

}

else if (a == 0) {

x1 = (-c) / b;

this->txtRes1->Text = Convert::ToString(x1);

this->lblRes->Text = L"Линейное уравнение – имеет один корень.";

}

else if (D < 0) {

this->lblRes->Text = L"Дискриминант < 0. Уравнение не имеет действительных корней.";

}

else if (D == 0) {

x1 = -b / (2 \* a);

x2 = x1;

this->txtRes1->Text = Convert::ToString(x1);

this->txtRes2->Text = Convert::ToString(x2);

this->lblRes->Text = L"Дискриминант = 0. Уравнение имеет два одинаковых корня.";

}

else if (D > 0) {

D = sqrt(D);

x1 = (-b + D) / (2 \* a);

x2 = (-b - D) / (2 \* a);

this->txtRes1->Text = Convert::ToString(x1);

this->txtRes2->Text = Convert::ToString(x2);

this->lblRes->Text = L"Дискриминант > 0. Уравнение имеет два действительных корня.";

}

}

}

# **4 Результаты тестирования программы**

На рисунке 8 представлен интерфейс рабочей программы.

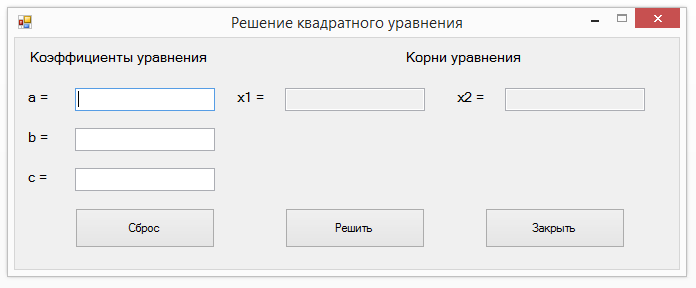


Рисунок 8 – Интерфейс программы

В таблице 5 представлены результаты тестирования программы.

Таблица 5 –

|  |  |
| --- | --- |
| **Введенные значения** | **Результат** |
| 1)А = 1; В = 2; С = 6 | Дискриминант < 0. Уравнение не имеет действительных корней. |
| 2)А = 1; В = 2; С = -3 | Х1= 1; Х2= -3  Дискриминант > 0. Уравнение имеет два корня. |
| 3)А = 0; В = -3; С = 3 | Х1= 1  Линейное уравнение – имеет корень. |
| 4)А = 1; В = 2; С = 1 | Х1= -1; Х2= -1  Дискриминант = 0. Уравнение имеет два одинаковых корня |
| 5) A = 0; B = 0; C = 0 | Выражение тождественно (0 = 0) |

На рисунке 9 представлена корректная работа программы при D<0.

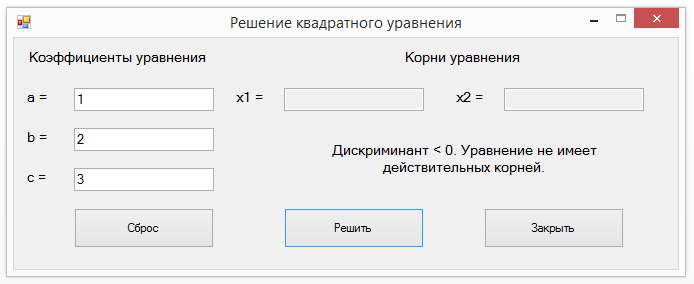


Рисунок 9 – Корректная работа программы при дискриминанте меньше 0

На рисунке 10 представлена корректная работа программы при D>0.

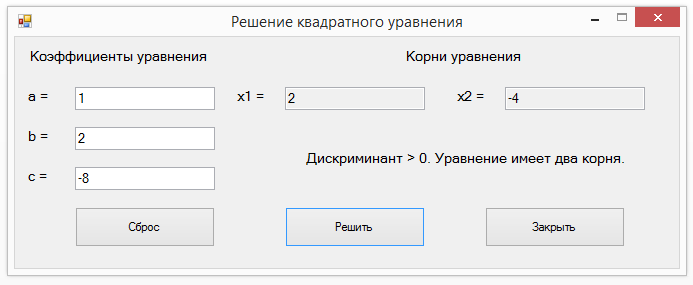


Рисунок 10 – Работа программы при дискриминанте больше 0

На рисунке 11 представлена корректная работа программы при D=0.

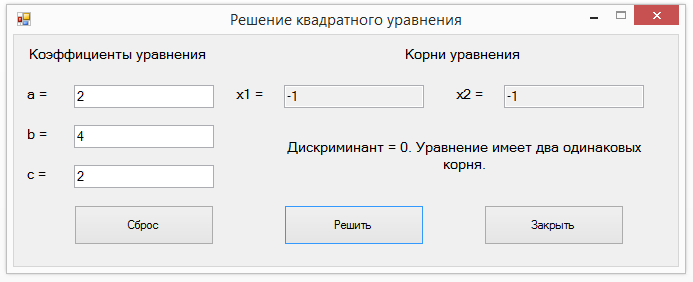


Рисунок 11 – Работа программы при дискриминанте равном 0

## **Вывод**

В ходе данной лабораторной работы были получены практические знания, связанные с алгоритмами ветвления, которые потребовались для создания простейшей программы по вычислению корней квадратного уравнения.