# Содержание

[Содержание 3](#_Toc166891952)

[Введение 4](#_Toc166891953)

[1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 6](#_Toc166891954)

[1.1 Описание программируемой системы 6](#_Toc166891955)

[1.2 Обзор существующих решений 8](#_Toc166891956)

[1.3 Требования к программируемой системе 11](#_Toc166891957)

[2 ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ 13](#_Toc166891958)

[2.1 Диаграмма состояний системы 13](#_Toc166891959)

[2.2 Диаграмма классов системы 14](#_Toc166891960)

[2.3 Диаграмма последовательности системы 14](#_Toc166891961)

[3 ПРАКТИЧЕСКАЯ ЧАСТЬ 15](#_Toc166891962)

[3.1 Реализация требований к системе 15](#_Toc166891963)

[3.2 Функциональное тестирование программного продукта 21](#_Toc166891964)

[3.3 Инструкция по эксплуатации 23](#_Toc166891965)

[ЗАКЛЮЧЕНИЕ 24](#_Toc166891966)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 25](#_Toc166891967)

[ПРИЛОЖЕНИЯ 26](#_Toc166891968)

[Приложение А – Исходный код программы 26](#_Toc166891969)

# Введение

В настоящее время рынок разработки программного обеспечения требует от специалистов не только владения узкой областью знаний, но и знаний во множестве связанных технологических областей. Данное исследование посвящено созданию современного программного обеспечения в соответствии с поставленной задачей и темой работы. Эта работа помимо получения практического опыта также способствует развитию нового взгляда на традиционные задачи разработки программных комплексов и программного обеспечения.

**Цель работы**: разработка программного продукта для решения квадратных уравнений с заданными коэффициентами «Решатель» на языке С++.

**Задачи работы:**

1. Описать программируемую систему.
2. Рассмотреть существующие решения-аналоги по данной, либо смежной темам.
3. Сформировать требования к программируемой системе.
4. Спроектировать диаграмму состояний системы.
5. Спроектировать диаграмму классов системы.
6. Спроектировать диаграмму последовательности для системы.
7. Реализовать программный продукт в соответствии с требованиями.
8. Провести функциональное тестирование программного продукта.
9. Составить инструкцию по использованию программного продукта.
10. Составить отчет по работе.
11. Сдать отчет и представить его к защите.

**Объектом данного исследования** является разработка автоматизированных методов решения элементарных математических задач с использованием различных языков программирования.

**Данная работа направлена** на автоматизацию решения квадратных уравнений с указанными коэффициентами с использованием языка программирования C++.

**Для данного исследования** были применены методы анализа, сравнения и моделирования. Практическая реализация поставленной задачи соответствует основным принципам разработки программного обеспечения.

**Информационной базой** исследования являются открытые источники, в том числе доступные в сети Интернет, а также материалы курса «Технологии индустриального программирования», доступные через систему дистанционного обучения РТУ МИРЭА.

В данном отчете будет описан процесс разработки программного продукта, включая теоретический обзор предметной области и системы, технологическое проектирование и описание самой системы, а также представление конечных результатов разработки.

# 1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

## 1.1 Описание программируемой системы

Программируемая система для решения квадратных уравнений является инструментом, который обеспечивает возможность ввода квадратного уравнения в определенном формате и автоматического вычисления его корней. Обычно такая система предоставляет пользователю интерфейс для ввода коэффициентов квадратного уравнения (a, b и c) и затем использует специальный алгоритм для расчета корней уравнения.

Программируемые системы для решения квадратных уравнений могут быть разработаны на различных языках программирования, таких как Python, Java, C++ и другие, и могут быть представлены в виде отдельных приложений, веб-сервисов или библиотек.

Квадратное уравнение — это алгебраическое уравнение второй степени, общий вид которого:

* *Приведённое* квадратное уравнение — это уравнение, в котором коэффициент .
* *Неприведённое* квадратное уравнение — это уравнение, в котором коэффициент .
* *Полное* квадратное уравнение — это уравнение, в котором все коэффициенты отличны от нуля.
* *Неполное* квадратное уравнение — это уравнение, в котором хотя бы один коэффициент равен нулю.

В зависимости от значений, которые могут принимать коэффициенты квадратного уравнения, можно выделить шесть типов квадратных уравнений(неполных квадратных уравнений):

1. — все коэффициенты равны нулю, следовательно, уравнение имеет решение при любом .
2. — при , следовательно, уравнение раскладывается в вид и имеет два корня: и
3. — при , следовательно, уравнение раскладывается в вид и имеет единственные корни при , если же , то уравнение не имеет действительных корней.
4. — при , следовательно, уравнение имеет вид и имеет единственный корень
5. — при , следовательно, уравнение имеет вид и единственный корень
6. — при , следовательно, уравнение имеет вид и единственный корень

Для решения квадратных уравнений (полных квадратных уравнений) используются различные методы.

Вот несколько из них:

1. С помощью дискриминанта:

* При — корни уравнения
* При — корень уравнения
* При — действительных корней нет

1. Выделение полного квадрата. Эта процедура подразумевает, что можно решить квадратное уравнение с помощью формул квадрата:  
   и  
   Если же не получается применить формулу квадрата суммы или разности, то нужно вычесть, добавить или разделить коэффициент так, что можно будет применить формулу квадрата суммы или разности.
2. Разложение на множители. Если — корни уравнения , то квадратный трёхчлен можно разложить на множители:  
   .
3. Теорема Виета. Согласно этой теореме, сумма корней приведённого квадратного уравнения равна второму коэффициенту, взятому с противоположным знаком, а произведение:
4. Если , то корнями являются и  .

## 1.2 Обзор существующих решений

Появление и развитие новых технологий и компьютерных вычислений позволили автоматизировать и ускорить процесс решения различных математических задач, включая квадратные уравнения. Рассмотрены три приложения, предназначенные для решения квадратных уравнений.

1. **Microsoft Math Solver**

**Microsoft Math Solver** — задействована система искусственного интеллекта, которая отвечает за поиск правильных решений уравнений, вычислений и статистических данных. Приложение также может показывать ход решения задачи для того, чтобы было проще понять, каким именно образом получился правильный ответ.

Ввести задачу или формулу можно с использованием научного калькулятора или рукописным вводом на экране. Кроме того, можно использовать камеру и сканировать формулы. Приложение распознаёт математически символы и переводит их в форму, понятную системе искусственного интеллекта. Та же самая технология сканирования используется при работе с документами в мобильных офисных приложениях Microsoft.

Плюсы:

* Универсальность: Microsoft Math Solver решает различные математические задачи, от базовой арифметики до сложных задач по исчислению и даже химии.
* Пошаговые решения: Программа содержит подробные объяснения решения задач, что улучшает их понимание.
* Интеграция с другими инструментами: Пользователи могут рисовать или писать математические задачи прямо на экране, кроме того, программа интегрируется с приложениями Microsoft Office.
* Имеет мобильную и веб-браузерскую версию.

Минусы:

* Контекстное понимание: как и многие другие инструменты искусственного интеллекта, Microsoft Math Solver может не полностью понимать контекст задачи, что может привести к неправильной интерпретации.
* Ограниченные расширенные возможности: несмотря на свои возможности, он может быть менее специализированным или всеобъемлющим для решения высокотехнических или узкоспециализированных математических задач.

1. **Photomath**

**Photomath —** мобильное приложение, описанное как «камера-калькулятор», использующее камеру телефона для распознания математических уравнений и отображения пошагового решения на экране. Приложение бесплатно доступно на Android и iOS. Photomath создан компанией Microblink, которая специализируется на оптическом распознавании символов. Релиз состоялся в октябре 2014 года.

Плюсы:

* Распознавание изображений: Отличительной особенностью Photomath является возможность решения математических задач с использованием изображений из учебников или рукописных заметок.
* Пошаговые объяснения: подобно персональному репетитору, программа разбивает решения на подробные пошаговые объяснения.
* Образовательная направленность: Photomath призван способствовать более глубокому пониманию математических концепций посредством визуального и интерактивного обучения.

Минусы:

* Сложные концепции: хотя Photomath эффективен для решения многих математических задач, ему может потребоваться помощь при работе с очень сложными или нестандартными математическими концепциями.
* Зависимость от качества исходных данных: Точность решений зависит от качества исходного изображения; нечеткие или искаженные изображения могут привести к ошибкам.

1. **Mathway**

**Mathway —** мировой лидер среди программ для решения задач, в арсенале которого миллиарды решенных задач и которому доверяют миллионы пользователей. От элементарной алгебры до комплексных расчетов.

Плюсы:

* Разнообразие типов задач: Mathway способен решать различные математические задачи, включая алгебру, исчисление и тригонометрию.
* Мгновенные решения: Mathway предоставляет быстрые решения математических задач, что делает его ценным ресурсом для немедленного решения задач.
* Пошаговый вариант: Пошаговая инструкция помогает понять процесс решения.

Минусы:

* Ограниченное понимание контекста: несмотря на то, что Mathway может давать точные ответы, может потребоваться полное понимание контекста задачи, что может привести к неправильному толкованию.
* Отсутствие концептуальных объяснений: несмотря на полезность пошаговой функции, ей могут потребоваться более глубокие концептуальные объяснения, которые необходимы для реального обучения.

## 1.3 Требования к программируемой системе

В таблице 1.1 представлены требования к программируемой системе.

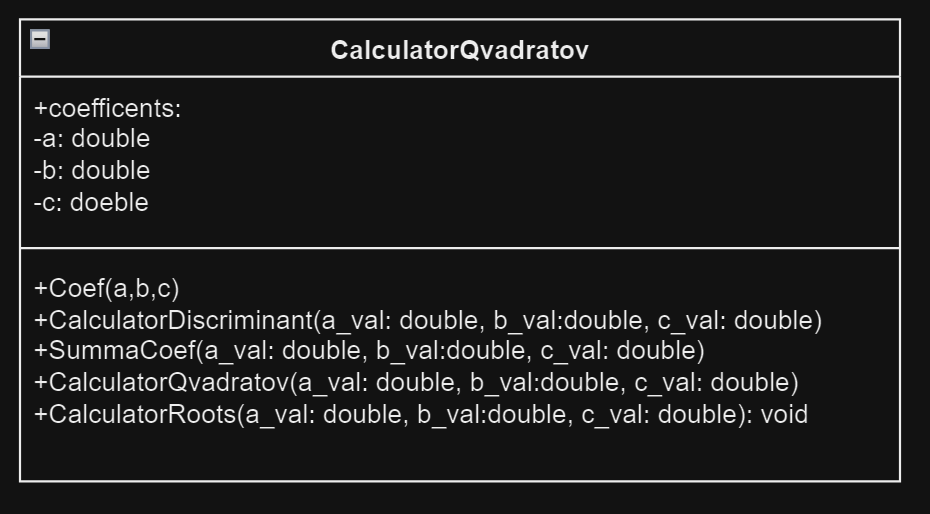
Таблица 1.1 – Требования к программируемой системе

|  |  |  |
| --- | --- | --- |
| № | Требование | Значение |
| 1 | Язык программирования | C++. |
| 2 | Корректность работы | Приложение запускается и поддерживает стабильный цикл работы от момента старта до завершения. |
| 3 | Применение принципов объектно-ориентированного программирования | При написании приложения, как минимум, были использованы классы в C++, объектный подход к проектированию системы, а также инкапсуляция. |
| 4 | Интерфейс пользователя | Создан интерфейс пользователя, поддерживающий корректный пользовательский опыт и содержащий все необходимые пояснения к работе и эксплуатации. |
| 5 | Инструкция по эксплуатации | Написана инструкция по эксплуатации, содержащая, в том числе, основные рекомендации по использованию и пояснения к возможным ошибкам в программе. |
| 6 | Удобство использования | Информация должна быть отображена в легкодоступной и понятной форме для пользователей. |
| 7 | Метод решения | Программа подбирает наиболее подходящий метод решения, в зависимости от заданных коэффициентов. |
| 8 | Простота ввода и изменения данных | Ввод и редактирование информации должны быть интуитивно понятными и удобными. |
| 9 | Показ дискриминанта | Программа должна показывать значение дискриминанта. |
| 10 | Пользовательская справка | Программа должна иметь справку, в которой описаны решения, в зависимости от коэффициентов. |

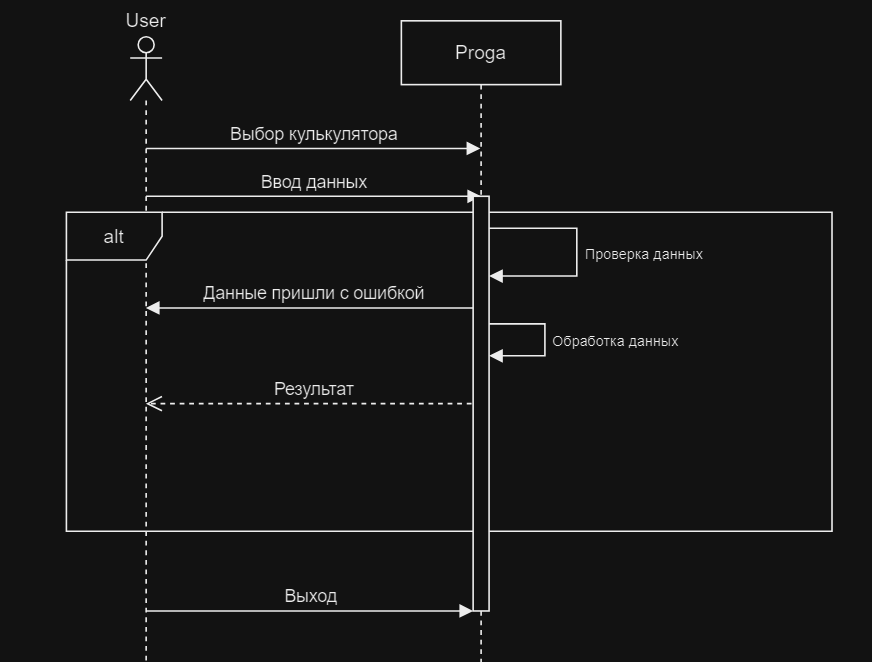
# 2 ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ

## 2.1 Диаграмма состояний системы

## 2.2 Диаграмма классов системы

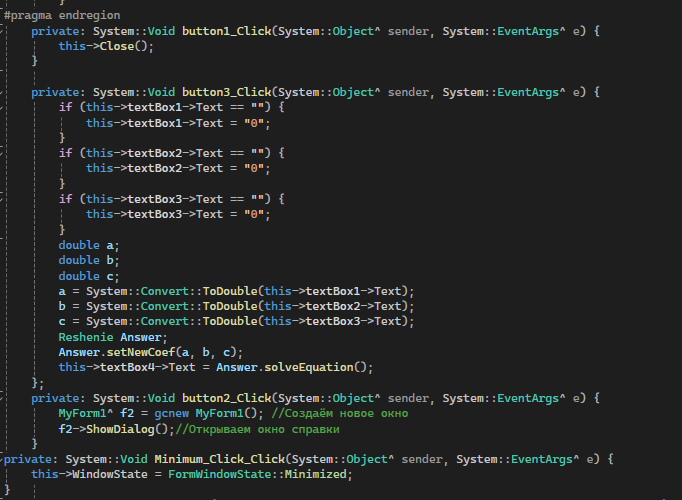


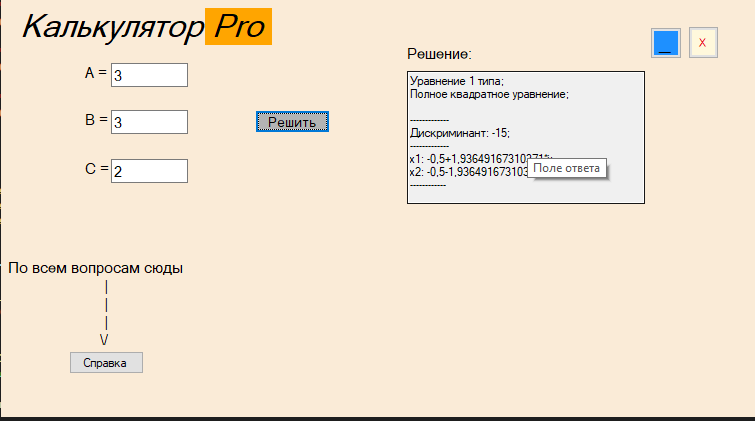
## 2.3 Диаграмма последовательности системы

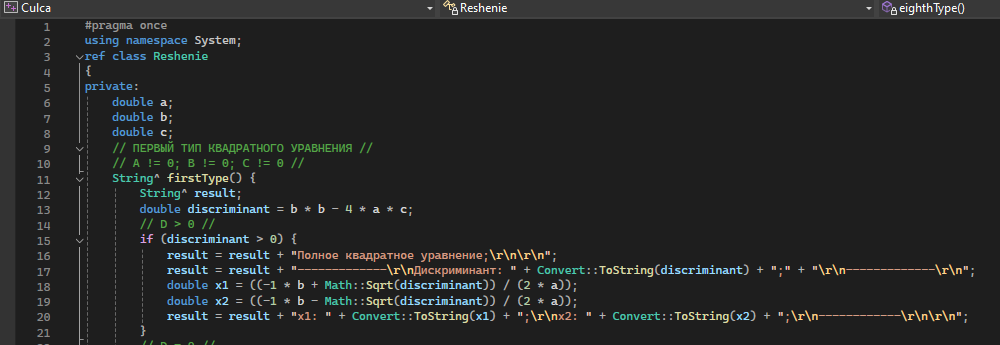


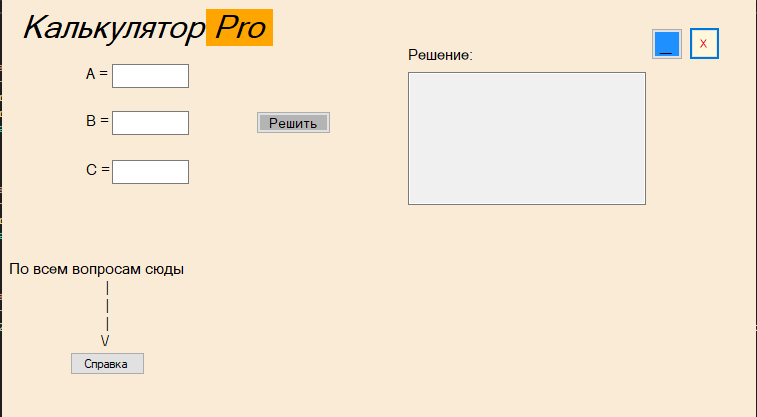
# 3 ПРАКТИЧЕСКАЯ ЧАСТЬ

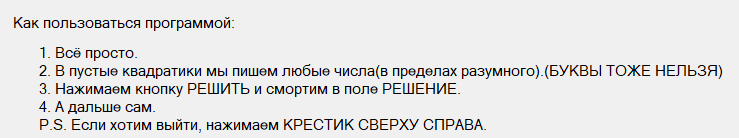
## 3.1 Реализация требований к системе

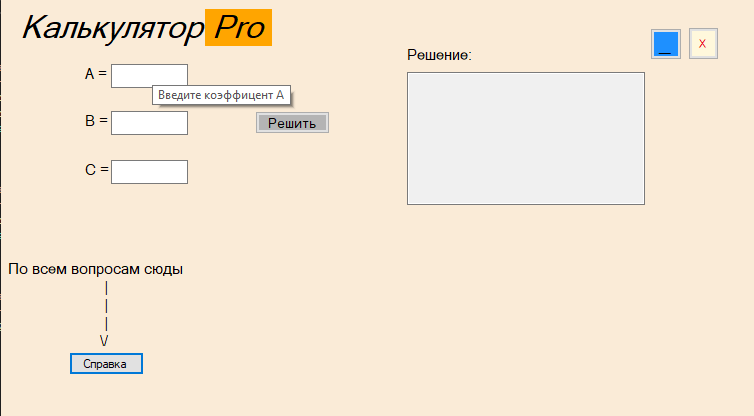
1. Язык программирования C++

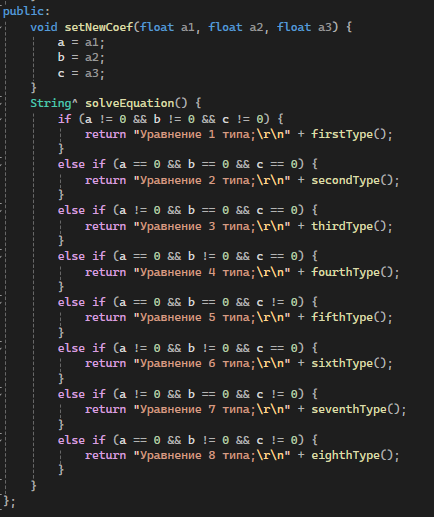
2. Корректность работы – приложение запускается и поддерживает стабильный цикл работы от момента старта до завершения

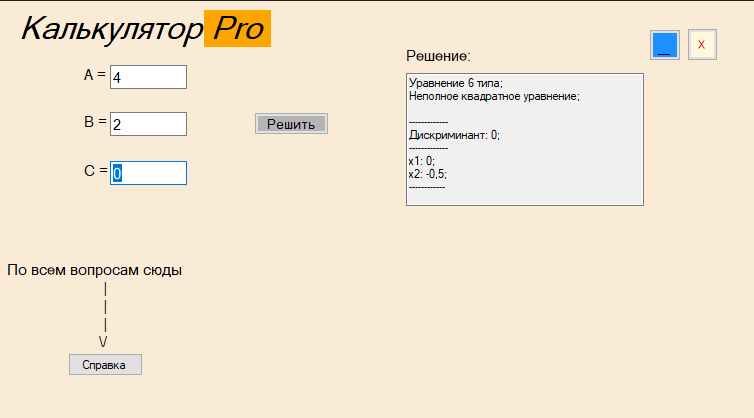
3. Применение принципов объектно-ориентированного программирования - при написании приложения, как минимум, были использованы классы в C++, объектный подход к проектированию системы, а также инкапсуляция. 

4. Интерефейс пользователя - создан интерфейс пользователя, поддерживающий корректный пользовательский опыт и содержащий все необходимые пояснения к работе и эксплуатации. 

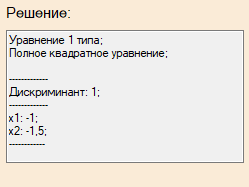
5. Инструкция по эсплуатации - написана инструкция по эксплуатации, содержащая, в том числе, основные рекомендации по использованию и пояснения к возможным ошибкам в программе. 

6. Удобство пользования - информация должна быть отображена в легкодоступной и понятной форме для пользователей. 

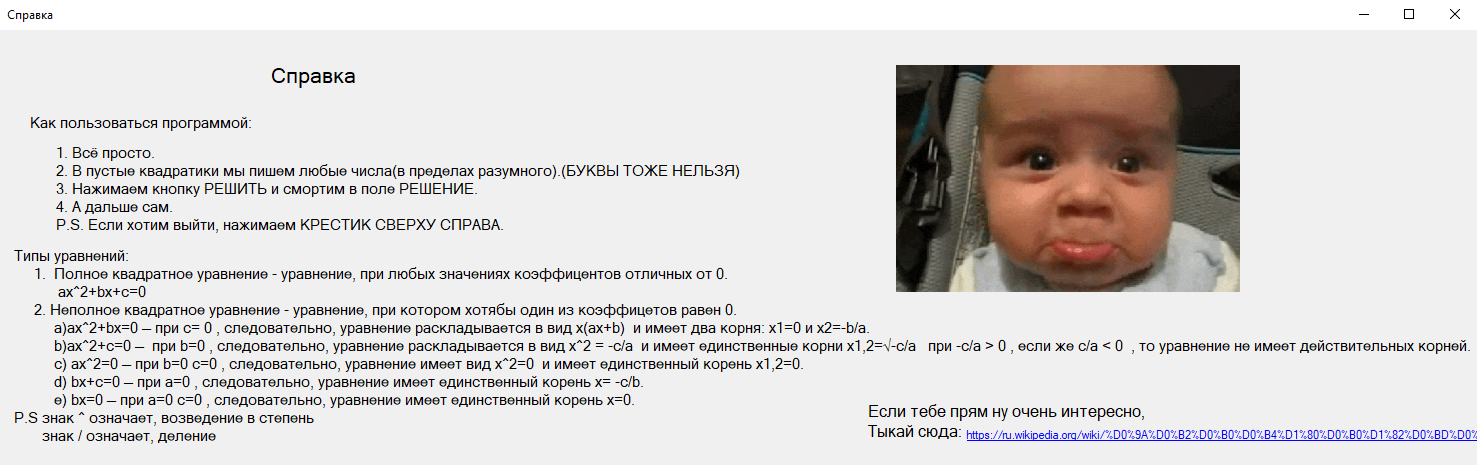
7. Метод решения - программа подбирает наиболее подходящий метод решения, в зависимости от заданных коэффициентов. 

8. Простота ввода и изменение данных - ввод и редактирование информации должны быть интуитивно понятными и удобными. 

9. Показ дискриминанта - программа должна показывать значение дискриминанта.



10. Пользовательская справка - программа должна иметь справку, в которой описаны решения, в зависимости от коэффициентов.



## 3.2 Функциональное тестирование программного продукта

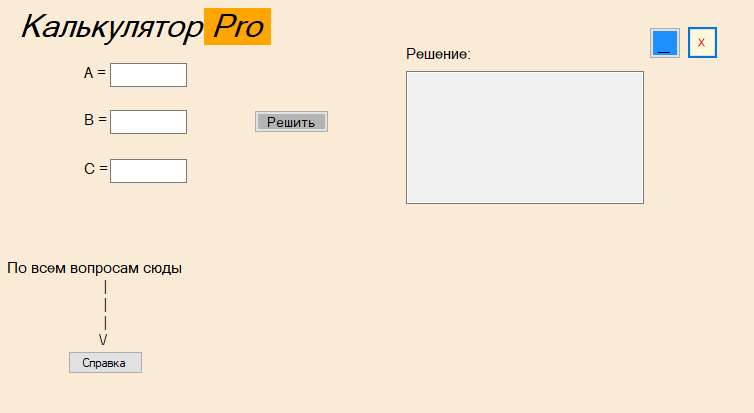
Работа программы начинается с открытия файла с программой. Далее её запуска и показа главного окна (Рисунок 3.1).

Рисунок 3.1 — Главное окно

Работа каждого из пунктов меню реализуется нажатием на отдельные кнопки. Программа включает в себя работу с меню, которое формирует интерфейс пользователя. (Приложение А, Листинг А.1 – А.2).

Для реализации функционала, связанного с решением квадратных уравнений, реализован класс, содержащий методы по всем типам квадратных уравнений. Порядок типов уравнений соответствует порядку, в котором они перечислены в подразделе 1.1 (Приложение А, Листинг А.3).

В главном меню происходит ввод коэффициентов – вводятся коэффициенты квадратного уравнения «A», «B», «C», которые в дальнейшем будут обрабатываться (Рисунок 3.2).

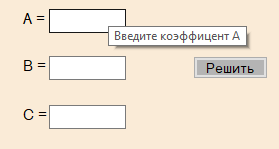


Рисунок 3.2 — Пункт «Ввод коэффициентов»

Кнопка «Справка» вызывает окно справки, в котором описано инструкция по эксплуатации программы пользователем, а также подробное описание методов решения уравнений, используемых в данной программе. Скриншоты работы данного пункта представлены на Рисунках 3.3-3.4 (Приложение А.4 – А.5).

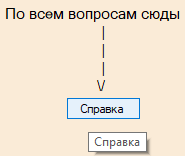


Рисунок 3.3 — Кнопка «Справка» (начало)

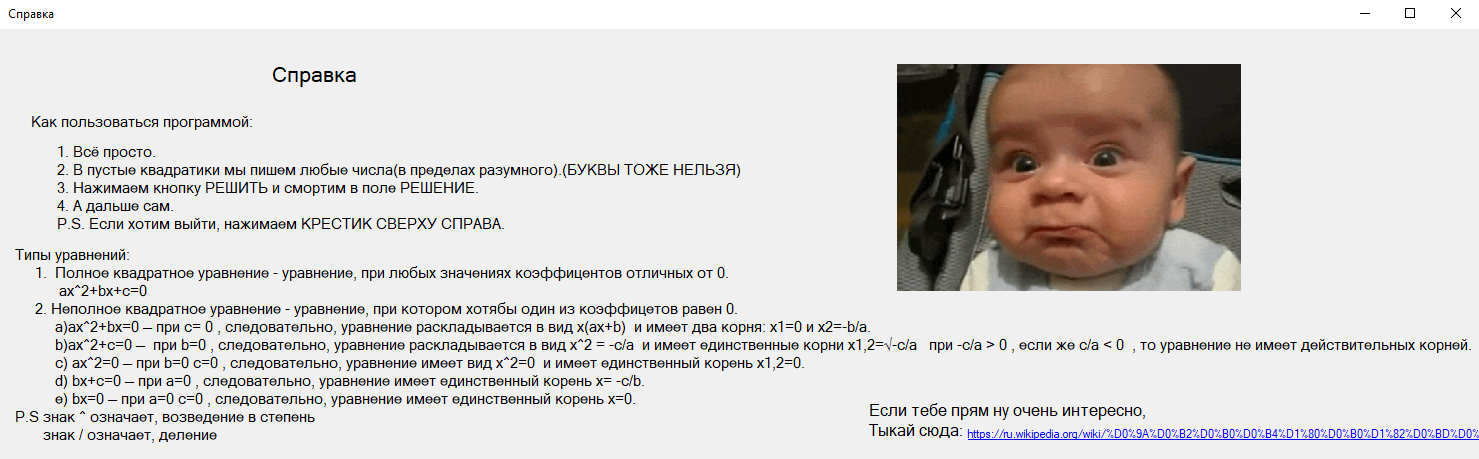


Рисунок 3.4 — Кнопка «Справка» (конец)

Кнопка «Решить» пользователь запускает решение уравнения, заданного коэффициентами «A», «B», «C», введённые в определённые поля, после чего формируется ответ и выводится пользователю в поле «Решение». Скриншоты работы данного пункта представлены на Рисунках 3.5-3.6.



Рисунок 3.4 — Кнопка «Решить»

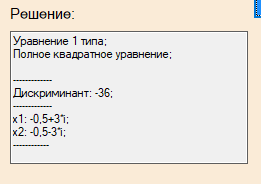


Рисунок 3.5 — Поле «Решение»

## 3.3 Инструкция по эксплуатации

1. Открыть файл с программой.
2. Запустить программу.
3. Введите коэффициенты уравнения в пустые поля «A», «B», «C».
4. Нажмите на кнопку «Решить». Далее смотрите в поле «Решение».
5. При некорректной работе программы откройте окно справки, нажав на кнопку «Справка».

# ЗАКЛЮЧЕНИЕ

В результате выполнения курсовой работы была описана программируемая система, рассмотрены существующие решения-аналоги по теме, сформированы требования к системе, спроектированы диаграммы состояний, классов и последовательности самой системы. Сам программный продукты был разработан в соответствии с требованиями, протестирован, а также была написана инструкция по его эксплуатации.

Перспективы развития данного продукта могут состоять в добавлении дополнительных математических операций, разрешении квадратных уравнений с комплексными корнями, а также в улучшении интерфейса пользователя.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Теоретический материал по квадратным уравнениям из свободного источника - <https://ru.wikipedia.org/wiki/Квадратное_уравнение>
2. Теоретический материал по решению квадратных уравнений - <https://tetrika-school.ru/blog/kvadratnie-uravneniya>
3. Теоретический материал по решению квадратных уравнений с помощью разложения на множители - <https://foxford.ru/wiki/matematika/razlozheniye-na-mnozhiteli-kvadratnykh-trekhchlenov?utm_referrer=https%3A%2F%2Fyandex.ru%2F>
4. Плюсы и минусы популярных калькуляторов для решения квадратных уравнений - <https://www.aiproductslist.com/ru/blog/11-best-ai-tools-for-math-success>
5. Информация про калькулятор Math Solver - <https://www.iguides.ru/main/other/microsoft_math_solver_prilozhenie_dlya_avtomaticheskogo_resheniya_matematicheskikh_zadach>
6. Информация про калькулятор Photomath - <https://ru.wikipedia.org/wiki/Photomath#:~:text=Photomath%20—%20мобильное%20приложение%2C%20описанное,состоялся%20в%20октябре%202014%20года>
7. Информация про калькулятор Mathway - <https://4pda.to/forum/index.php?showtopic=831214#:~:text=Описание%3A%20Mathway%20—%20мировой%20лидер,вы%20немедленно%20получите%20бесплатный%20ответ>
8. Как построить графики функций на языке C++ - <https://www.youtube.com/watch?v=hnR8s3XUnuw&ab_channel=Duha>
9. Работы с графиками на C++ - <https://learn.microsoft.com/ru-ru/cpp/dotnet/graphics-operations-cpp-cli?view=msvc-170>

# ПРИЛОЖЕНИЯ

## Приложение А – Исходный код программы

Листинг А.1 — Файл MyForm.cpp

|  |
| --- |
| *#include "MyForm.h"*  *using namespace System;*  *using namespace System::Windows::Forms;*  *[STAThreadAttribute]*  *int main(array<String^>^ args) {*  *Application::SetCompatibleTextRenderingDefault(false);*  *Application::EnableVisualStyles();*  *Culca::MyForm form;*  *Application::Run(% form);*  *}* |

Листинг А.2 — Файл MyForm.h

|  |
| --- |
| #pragma once  #include "MyForm1.h"  #include "Header.h"  namespace Culca {  using namespace System;  using namespace System::ComponentModel;  using namespace System::Collections;  using namespace System::Windows::Forms;  using namespace System::Data;  using namespace System::Drawing;  /// <summary>  /// Сводка для MyForm  /// </summary>  public ref class MyForm : public System::Windows::Forms::Form  {  public:bool mouseDown;  private: System::Windows::Forms::ToolTip^ toolTip1;  private: System::Windows::Forms::ToolTip^ toolTip2;  private: System::Windows::Forms::ToolTip^ toolTip3;  private: System::Windows::Forms::ToolTip^ toolTip4;  private: System::Windows::Forms::ToolTip^ toolTip5;  private: System::Windows::Forms::ToolTip^ toolTip6;  private: System::Windows::Forms::Label^ label6;  private: System::Windows::Forms::ToolTip^ toolTip7;  private: System::Windows::Forms::ToolTip^ toolTip8;  private: System::Windows::Forms::Label^ label7;  public:  public: Point lastLocation;  public:  MyForm(void)  {  InitializeComponent();  //  //TODO: добавьте код конструктора  //  }  protected:  /// <summary>  /// Освободить все используемые ресурсы.  /// </summary>  ~MyForm()  {  if (components)  {  delete components;  }  }  private: System::Windows::Forms::Button^ button1;  private: System::Windows::Forms::Label^ label1;  private: System::Windows::Forms::TextBox^ textBox1;  private: System::Windows::Forms::TextBox^ textBox2;  private: System::Windows::Forms::Label^ label2;  private: System::Windows::Forms::TextBox^ textBox3;  private: System::Windows::Forms::Label^ label3;  private: System::Windows::Forms::Label^ label4;  private: System::Windows::Forms::TextBox^ textBox4;  private: System::Windows::Forms::Button^ button2;  private: System::Windows::Forms::Button^ button3;  private: System::Windows::Forms::Label^ label5;  private: System::Windows::Forms::Button^ Minimum\_Click;  private: System::ComponentModel::IContainer^ components;  protected:  protected:  private:  /// <summary>  /// Обязательная переменная конструктора.  /// </summary>  #pragma region Windows Form Designer generated code  /// <summary>  /// Требуемый метод для поддержки конструктора — не изменяйте  /// содержимое этого метода с помощью редактора кода.  /// </summary>  void InitializeComponent(void)  {  this->components = (gcnew System::ComponentModel::Container());  this->button1 = (gcnew System::Windows::Forms::Button());  this->label1 = (gcnew System::Windows::Forms::Label());  this->textBox1 = (gcnew System::Windows::Forms::TextBox());  this->textBox2 = (gcnew System::Windows::Forms::TextBox());  this->label2 = (gcnew System::Windows::Forms::Label());  this->textBox3 = (gcnew System::Windows::Forms::TextBox());  this->label3 = (gcnew System::Windows::Forms::Label());  this->label4 = (gcnew System::Windows::Forms::Label());  this->textBox4 = (gcnew System::Windows::Forms::TextBox());  this->button2 = (gcnew System::Windows::Forms::Button());  this->button3 = (gcnew System::Windows::Forms::Button());  this->label5 = (gcnew System::Windows::Forms::Label());  this->Minimum\_Click = (gcnew System::Windows::Forms::Button());  this->toolTip1 = (gcnew System::Windows::Forms::ToolTip(this->components));  this->toolTip2 = (gcnew System::Windows::Forms::ToolTip(this->components));  this->toolTip3 = (gcnew System::Windows::Forms::ToolTip(this->components));  this->toolTip4 = (gcnew System::Windows::Forms::ToolTip(this->components));  this->toolTip5 = (gcnew System::Windows::Forms::ToolTip(this->components));  this->toolTip6 = (gcnew System::Windows::Forms::ToolTip(this->components));  this->label6 = (gcnew System::Windows::Forms::Label());  this->toolTip7 = (gcnew System::Windows::Forms::ToolTip(this->components));  this->toolTip8 = (gcnew System::Windows::Forms::ToolTip(this->components));  this->label7 = (gcnew System::Windows::Forms::Label());  this->SuspendLayout();  //  // button1  //  this->button1->BackColor = System::Drawing::Color::Cornsilk;  this->button1->ForeColor = System::Drawing::SystemColors::Desktop;  this->button1->Location = System::Drawing::Point(687, 27);  this->button1->Name = L"button1";  this->button1->Size = System::Drawing::Size(31, 33);  this->button1->TabIndex = 0;  this->button1->Text = L"X";  this->toolTip1->SetToolTip(this->button1, L"Выход");  this->button1->UseVisualStyleBackColor = false;  this->button1->Click += gcnew System::EventHandler(this, &MyForm::button1\_Click);  //  // label1  //  this->label1->AutoSize = true;  this->label1->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 11.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,  static\_cast<System::Byte>(204)));  this->label1->Location = System::Drawing::Point(81, 64);  this->label1->Name = L"label1";  this->label1->Size = System::Drawing::Size(30, 18);  this->label1->TabIndex = 1;  this->label1->Text = L"A =";  //  // textBox1  //  this->textBox1->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 11.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,  static\_cast<System::Byte>(204)));  this->textBox1->Location = System::Drawing::Point(110, 64);  this->textBox1->Name = L"textBox1";  this->textBox1->Size = System::Drawing::Size(77, 24);  this->textBox1->TabIndex = 2;  this->toolTip3->SetToolTip(this->textBox1, L"Введите коэффицент А");  //  // textBox2  //  this->textBox2->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 11.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,  static\_cast<System::Byte>(204)));  this->textBox2->Location = System::Drawing::Point(110, 111);  this->textBox2->Name = L"textBox2";  this->textBox2->Size = System::Drawing::Size(77, 24);  this->textBox2->TabIndex = 4;  this->toolTip4->SetToolTip(this->textBox2, L"Введите коэффицент В");  //  // label2  //  this->label2->AutoSize = true;  this->label2->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 11.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,  static\_cast<System::Byte>(204)));  this->label2->Location = System::Drawing::Point(81, 111);  this->label2->Name = L"label2";  this->label2->Size = System::Drawing::Size(31, 18);  this->label2->TabIndex = 3;  this->label2->Text = L"B =";  //  // textBox3  //  this->textBox3->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 11.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,  static\_cast<System::Byte>(204)));  this->textBox3->Location = System::Drawing::Point(110, 160);  this->textBox3->Name = L"textBox3";  this->textBox3->Size = System::Drawing::Size(77, 24);  this->textBox3->TabIndex = 6;  this->toolTip5->SetToolTip(this->textBox3, L"Введите коэффицент С");  //  // label3  //  this->label3->AutoSize = true;  this->label3->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 11.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,  static\_cast<System::Byte>(204)));  this->label3->Location = System::Drawing::Point(81, 160);  this->label3->Name = L"label3";  this->label3->Size = System::Drawing::Size(32, 18);  this->label3->TabIndex = 5;  this->label3->Text = L"C =";  //  // label4  //  this->label4->AutoSize = true;  this->label4->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 11.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,  static\_cast<System::Byte>(204)));  this->label4->Location = System::Drawing::Point(403, 45);  this->label4->Name = L"label4";  this->label4->Size = System::Drawing::Size(73, 18);  this->label4->TabIndex = 7;  this->label4->Text = L"Решение:";  //  // textBox4  //  this->textBox4->Location = System::Drawing::Point(406, 72);  this->textBox4->Multiline = true;  this->textBox4->Name = L"textBox4";  this->textBox4->ReadOnly = true;  this->textBox4->Size = System::Drawing::Size(238, 133);  this->textBox4->TabIndex = 8;  this->toolTip8->SetToolTip(this->textBox4, L"Поле ответа");  //  // button2  //  this->button2->Location = System::Drawing::Point(68, 352);  this->button2->Name = L"button2";  this->button2->Size = System::Drawing::Size(75, 23);  this->button2->TabIndex = 9;  this->button2->Text = L"Справка";  this->toolTip6->SetToolTip(this->button2, L"Справка");  this->button2->UseVisualStyleBackColor = true;  this->button2->Click += gcnew System::EventHandler(this, &MyForm::button2\_Click);  //  // button3  //  this->button3->BackColor = System::Drawing::SystemColors::ActiveBorder;  this->button3->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 9.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,  static\_cast<System::Byte>(204)));  this->button3->ForeColor = System::Drawing::SystemColors::ControlText;  this->button3->Location = System::Drawing::Point(254, 111);  this->button3->Name = L"button3";  this->button3->Size = System::Drawing::Size(75, 23);  this->button3->TabIndex = 10;  this->button3->Text = L"Решить";  this->toolTip7->SetToolTip(this->button3, L"Нажми, чтобы решить");  this->button3->UseVisualStyleBackColor = false;  this->button3->Click += gcnew System::EventHandler(this, &MyForm::button3\_Click);  //  // label5  //  this->label5->AutoSize = true;  this->label5->BackColor = System::Drawing::Color::AntiqueWhite;  this->label5->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 24, System::Drawing::FontStyle::Italic, System::Drawing::GraphicsUnit::Point,  static\_cast<System::Byte>(0)));  this->label5->ForeColor = System::Drawing::SystemColors::InfoText;  this->label5->Location = System::Drawing::Point(12, 9);  this->label5->Name = L"label5";  this->label5->Size = System::Drawing::Size(199, 37);  this->label5->TabIndex = 11;  this->label5->Text = L"Калькулятор";  //  // Minimum\_Click  //  this->Minimum\_Click->BackColor = System::Drawing::Color::DodgerBlue;  this->Minimum\_Click->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 14.25F, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,  static\_cast<System::Byte>(204)));  this->Minimum\_Click->Location = System::Drawing::Point(649, 28);  this->Minimum\_Click->Name = L"Minimum\_Click";  this->Minimum\_Click->Size = System::Drawing::Size(32, 32);  this->Minimum\_Click->TabIndex = 12;  this->Minimum\_Click->Text = L"\_\_";  this->toolTip2->SetToolTip(this->Minimum\_Click, L"Свернуть");  this->Minimum\_Click->UseVisualStyleBackColor = false;  this->Minimum\_Click->Click += gcnew System::EventHandler(this, &MyForm::Minimum\_Click\_Click);  //  // label6  //  this->label6->AutoSize = true;  this->label6->BackColor = System::Drawing::Color::Orange;  this->label6->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 24, System::Drawing::FontStyle::Italic, System::Drawing::GraphicsUnit::Point,  static\_cast<System::Byte>(204)));  this->label6->Location = System::Drawing::Point(204, 9);  this->label6->Name = L"label6";  this->label6->Size = System::Drawing::Size(67, 37);  this->label6->TabIndex = 13;  this->label6->Text = L"Pro";  //  // label7  //  this->label7->AutoSize = true;  this->label7->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 11.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,  static\_cast<System::Byte>(204)));  this->label7->Location = System::Drawing::Point(4, 259);  this->label7->Name = L"label7";  this->label7->Size = System::Drawing::Size(183, 90);  this->label7->TabIndex = 14;  this->label7->Text = L"По всем вопросам сюды\r\n |\r\n |\r\n "  L" |\r\n \\/";  //  // MyForm  //  this->AccessibleName = L"";  this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);  this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;  this->BackColor = System::Drawing::Color::AntiqueWhite;  this->ClientSize = System::Drawing::Size(754, 418);  this->Controls->Add(this->label7);  this->Controls->Add(this->label6);  this->Controls->Add(this->Minimum\_Click);  this->Controls->Add(this->label5);  this->Controls->Add(this->button3);  this->Controls->Add(this->button2);  this->Controls->Add(this->textBox4);  this->Controls->Add(this->label4);  this->Controls->Add(this->textBox3);  this->Controls->Add(this->label3);  this->Controls->Add(this->textBox2);  this->Controls->Add(this->label2);  this->Controls->Add(this->textBox1);  this->Controls->Add(this->label1);  this->Controls->Add(this->button1);  this->FormBorderStyle = System::Windows::Forms::FormBorderStyle::None;  this->Name = L"MyForm";  this->StartPosition = System::Windows::Forms::FormStartPosition::CenterScreen;  this->Text = L"Калькулятор";  this->Load += gcnew System::EventHandler(this, &MyForm::MyForm\_Load);  this->MouseDown += gcnew System::Windows::Forms::MouseEventHandler(this, &MyForm::MyForm\_MouseDown);  this->MouseMove += gcnew System::Windows::Forms::MouseEventHandler(this, &MyForm::MyForm\_MouseMove);  this->MouseUp += gcnew System::Windows::Forms::MouseEventHandler(this, &MyForm::MyForm\_MouseUp);  this->ResumeLayout(false);  this->PerformLayout();  }  #pragma endregion  private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e) {  this->Close();  }  private: System::Void button3\_Click(System::Object^ sender, System::EventArgs^ e) {  if (this->textBox1->Text == "") {  this->textBox1->Text = "0";  }  if (this->textBox2->Text == "") {  this->textBox2->Text = "0";  }  if (this->textBox3->Text == "") {  this->textBox3->Text = "0";  }  double a;  double b;  double c;  a = System::Convert::ToDouble(this->textBox1->Text);  b = System::Convert::ToDouble(this->textBox2->Text);  c = System::Convert::ToDouble(this->textBox3->Text);  Reshenie Answer;  Answer.setNewCoef(a, b, c);  this->textBox4->Text = Answer.solveEquation();  };  private: System::Void button2\_Click(System::Object^ sender, System::EventArgs^ e) {  MyForm1^ f2 = gcnew MyForm1(); //Создаём новое окно  f2->ShowDialog();//Открываем окно справки  }  private: System::Void Minimum\_Click\_Click(System::Object^ sender, System::EventArgs^ e) {  this->WindowState = FormWindowState::Minimized;  }  private: void MyForm\_MouseDown(System::Object^ sender, System::Windows::Forms::MouseEventArgs^ e)  {  mouseDown = true;  lastLocation = e->Location;  }  private: void MyForm\_MouseMove(System::Object^ sender, System::Windows::Forms::MouseEventArgs^ e)  {  if (mouseDown)  {  this->Location = System::Drawing::Point(  (this->Location.X - lastLocation.X) + e->X, (this->Location.Y - lastLocation.Y) + e->Y);  this->Update();  }  }  private: void MyForm\_MouseUp(System::Object^ sender, System::Windows::Forms::MouseEventArgs^ e)  {  mouseDown = false;  }  private: System::Void MyForm\_Load(System::Object^ sender, System::EventArgs^ e) {  }  };  } |

Листинг А.3 — Файл Header.h

|  |
| --- |
| #pragma once  using namespace System;  ref class Reshenie  {  private:  double a;  double b;  double c;  // ПЕРВЫЙ ТИП КВАДРАТНОГО УРАВНЕНИЯ //  // A != 0; B != 0; C != 0 //  String^ firstType() {  String^ result;  double discriminant = b \* b - 4 \* a \* c;  // D > 0 //  if (discriminant > 0) {  result = result + "Полное квадратное уравнение;\r\n\r\n";  result = result + "-------------\r\nДискриминант: " + Convert::ToString(discriminant) + ";" + "\r\n-------------\r\n";  double x1 = ((-1 \* b + Math::Sqrt(discriminant)) / (2 \* a));  double x2 = ((-1 \* b - Math::Sqrt(discriminant)) / (2 \* a));  result = result + "x1: " + Convert::ToString(x1) + ";\r\nx2: " + Convert::ToString(x2) + ";\r\n------------\r\n\r\n";  }  // D = 0 //  else if (discriminant == 0) {  result = result + "Полное квадратное уравнение;\r\n\r\n";  result = result + "-------------\r\nДискриминант: " + Convert::ToString(discriminant) + ";" + "\r\n-------------\r\n";  double x1 = ((-1 \* b + Math::Sqrt(discriminant)) / (2 \* a));  result = result + "x1: " + Convert::ToString(x1) + "\r\n------------\r\n\r\n";  }  // D < 0 //  else {  result = result + "Полное квадратное уравнение;\r\n\r\n";  result = result + "-------------\r\nДискриминант: " + Convert::ToString(discriminant) + ";" + "\r\n-------------\r\n";  result = result + "x1: " + -b/(2\*a) + "+" + (Math::Sqrt(Math::Abs(discriminant))) / 2 + "\*i" + ";\r\nx2: " + -b / (2 \* a) + "-" + (Math::Sqrt(Math::Abs(discriminant))) / 2 + "\*i" + ";" + "\r\n------------\r\n\r\n";  }  return result;  }  // ВТОРОЙ ТИП КВАДРАТНОГО УРАВНЕНИЯ //  // A = 0; B = 0; C = 0 //  String^ secondType() {  String^ result;  double discriminant = b \* b - 4 \* a \* c;  result = result + "Неполное квадратное уравнение;\r\n\r\n";  result = result + "-------------\r\nДискриминант: " + Convert::ToString(discriminant) + ";" + "\r\n-------------\r\n";  result = result + "Всегда будет 0\r\n------------\r\n\r\n";  return result;  }  // ТРЕТИЙ ТИП КВАДРАТНОГО УРАВНЕНИЯ.  // A != 0; B = 0; C = 0 //  String^ thirdType() {  String^ result;  double discriminant = b \* b - 4 \* a \* c;  result = result + "Неполное квадратное уравнение;\r\n\r\n";  result = result + "-------------\r\nДискриминант: " + Convert::ToString(discriminant) + ";" + "\r\n-------------\r\n";  result = result + "x = 0;\r\n------------\r\n\r\n";  return result;  }  // ЧЕТВЕРТЫЙ ТИП КВАДРАТНОГО УРАВНЕНИЯ.  // A = 0; B != 0; C = 0 //  String^ fourthType() {  String^ result;  double discriminant = b \* b - 4 \* a \* c;  result = result + "Неполное квадратное уравнение;\r\n\r\n";  result = result + "-------------\r\nДискриминант: " + "0" + ";" + "\r\n-------------\r\n";  result = result + "x = 0;\r\n------------\r\n\r\n";  return result;  }  // ПЯТЫЙ ТИП КВАДРАТНОГО УРАВНЕНИЯ //  // A = 0; B = 0; C != 0;  String^ fifthType() {  String^ result;  double discriminant = b \* b - 4 \* a \* c;  result = result + "Неполное квадратное уравнение;\r\n\r\n";  result = result + "-------------\r\nДискриминант: " + "0" + ";" + "\r\n-------------\r\n";  result = result + "Уравнение не имеет корней (является неверным);\r\n------------\r\n\r\n";  return result;  }  // ШЕСТОЙ ТИП КВАДРАТНОГО УРАВНЕНИЯ //  // A != 0; B != 0; C = 0 //  String^ sixthType() {  String^ result;  double discriminant = b \* b - 4 \* a \* c;  result = result + "Неполное квадратное уравнение;\r\n\r\n";  result = result + "-------------\r\nДискриминант: " + "0" + ";" + "\r\n-------------\r\n";  String^ x1 = Convert::ToString(0);  String^ x2 = Convert::ToString(-1 \* (b / a));  result = result + "x1: " + x1 + ";" + "\r\n" + "x2: " + x2 + ";\r\n------------\r\n\r\n";  return result;  }  // СЕДЬМОЙ ТИП КВАДРАТНОГО УРАВНЕНИЯ //  // A != 0; B = 0; C != 0 //  String^ seventhType() {  String^ result;  double discriminant = b \* b - 4 \* a \* c;  result = result + "Неполное квадратное уравнение;\r\n\r\n";  result = result + "-------------\r\nДискриминант: " + "0" + ";" + "\r\n-------------\r\n";  String^ x1 = Convert::ToString(Math::Sqrt(-1 \* (c / a)));  String^ x2 = Convert::ToString(-1 \* System::Math::Sqrt(-1 \* (c / a)));  result = result + "x1: " + x1 + ";\r\n x2: " + x2 + "\r\n------------\r\n\r\n";  return result;  }  // ВОСЬМОЙ ТИП КВАДРАТНОГО УРАВНЕНИЯ //  // A = 0; B != 0; C != 0 //  String^ eighthType() {  String^ result;  double discriminant = b \* b - 4 \* a \* c;  result = result + "Неполное квадратное уравнение;\r\n\r\n";  result = result + "-------------\r\nДискриминант: " + "0" + ";" + "\r\n-------------\r\n";  String^ x1 = Convert::ToString(-1 \* (c / b));  result = result + "x: " + x1 + "\r\n------------\r\n\r\n";  return result;  }  public:  void setNewCoef(float a1, float a2, float a3) {  a = a1;  b = a2;  c = a3;  }  String^ solveEquation() {  if (a != 0 && b != 0 && c != 0) {  return "Уравнение 1 типа;\r\n" + firstType();  }  else if (a == 0 && b == 0 && c == 0) {  return "Уравнение 2 типа;\r\n" + secondType();  }  else if (a != 0 && b == 0 && c == 0) {  return "Уравнение 3 типа;\r\n" + thirdType();  }  else if (a == 0 && b != 0 && c == 0) {  return "Уравнение 4 типа;\r\n" + fourthType();  }  else if (a == 0 && b == 0 && c != 0) {  return "Уравнение 5 типа;\r\n" + fifthType();  }  else if (a != 0 && b != 0 && c == 0) {  return "Уравнение 6 типа;\r\n" + sixthType();  }  else if (a != 0 && b == 0 && c != 0) {  return "Уравнение 7 типа;\r\n" + seventhType();  }  else if (a == 0 && b != 0 && c != 0) {  return "Уравнение 8 типа;\r\n" + eighthType();  }  }  }; |

Листинг А.4 — Файл MyForm1.cpp

|  |
| --- |
| *#include "MyForm1.h"* |

Листинг А.5 — Файл MyForm1.h

|  |
| --- |
| #pragma once  namespace Culca {  using namespace System;  using namespace System::ComponentModel;  using namespace System::Collections;  using namespace System::Windows::Forms;  using namespace System::Data;  using namespace System::Drawing;  /// <summary>  /// Сводка для MyForm1  /// </summary>  public ref class MyForm1 : public System::Windows::Forms::Form  {  public:  MyForm1(void)  {  InitializeComponent();  //  //TODO: добавьте код конструктора  //  }  protected:  /// <summary>  /// Освободить все используемые ресурсы.  /// </summary>  ~MyForm1()  {  if (components)  {  delete components;  }  }  private: System::Windows::Forms::Label^ label1;  private: System::Windows::Forms::Label^ label2;  private: System::Windows::Forms::Label^ label3;  private: System::Windows::Forms::Label^ label4;  private: System::Windows::Forms::PictureBox^ pictureBox1;  private: System::Windows::Forms::Label^ label5;  private: System::Windows::Forms::LinkLabel^ linkLabel1;  protected:  protected:  private:  /// <summary>  /// Обязательная переменная конструктора.  /// </summary>  System::ComponentModel::Container ^components;  #pragma region Windows Form Designer generated code  /// <summary>  /// Требуемый метод для поддержки конструктора — не изменяйте  /// содержимое этого метода с помощью редактора кода.  /// </summary>  void InitializeComponent(void)  {  System::ComponentModel::ComponentResourceManager^ resources = (gcnew System::ComponentModel::ComponentResourceManager(MyForm1::typeid));  this->label1 = (gcnew System::Windows::Forms::Label());  this->label2 = (gcnew System::Windows::Forms::Label());  this->label3 = (gcnew System::Windows::Forms::Label());  this->label4 = (gcnew System::Windows::Forms::Label());  this->pictureBox1 = (gcnew System::Windows::Forms::PictureBox());  this->label5 = (gcnew System::Windows::Forms::Label());  this->linkLabel1 = (gcnew System::Windows::Forms::LinkLabel());  (cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox1))->BeginInit();  this->SuspendLayout();  //  // label1  //  this->label1->AutoSize = true;  this->label1->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 15.75F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,  static\_cast<System::Byte>(204)));  this->label1->Location = System::Drawing::Point(267, 33);  this->label1->Name = L"label1";  this->label1->Size = System::Drawing::Size(97, 25);  this->label1->TabIndex = 0;  this->label1->Text = L"Справка";  //  // label2  //  this->label2->AutoSize = true;  this->label2->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 11.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,  static\_cast<System::Byte>(204)));  this->label2->Location = System::Drawing::Point(28, 83);  this->label2->Name = L"label2";  this->label2->Size = System::Drawing::Size(230, 18);  this->label2->TabIndex = 1;  this->label2->Text = L"Как пользоваться программой:";  //  // label3  //  this->label3->AutoSize = true;  this->label3->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 11.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,  static\_cast<System::Byte>(204)));  this->label3->Location = System::Drawing::Point(54, 113);  this->label3->Name = L"label3";  this->label3->Size = System::Drawing::Size(692, 90);  this->label3->TabIndex = 2;  this->label3->Text = resources->GetString(L"label3.Text");  //  // label4  //  this->label4->AutoSize = true;  this->label4->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 11.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,  static\_cast<System::Byte>(204)));  this->label4->Location = System::Drawing::Point(12, 216);  this->label4->Name = L"label4";  this->label4->Size = System::Drawing::Size(1465, 198);  this->label4->TabIndex = 3;  this->label4->Text = resources->GetString(L"label4.Text");  //  // pictureBox1  //  this->pictureBox1->Image = (cli::safe\_cast<System::Drawing::Image^>(resources->GetObject(L"pictureBox1.Image")));  this->pictureBox1->Location = System::Drawing::Point(897, 33);  this->pictureBox1->Name = L"pictureBox1";  this->pictureBox1->Size = System::Drawing::Size(344, 232);  this->pictureBox1->SizeMode = System::Windows::Forms::PictureBoxSizeMode::Zoom;  this->pictureBox1->TabIndex = 4;  this->pictureBox1->TabStop = false;  //  // label5  //  this->label5->AutoSize = true;  this->label5->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 12, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,  static\_cast<System::Byte>(204)));  this->label5->Location = System::Drawing::Point(865, 371);  this->label5->Name = L"label5";  this->label5->Size = System::Drawing::Size(286, 40);  this->label5->TabIndex = 5;  this->label5->Text = L"Если тебе прям ну очень интересно,\r\nТыкай сюда:";  //  // linkLabel1  //  this->linkLabel1->AutoSize = true;  this->linkLabel1->Location = System::Drawing::Point(965, 398);  this->linkLabel1->Name = L"linkLabel1";  this->linkLabel1->Size = System::Drawing::Size(971, 13);  this->linkLabel1->TabIndex = 6;  this->linkLabel1->TabStop = true;  this->linkLabel1->Text = L"https://ru.wikipedia.org/wiki/%D0%9A%D0%B2%D0%B0%D0%B4%D1%80%D0%B0%D1%82%D0%BD%D0"  L"%BE%D0%B5\_%D1%83%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5";  //  // MyForm1  //  this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);  this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;  this->ClientSize = System::Drawing::Size(1479, 436);  this->Controls->Add(this->linkLabel1);  this->Controls->Add(this->label5);  this->Controls->Add(this->pictureBox1);  this->Controls->Add(this->label4);  this->Controls->Add(this->label3);  this->Controls->Add(this->label2);  this->Controls->Add(this->label1);  this->FormBorderStyle = System::Windows::Forms::FormBorderStyle::FixedDialog;  this->Name = L"MyForm1";  this->StartPosition = System::Windows::Forms::FormStartPosition::CenterScreen;  this->Text = L"Справка";  this->Load += gcnew System::EventHandler(this, &MyForm1::MyForm1\_Load);  (cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox1))->EndInit();  this->ResumeLayout(false);  this->PerformLayout();  }  #pragma endregion  private: System::Void MyForm1\_Load(System::Object^ sender, System::EventArgs^ e) {  }  };  } |