**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 2](#_Toc137073348)

[1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 4](#_Toc137073349)

[1.1 Обзор области разработки 4](#_Toc137073350)

[1.2 Обзор существующих продуктов 6](#_Toc137073351)

[1.2.1 Photomath 6](#_Toc137073352)

[1.2.2 Mathpix 7](#_Toc137073353)

[1.2.3 Microsoft Math Solver 7](#_Toc137073354)

[2. ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ 8](#_Toc137073355)

[2.1 Схема работы системы 8](#_Toc137073356)

[2.2 Архитектура классов программы 8](#_Toc137073357)

[3. ПРАКТИЧЕСКАЯ ЧАСТЬ 11](#_Toc137073358)

[3.1 Разработка программы 11](#_Toc137073359)

[3.2 Тестирование 11](#_Toc137073360)

[ЗАКЛЮЧЕНИЕ 13](#_Toc137073361)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 14](#_Toc137073362)

[Приложение А](#_Toc137073363). [Исходный код программы 15](#_Toc137073364)

# **ВВЕДЕНИЕ**

Темой работы является моделирование решения квадратных уравнений с заданными коэффициентами на языке C++. Основной идеей работы по данной теме является изучение возможностей языка C++ для решения задачи, имеющей практическую составляющую.

Актуальность курсовой работы в первую очередь обусловлена тем, что квадратные уравнения – одни из основных понятий в алгебре. С помощью них решаются, как и элементарные примеры, так и более сложные задачи не только в алгебре, но и в других различных науках, например, физика и экономика. Умение решать квадратные уравнения немного облегчает нахождению ответа.

Цель работы — разработка программы для решения квадратных уравнений с заданными коэффициентами и сравнения методов их решения на языке C++.

Задачи работы:

* Провести библиографический обзор источников по теме работ;
* Отметить существующие решения, подходящие под тематику разработки, выделить их достоинства и недостатки;
* Составить алгоритм работы программы в виде текстового описания и схемы возможных состояний системы;
* Составить иерархию объектов системы в виде архитектуры классов;
* Разработать программу на языке C++;
* Протестировать работу программы;
* Составить отчет по выполненной работе в соответствии с требованиями и защитить его.

Чтобы объективно оценить выполненный проект и провести тестирование программы, сформирована таблица критериев достижения успеха (Таблица 1), содержащая десять основных пунктов и представляющая собой макет технического задания в упрощённом виде.

*Таблица 1 — Критерии достижения успеха*

|  |  |  |
| --- | --- | --- |
| № | Критерий | Значение |
| 1 | Установленный цвет фона оконного приложения и цвет текста оконного приложения для удобства пользователя | Цвет фона оконного приложения — голубой, цвет текста оконного приложения — черный |
| 2 | Возможность решения квадратных уравнений с помощью используемых в программе методов | Наличие функций, вычисляющих корни квадратных уравнений методами, используемыми в программе |
| 3 | Возможность ввода пользователем коэффициентов квадратного уравнения для решения | Наличие соответствующей функции ввода |
| 4 | Возможность завершения работы программы | Наличие функции, закрывающей оконное приложение |
| 5 | Возможность вводить коэффициенты несколько раз, не запуская приложение заново | Наличие соответствующей функции |
| 6 | При вводе нечисловых коэффициентов выводится ошибка о некорректном вводе с предложением ввести коэффициенты повторно | Наличие соответствующей функции |

# **ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**

# **Обзор области разработки**

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

В зависимости от значений, которые могут принимать коэффициенты квадратного уравнения, можно выделить восемь типов квадратных уравнений:

1. . Все три коэффициента равны нулю. Уравнение верно при любом .
2. . Задан только коэффициент . Такое уравнение не имеет корней и является неверным.
3. . Это классическое квадратное уравнение, имеющее от нуля до двух действительных корней.
4. . Левая часть уравнения раскладывается в вид  
   , т. е. уравнение имеет два корня: 0 и .
5. . Уравнение имеет корни при .
6. . Коэффициенты и равны нулю. Единственным корнем уравнения является 0.
7. . При уравнение является уравнением первой степени с единственным корнем .
8. . По аналогии с предыдущим типом это уравнение первой степени, но коэффициент здесь равен нулю. Единственным корнем уравнения является 0.

Для решения квадратных уравнений используются различные методы. Выбор метода зависит от типа квадратного уравнения.

К основным методам решения квадратных уравнений обычно относят следующие:

1. Выделение полного квадрата. Предполагается применение формулы квадрата суммы , а также других формул сокращённого умножения.
2. Разложение на множители. Используется утверждение о том, что произведение множителей равно нулю, если хотя бы один из множителей равен нулю.
3. Решение с помощью дискриминанта. Применяется формула для вычисления дискриминанта . В зависимости от значения дискриминанта определяется количество действительных корней уравнения. Если , то уравнение имеет два действительных корня. Если , то уравнение имеет один действительный корень. Если , то уравнение не имеет действительных корней. Корни уравнения можно вычислить по формуле:
4. Теорема Виета. Согласно этой теореме, сумма корней приведённого квадратного уравнения равна второму коэффициенту, взятому с противоположным знаком, а произведение корней равно свободному члену. В более простом виде это можно записать так:

Есть и другие, менее популярные, но рабочие методы решения квадратных уравнений.

Например, если для квадратного уравнения выполняется условие , то корнями являются –1 и .

Используя ресурсы вычислительной машины, можно находить корни квадратных уравнений методом подбора. Для этого нужно определить интервал, в котором будет выполняться поиск корня, установить шаг и проверять все значения многочлена уравнения в данном интервале с установленным шагом. Для человека такая задача является более трудоёмкой и ресурсозатратной, чем для компьютера.

# **Обзор существующих продуктов**

С появлением и развитием средств вычислительной техники решались различные математические задачи, в том числе и задача по нахождению корней квадратного уравнения. Рассмотрено три программных продукта, функционал которых включает в себя решение квадратных уравнений.

# **Photomath**

Photomath – это приложение, которое помогает учиться решать математические задания. Задания можно вводить как вручную, так и с помощью камеры.

Достоинствами приложения являются:

* возможность ввода заданий с помощью камеры;
* удобный интерфейс;
* возможность работы без доступа к интернету;
* поэтапное решение заданий;
* сохранение истории при вводе заданий;
* способность решать огромное множество заданий за доли секунды;
* практически полное отсутствие ошибок при решении.

Из недостатков можно выделить следующие:

* приложение доступно только на мобильных устройствах;
* приложение не всегда распознает написанное;
* есть определённые ограничения в решениях (например, приложение не воспринимает текстовые условия).

# **Mathpix**

Mathpix – еще одна программа, призванная облегчить жизнь тем, кто не силен в решении сложных математических заданий. Функционал предполагает возможность решения дробей, нахождения производных и интегралов, логарифмов и множество других математических задач.

Достоинствами приложения являются:

* удобный пошаговый решатель;
* есть ввод с камеры;
* может распознавать задания, загруженные PDF файлом.

Из недостатков можно отметить:

* не подходит для заданий по линейной алгебре и матриц;
* обязательная регистрация;
* не подходит для расчетов по геометрии.

# **Microsoft Math Solver**

Microsoft Math Solver – это еще один решатель математических задач, позволяющий получать пошаговые объяснения, создавать графическое представление математических задач, получать различные справки и пояснения по математике.

* доступность как в интернете, так и в виде мобильного приложения;
* возможность ввода заданий с помощью камеры;
* сохранение истории решения примеров.

Среди недостатков:

* необходимость подключения к интернету;
* плохая реализация функции рисования формулы.

# **ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ**

# **Схема работы системы**

Оконное приложение включает в себя алгоритм решения квадратных уравнений и выполнение необходимых для этого манипуляций. Как уже было отмечено в подразделе 1.1, в зависимости от значений коэффициентов квадратное уравнение может принимать различный вид. Визуально это можно выразить с помощью схемы, представленной на Таблице 2

*Таблица 2— Схема типов квадратных уравнений в зависимости от значений коэффициентов*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | B = 0 | | B ! = 0 | |
| A = 0 | 0 = 0 (1) | C = 0 (3) | Bx + C = 0 (5) | Bx = 0 (7) |
| A ! = 0 | Ax^2 = 0 (2) | Ax^2 + C = 0 (4) | Ax^2 + Bx + C = 0 (6) | Ax^2 + Bx = 0 (8) |
|  | C = 0 | C ! = 0 | | C = 0 |

Номера, приведённые в скобках на Таблице 2, соответствуют номерам перечисленных в подразделе 1.1 типов квадратных уравнений.

Алгоритм вычисления корней квадратного уравнения формируется в соответствии с типом конкретного квадратного уравнения.

# **Архитектура классов программы**

В программе реализовано восемь классов, соответствующих типам квадратных уравнений, перечисленных в подразделе 1.1. Далее приведены описания свойств и методов этих классов (Таблицы 3.1-2.8).

*Таблица 3.1 — Класс Type1*

|  |  |
| --- | --- |
| Свойства и методы | Описание |
| string Get\_answer() | Вычисление корней квадратного уравнения |
| string show() | Вывод корней квадратного уравнения |

*Таблица 3.2 — Класс Type2*

|  |  |
| --- | --- |
| Свойства и методы | Описание |
| double A | Коэффициент квадратного уравнения |
| type2(double a1) | Обновляет коэффициент квадратного уравнения |
| string Get\_answer() | Вычисление корней квадратного уравнения |
| *Окончание Таблицы 3.2* |  |
| string show() | Вывод корней квадратного уравнения |

*Таблица 3.3 — Класс Type3*

|  |  |
| --- | --- |
| Свойства и методы | Описание |
| double C | Коэффициент квадратного уравнения |
| type3(double c1) | Обновляет коэффициент квадратного уравнения |
| string Get\_answer() | Вычисление корней квадратного уравнения |
| string show() | Вывод корней квадратного уравнения |

*Таблица 3.4 — Класс Type4*

|  |  |
| --- | --- |
| Свойства и методы | Описание |
| double A | Коэффициент квадратного уравнения |
| double C | Коэффициент квадратного уравнения |
| type4(double a1, double c1) | Обновляет коэффициенты квадратного уравнения |
| string Get\_answer() | Вычисление корней квадратного уравнения |
| string Get\_answer\_podbor() | Вычисление корней квадратного уравнения методом подбора |
| string show() | Вывод корней квадратного уравнения |

*Таблица 3.5 — Класс Type5*

|  |  |
| --- | --- |
| Свойства и методы | Описание |
| double B | Коэффициент квадратного уравнения |
| double C | Коэффициент квадратного уравнения |
| type5(double b1, double c1) | Обновляет коэффициенты квадратного уравнения |
| string Get\_answer() | Вычисление корней квадратного уравнения |
| string Get\_answer\_podbor() | Вычисление корней квадратного уравнения методом подбора |
| string show() | Вывод корней квадратного уравнения |

*Таблица 3.6 — Класс Type6*

|  |  |
| --- | --- |
| Свойства и методы | Описание |
| double A | Коэффициент квадратного уравнения |
| double B | Коэффициент квадратного уравнения |
| double C | Коэффициент квадратного уравнения |
| string Get\_answer() | Вычисление корней квадратного уравнения |
| string Get\_answer\_podbor() | Вычисление корней квадратного уравнения методом подбора |
| string show() | Вывод корней квадратного уравнения |

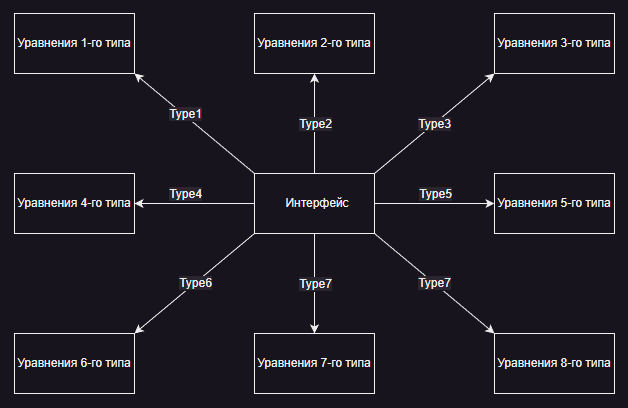
*Таблица 3.7 — Класс Type7*

|  |  |
| --- | --- |
| Свойства и методы | Описание |
| double B | Коэффициент квадратного уравнения |
| type7(double b1) |  |
| string Get\_answer() | Вычисление корней квадратного уравнения |
| string show() | Вывод корней квадратного уравнения |

*Таблица 3.8 — Класс Type8*

|  |  |
| --- | --- |
| Свойства и методы | Описание |
| double A | Коэффициент квадратного уравнения |
| double B | Коэффициент квадратного уравнения |
| type8(double a1, double b1) | Обновляет коэффициенты квадратного уравнения |
| string Get\_answer() | Вычисление корней квадратного уравнения |
| string Get\_answer\_podbor() | Вычисление корней квадратного уравнения методом подбора |
| string show() | Вывод корней квадратного уравнения |

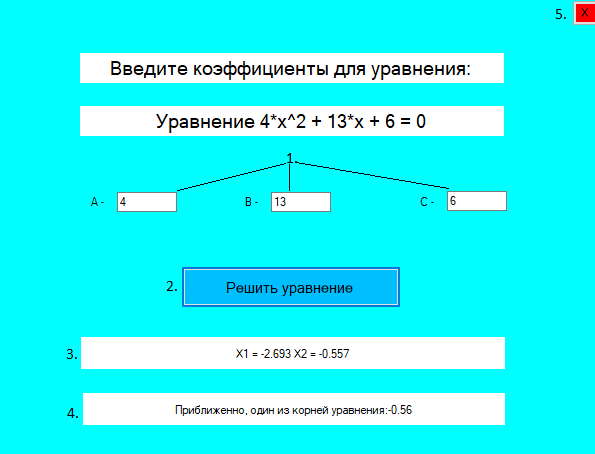
В некоторых описаниях, приведённых выше, уточнялось, что применяется классический метод решения или метод подбора. Под классическим методом решения подразумевается метод, основанный на решении или с помощью дискриминанта, или путём выражения неизвестного. Если в описании метод решения не указан явно, то по умолчанию решение производится с помощью классического метода.

*Рисунок 1 – схема архитектуры классов*

# **ПРАКТИЧЕСКАЯ ЧАСТЬ**

# **Разработка программы**

Работа программы начинается с вывода оконного приложения (Рисунок 1).



**Рисунок 3 – оконное приложение**

Пункт 1. отвечает за ввод коэффициентов A, B и C для решения квадратного уравнения.

Пункт 2. – кнопка, нажав на которую, программа обрабатывает коэффициенты, введенные пользователем.

Затем в поле пункта 3. выводятся корни уравнения.

Поле пункта 4. отвечает за вывод корня, вычисленный методом подбора.

Пункт 5. – кнопка, закрывающая оконное приложение.

# **Тестирование**

Для тестирования программы сформирован журнал испытаний программы (Таблица 4.1)

*Таблица 4.1 – журнал испытания программы*

|  |  |  |  |
| --- | --- | --- | --- |
| № Испытания | Испытание | Эталонный результат | Фактический результат |
| 1 | Запуск программы | Открывается оконное приложение(цвет фона – голубой, цвет текста – черный) | Открывается оконное приложение(цвет фона – голубой, цвет текста – черный) |
| 2 | Ввод нечисловых значений в поля для ввода коэффициентов для решения уравнения | Выводится сообщение о некорректном вводе, предлагается выполнить ввод повторно | Выводится сообщение о некорректном вводе, предлагается выполнить ввод повторно |
| 3 | Корректный ввод коэффициентов для решения уравнения | Отображается введённое уравнение, выводятся корни введённого уравнения | Отображается введённое уравнение, выводятся корни введённого уравнения |
| 4 | Закрытие программы с помощью специальной кнопки | Программа завершает свою работу | Программа завершает свою работу |
| 5 | Ввод коэффициентов несколько раз без закрытия программы | После решения каждого уравнения можно повторно ввести коэффициенты для решения следующего уравнения | После решения каждого уравнения можно повторно ввести коэффициенты для решения следующего уравнения |

Итоговое тестирование показало, что функционал программы реализован и работает корректно.

# **ЗАКЛЮЧЕНИЕ**

В результате выполнения работы были выполнены следующие задачи:

* проведён библиографический обзор источников по теме работы;
* выделено три существующих решения, соответствующих тематике разработки, выделены их достоинства и недостатки;
* составлена иерархия объектов системы в виде архитектуры классов;
* разработана программа для решения квадратных уравнений с заданными коэффициентами и сравнения методов их решения на языке C++;
* проведено тестирование работы программы;
* составлен отчёт по выполненной работе в соответствии с требованиями.

Успешное тестирование программы показывает, что реализация программы выполнена успешно, а цель работы достигнута в полном объёме.

На практическом опыте удалось убедиться в эффективности применения языка C++ и объектно-ориентированного подхода для решения математических задач.

Сравнение классического метода решения и метода подбора для вычисления корней квадратного уравнения показало, что классический метод решения представляется более точным и эффективным. Однако метод подбора имеет свои достоинства: в некоторых случаях он решает задачу за сопоставимое или даже меньшее время, а также не прибегает к использованию сложных математический операций, таких как, например, вычисление квадратного корня. При этом метод подбора во многих случаях даёт результат, достаточно близкий к результату, полученному с помощью классического метода решения.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Давыдова, Н.Е. Различные методы решения квадратных уравнений в 8 классе / Н.Е. Давыдова // Наука и образование: отечественный и зарубежный опыт: Сборник трудов конференции Двадцать первой Международной научно-практической конференции, Белгород, 17 июня 2019 года. — Белгород: ООО ГиК, 2019. — С. 153-155. — URL: https://elibrary.ru/item.asp?id=39195788 (дата обращения: 12.05.2023).
2. Прямостанов, С.М. Метод коэффициентов при решении квадратных уравнений / С.М. Прямостанов, Л.В. Лысогорова // Юный учёный. — 2018. — № 1.1 (15.1). — С. 66-67. — URL: https://moluch.ru/young/archive/15/1165/ (дата обращения: 13.05.2023).
3. Лучшие приложения для решения математики, алгебры, геометрии: ТОП-8 на 2023 год // Андроид-приложения для вашего гаджета: [сайт]. — URL: https://droid-top.ru/obuchenie/prilozheniya-reshenie-matematiki/ (дата обращения: 13.05.2023).
4. Рейтинг лучший приложений для решения задач по фото: [сайт]. – URL: https://dzen.ru/a/Y67BLaq\_PRqa73-U (дата обращения 13.05.2023)
5. Приложение Photomath — калькулятор онлайн в твоём телефоне // Apps4Life: [сайт]. — URL: https://apps4.life/prilozhenie-photomath-kalkuljator-onlajn-v-tvoem-telefone/ (дата обращения: 14.05.2023).

# **Приложение А**

# Исходный код программы

*Листинг А.1 - файл Header.h*

|  |
| --- |
| #pragma once  #include <iostream>  #include <cmath>  #include <string>  using namespace std;  std::string RemoveTrailingZeros(const std::string& input) {  std::string result = input;  size\_t found = result.find\_last\_not\_of('0');  if (found != std::string::npos && result[found] == '.')  found--;  result = result.substr(0, found + 1);  return result;  }  class type1 {  public:  string Get\_answer() {  return ("Ответом является любое значение Х");  }  string show() {  return ("Уравнение 0 = 0");  }  };  class type2 {  private:  double A;  public:  type2(double a1) {  A = a1;  }  string Get\_answer() {  return ("Ответом является Х = 0");  }  string show() {  return ("Уравнение " + (RemoveTrailingZeros(to\_string(A))) + "\*x^2 = 0");  }  };  class type3 {  private:  double C;  public:  type3(double c1) {  C = c1;  }  string Get\_answer() {  return ("Корней нет");  }  string show() {  return (RemoveTrailingZeros(to\_string(C)) + " = 0");  }  };  class type4 {  private:  double A, C;  public:  type4(double a1, double c1) {  A = a1;  C = c1;  }  string Get\_answer() {  double k = ((-1) \* C) / A;  if (k >= 0) {  double x1 = sqrt(k);  double x2 = sqrt(k) \* (-1);  return ("X1 = " + RemoveTrailingZeros(to\_string(x1)) + " X2 = " + RemoveTrailingZeros(to\_string(x2)));  }  else {  return ("Корней нет");  }  }  string Get\_answer\_podbor() {  if ((((-1) \* C) / A) < 0) {  return ("Корней методом подбора нет");  }  else {  double min = abs(0 - A \* (-20) \* (-20) - C);  double x = -20;  for (double i = -20; i < 20.01; i = i + 0.01) {  double temp = abs(0 - A \* (i) \* (i)-C);  if (temp < min) {  min = temp;  x = i;  }  }  return ("Приближенно, один из корней уравнения: " + RemoveTrailingZeros(to\_string(x)));  }  }  string show() {  return ("Уравнение " + RemoveTrailingZeros(to\_string(A)) + "\*x^2 + " + RemoveTrailingZeros(to\_string(C)) + " = 0");  }  };  class type5 {  private:  double B, C;  public:  type5(double b1, double c1) {  B = b1;  C = c1;  }  string Get\_answer() {  return ("Корень уравнения: " + RemoveTrailingZeros(to\_string((-1) \* (C / B))));  }  string Get\_answer\_podbor() {  double min = abs(0 - B \* (-20) - C);  double x = -20;  for (double i = -20; i < 20.01; i = i + 0.01) {  double temp = abs(0 - B \* (i)-C);  if (temp < min) {  min = temp;  x = i;  }  }  return ("Приближенно, один из корней уравнения: " + RemoveTrailingZeros(to\_string(x)));  }  string show() {  return ("Уравнение " + RemoveTrailingZeros(to\_string(B)) + "\*x + " + RemoveTrailingZeros(to\_string(C)) + " = 0");  }  };  class type6 {  private:  double A, B, C;  public:  type6(double a1, double b1, double c1) {  A = a1;  B = b1;  C = c1;  }  string Get\_answer() {  double D = B \* B - 4 \* A \* C;  if (D < 0) {  return ("Уравнение не имеет действительных корней");  }  else {  if (D == 0) {  double x = ((-1) \* B - sqrt(D)) / (2 \* A);  return ("Корень уравнения: " + RemoveTrailingZeros(to\_string(x)));  }  else {  double x1 = ((-1) \* B - sqrt(D)) / (2 \* A);  double x2 = ((-1) \* B + sqrt(D)) / (2 \* A);  return ("X1 = " + RemoveTrailingZeros(to\_string(x1)) + " X2 = " + RemoveTrailingZeros(to\_string(x2)));  }  }  }  string Get\_answer\_podbor() {  double D = B \* B - 4 \* A \* C;  if (D < 0) {  return ("Уравнение не имеет действительных корней");  }  else {  double min = abs(0 - A \* (-20) \* (-20) - B \* (-20) - C);  double x = -20;  for (double i = -20; i < 20.01; i = i + 0.01) {  double temp = abs(0 - A \* (i) \* (i)-B \* i - C);  if (temp < min) {  min = temp;  x = i;  }  }  return ("Приближенно, один из корней уравнения:" + RemoveTrailingZeros(to\_string(x)));  }  }  string show() {  return ("Уравнение " + RemoveTrailingZeros(to\_string(A)) + "\*x^2 + " + RemoveTrailingZeros(to\_string(B)) + "\*x + " + RemoveTrailingZeros(to\_string(C)) + " = 0");  }  };  class type7 {  private:  double B;  public:  type7(double b1) {  B = b1;  }  string Get\_answer() {  return ("Корень уравнения: x = 0");  }  string show() {  return ("Уравнение " + RemoveTrailingZeros(to\_string(B)) + "\*x = 0");  }  };  class type8 {  private:  double A, B;  public:  type8(double a1, double b1) {  A = a1;  B = b1;  }  string Get\_answer() {  double x = ((-1) \* B) / A;  return ("X1 = 0 X2 =" + RemoveTrailingZeros(to\_string(x)));  }  string Get\_answer\_podbor() {  double min = abs(0 - A \* (-20) \* (-20) - B \* (-20));  double x = -20;  for (double i = -20; i < 20.01; i = i + 0.01) {  if (i != 0) {  double temp = abs(0 - A \* i \* i - B \* (i));  if (temp < min) {  min = temp;  x = i;  }  }  }  return ("Приближенно, один из корней уравнения: " + RemoveTrailingZeros(to\_string(x)));  }  string show() {  return ("Уравнение " + RemoveTrailingZeros(to\_string(A)) + "\*x^2 + " + RemoveTrailingZeros(to\_string(B)) + "\*x = 0 ");  }  }; |

*Листинг А.2 - файл CourseWindow.h*

|  |
| --- |
| #pragma once  #include "Header.h"  #include <msclr/marshal\_cppstd.h>  #include <string>  #include <iostream>  #include <regex>  namespace CourseWork {  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 CourseWindow  /// </summary>  public ref class CourseWindow : public System::Windows::Forms::Form  {  public:  CourseWindow(void)  {  InitializeComponent();  //  //TODO: добавьте код конструктора  //  }  protected:  /// <summary>  /// Освободить все используемые ресурсы.  /// </summary>  ~CourseWindow()  {  if (components)  {  delete components;  }  }  protected:  private: System::Windows::Forms::TextBox^ textBox\_A;  private: System::Windows::Forms::TextBox^ textBox\_B;  private: System::Windows::Forms::TextBox^ textBox\_C;  private: System::Windows::Forms::Button^ button1;  private: System::Windows::Forms::Label^ label1;  private: System::Windows::Forms::Label^ label2;  private: System::Windows::Forms::Label^ label3;  private: String^ first\_c = " ";  private: String^ second\_c = " ";  private: String^ third\_c = " ";  private: System::Windows::Forms::Label^ label\_answer;  private: System::Windows::Forms::Label^ label\_main;  private: System::Windows::Forms::Label^ label5;  private: System::Windows::Forms::Label^ label\_podbor;  private: System::Windows::Forms::Button^ button2;  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(CourseWindow::typeid));  this->textBox\_A = (gcnew System::Windows::Forms::TextBox());  this->textBox\_B = (gcnew System::Windows::Forms::TextBox());  this->textBox\_C = (gcnew System::Windows::Forms::TextBox());  this->button1 = (gcnew System::Windows::Forms::Button());  this->label1 = (gcnew System::Windows::Forms::Label());  this->label2 = (gcnew System::Windows::Forms::Label());  this->label3 = (gcnew System::Windows::Forms::Label());  this->label\_answer = (gcnew System::Windows::Forms::Label());  this->label\_main = (gcnew System::Windows::Forms::Label());  this->label5 = (gcnew System::Windows::Forms::Label());  this->label\_podbor = (gcnew System::Windows::Forms::Label());  this->button2 = (gcnew System::Windows::Forms::Button());  this->SuspendLayout();  //  // textBox\_A  //  this->textBox\_A->Location = System::Drawing::Point(119, 192);  this->textBox\_A->Name = L"textBox\_A";  this->textBox\_A->Size = System::Drawing::Size(60, 20);  this->textBox\_A->TabIndex = 2;  this->textBox\_A->TextChanged += gcnew System::EventHandler(this, &CourseWindow::textBox\_A\_TextChanged);  //  // textBox\_B  //  this->textBox\_B->BackColor = System::Drawing::Color::White;  this->textBox\_B->Location = System::Drawing::Point(273, 192);  this->textBox\_B->Name = L"textBox\_B";  this->textBox\_B->Size = System::Drawing::Size(60, 20);  this->textBox\_B->TabIndex = 3;  this->textBox\_B->TextChanged += gcnew System::EventHandler(this, &CourseWindow::textBox\_B\_TextChanged);  //  // textBox\_C  //  this->textBox\_C->BackColor = System::Drawing::Color::White;  this->textBox\_C->Location = System::Drawing::Point(449, 191);  this->textBox\_C->Name = L"textBox\_C";  this->textBox\_C->Size = System::Drawing::Size(60, 20);  this->textBox\_C->TabIndex = 4;  this->textBox\_C->TextChanged += gcnew System::EventHandler(this, &CourseWindow::textBox\_C\_TextChanged);  //  // button1  //  this->button1->BackColor = System::Drawing::Color::DeepSkyBlue;  this->button1->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->button1->Location = System::Drawing::Point(183, 266);  this->button1->Name = L"button1";  this->button1->Size = System::Drawing::Size(220, 42);  this->button1->TabIndex = 5;  this->button1->Text = L"Решить уравнение";  this->button1->UseVisualStyleBackColor = false;  this->button1->Click += gcnew System::EventHandler(this, &CourseWindow::button1\_Click);  //  // label1  //  this->label1->AutoSize = true;  this->label1->Location = System::Drawing::Point(90, 195);  this->label1->Name = L"label1";  this->label1->Size = System::Drawing::Size(23, 13);  this->label1->TabIndex = 7;  this->label1->Text = L"А - ";  this->label1->Click += gcnew System::EventHandler(this, &CourseWindow::label1\_Click);  //  // label2  //  this->label2->AutoSize = true;  this->label2->Location = System::Drawing::Point(244, 195);  this->label2->Name = L"label2";  this->label2->Size = System::Drawing::Size(23, 13);  this->label2->TabIndex = 8;  this->label2->Text = L"В - ";  //  // label3  //  this->label3->AutoSize = true;  this->label3->Location = System::Drawing::Point(420, 195);  this->label3->Name = L"label3";  this->label3->Size = System::Drawing::Size(23, 13);  this->label3->TabIndex = 9;  this->label3->Text = L"С - ";  //  // label\_answer  //  this->label\_answer->BackColor = System::Drawing::Color::White;  this->label\_answer->Location = System::Drawing::Point(83, 337);  this->label\_answer->Name = L"label\_answer";  this->label\_answer->Size = System::Drawing::Size(424, 32);  this->label\_answer->TabIndex = 10;  this->label\_answer->TextAlign = System::Drawing::ContentAlignment::MiddleCenter;  this->label\_answer->Click += gcnew System::EventHandler(this, &CourseWindow::label\_answer\_Click);  //  // label\_main  //  this->label\_main->BackColor = System::Drawing::Color::White;  this->label\_main->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 14.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,  static\_cast<System::Byte>(204)));  this->label\_main->Location = System::Drawing::Point(82, 106);  this->label\_main->Name = L"label\_main";  this->label\_main->Size = System::Drawing::Size(424, 30);  this->label\_main->TabIndex = 11;  this->label\_main->Text = L"Ax^2 + Bx + C = 0";  this->label\_main->TextAlign = System::Drawing::ContentAlignment::MiddleCenter;  //  // label5  //  this->label5->BackColor = System::Drawing::Color::White;  this->label5->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 14.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,  static\_cast<System::Byte>(204)));  this->label5->Location = System::Drawing::Point(82, 53);  this->label5->Name = L"label5";  this->label5->Size = System::Drawing::Size(424, 30);  this->label5->TabIndex = 12;  this->label5->Text = L"Введите коэффициенты для уравнения:";  this->label5->TextAlign = System::Drawing::ContentAlignment::MiddleCenter;  this->label5->Click += gcnew System::EventHandler(this, &CourseWindow::label5\_Click);  //  // label\_podbor  //  this->label\_podbor->BackColor = System::Drawing::Color::White;  this->label\_podbor->Location = System::Drawing::Point(85, 393);  this->label\_podbor->Name = L"label\_podbor";  this->label\_podbor->Size = System::Drawing::Size(422, 32);  this->label\_podbor->TabIndex = 13;  this->label\_podbor->TextAlign = System::Drawing::ContentAlignment::MiddleCenter;  this->label\_podbor->Click += gcnew System::EventHandler(this, &CourseWindow::label\_podbor\_Click);  //  // button2  //  this->button2->BackColor = System::Drawing::Color::Red;  this->button2->FlatAppearance->BorderColor = System::Drawing::Color::Red;  this->button2->FlatAppearance->BorderSize = 0;  this->button2->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,  static\_cast<System::Byte>(204)));  this->button2->ForeColor = System::Drawing::Color::Black;  this->button2->Location = System::Drawing::Point(574, 0);  this->button2->Name = L"button2";  this->button2->Size = System::Drawing::Size(27, 26);  this->button2->TabIndex = 14;  this->button2->Text = L"X";  this->button2->UseVisualStyleBackColor = false;  this->button2->Click += gcnew System::EventHandler(this, &CourseWindow::button2\_Click);  //  // CourseWindow  //  this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);  this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;  this->BackColor = System::Drawing::Color::Aqua;  this->ClientSize = System::Drawing::Size(600, 460);  this->Controls->Add(this->button2);  this->Controls->Add(this->label\_podbor);  this->Controls->Add(this->label5);  this->Controls->Add(this->label\_main);  this->Controls->Add(this->label\_answer);  this->Controls->Add(this->label3);  this->Controls->Add(this->label2);  this->Controls->Add(this->label1);  this->Controls->Add(this->button1);  this->Controls->Add(this->textBox\_C);  this->Controls->Add(this->textBox\_B);  this->Controls->Add(this->textBox\_A);  this->FormBorderStyle = System::Windows::Forms::FormBorderStyle::None;  this->Icon = (cli::safe\_cast<System::Drawing::Icon^>(resources->GetObject(L"$this.Icon")));  this->Name = L"CourseWindow";  this->StartPosition = System::Windows::Forms::FormStartPosition::CenterScreen;  this->Text = L"Вычисления корней квадратного уравнения";  this->ResumeLayout(false);  this->PerformLayout();  }  #pragma endregion  //проверка на запрещенные символы  bool IsStringNumeric(String^ input)  {  std::string strInput = msclr::interop::marshal\_as<std::string>(input);  try  {  // Преобразование строки в число  std::stod(strInput);  return true;  }  catch (const std::exception&)  {  return false;  }  }  //проверка, является ли строка только нулями  bool IsStringAllZeros(String^ text) {  if (text->Length > 1) {  for (int i = 0; i < text->Length; i++) {  if (text[i] != '0') {  return false;  }  }  return true;  }  return false;  }  //преобразование double в string  String^ ConvertDoubleToString(double value)  {  // Преобразование double в String^ с помощью метода System::Convert::ToString()  String^ str = Convert::ToString(value);  return str;  }  //преобразование String^ в string  String^ ConvertToManagedString(const std::string& nativeString)  {  return msclr::interop::marshal\_as<String^>(nativeString);  }  //преобразование string в double  double ConvertStringToDouble(String^ str)  {  // Преобразование System::String^ в std::string  std::string stdStr = msclr::interop::marshal\_as<std::string>(str);  // Преобразование std::string в double  double result = std::stod(stdStr);  return result;  }  private: System::Void textBox\_A\_TextChanged(System::Object^ sender, System::EventArgs^ e) {  first\_c = textBox\_A->Text;  }  private: System::Void textBox\_B\_TextChanged(System::Object^ sender, System::EventArgs^ e) {  second\_c = textBox\_B->Text;  }  private: System::Void textBox\_C\_TextChanged(System::Object^ sender, System::EventArgs^ e) {  third\_c = textBox\_C->Text;  }  private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e) {  if (IsStringNumeric(first\_c) && IsStringNumeric(second\_c) && IsStringNumeric(third\_c)) {  if (IsStringAllZeros(first\_c) || IsStringAllZeros(second\_c) || IsStringAllZeros(third\_c)) {  MessageBox::Show("Неправильно введены коэффициенты!", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);  }  //1  if (first\_c == "0" && second\_c == "0" && third\_c == "0") {  type1 eq1;  label\_answer->Text = ConvertToManagedString(eq1.Get\_answer());  label\_podbor->Text = "Невозможно решить методом подбора";  label\_main->Text = ConvertToManagedString(eq1.show());  }  //2  else if (first\_c != "0" && second\_c == "0" && third\_c == "0") {  type2 eq2(Convert::ToInt32(first\_c));  label\_answer->Text = ConvertToManagedString(eq2.Get\_answer());  label\_podbor->Text = "Невозможно решить методом подбора";  label\_main->Text = ConvertToManagedString(eq2.show());  }  //3  else if (first\_c == "0" && second\_c == "0" && third\_c != "0") {  type3 eq3(Convert::ToInt32(third\_c));;  label\_answer->Text = ConvertToManagedString(eq3.Get\_answer());  label\_podbor->Text = "Невозможно решить методом подбора";  label\_main->Text = ConvertToManagedString(eq3.show());  }  //4  else if (first\_c != "0" && second\_c == "0" && third\_c != "0") {  type4 eq4(Convert::ToInt32(first\_c), Convert::ToInt32(third\_c));  label\_answer->Text = ConvertToManagedString(eq4.Get\_answer());  label\_podbor->Text = ConvertToManagedString(eq4.Get\_answer\_podbor());  label\_main->Text = ConvertToManagedString(eq4.show());  }  //5  else if (first\_c == "0" && second\_c != "0" && third\_c != "0") {  type5 eq5(Convert::ToInt32(second\_c), Convert::ToInt32(third\_c));  label\_answer->Text = ConvertToManagedString(eq5.Get\_answer());  label\_podbor->Text = ConvertToManagedString(eq5.Get\_answer\_podbor());  label\_main->Text = ConvertToManagedString(eq5.show());  }  //6  else if (first\_c != "0" && second\_c != "0" && third\_c != "0") {  type6 eq6(Convert::ToInt32(first\_c), Convert::ToInt32(second\_c), Convert::ToInt32(third\_c));  label\_answer->Text = ConvertToManagedString(eq6.Get\_answer());  label\_podbor->Text = ConvertToManagedString(eq6.Get\_answer\_podbor());  label\_main->Text = ConvertToManagedString(eq6.show());  }  //7  else if (first\_c == "0" && second\_c != "0" && third\_c == "0") {  type7 eq7(Convert::ToInt32(second\_c));  label\_answer->Text = ConvertToManagedString(eq7.Get\_answer());  label\_podbor->Text = "Невозможно решить методом подбора";  label\_main->Text = ConvertToManagedString(eq7.show());  }  //8  else if (first\_c != "0" && second\_c != "0" && third\_c == "0") {  type8 eq8(Convert::ToInt32(first\_c), Convert::ToInt32(second\_c));  label\_answer->Text = ConvertToManagedString(eq8.Get\_answer());  label\_podbor->Text = ConvertToManagedString(eq8.Get\_answer\_podbor());  label\_main->Text = ConvertToManagedString(eq8.show());  }  }  else {  MessageBox::Show("Неправильно введены коэффициенты! Введите коэффициенты повторно", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);  }  }  private: System::Void label5\_Click(System::Object^ sender, System::EventArgs^ e) {  }  private: System::Void label1\_Click(System::Object^ sender, System::EventArgs^ e) {  }  private: System::Void button2\_Click(System::Object^ sender, System::EventArgs^ e) {  this->Close();  }  private: System::Void label\_answer\_Click(System::Object^ sender, System::EventArgs^ e) {  }  private: System::Void label\_podbor\_Click(System::Object^ sender, System::EventArgs^ e) {  }  };  } |

*Листинг А.3 - файл CourseWindow.cpp*

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