

Министерство образования Российской Федерации
Пензенский государственный университет
Кафедра «Вычислительная техника»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе
по курсу «Программирование»
на тему «Инженерный калькулятор»

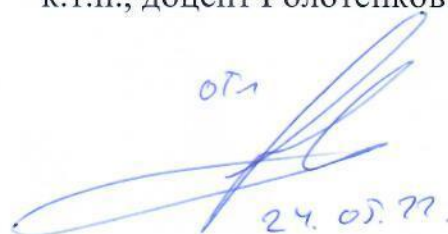
Выполнил:

студент группы 21ВВ1

Аляев А.О.

Принял:

к.т.н., доцент Голотенков Н.О.

от

24.05.22.

Пенза 2022

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет Вычислительной техники

Кафедра "Вычислительная техника"

"УТВЕРЖДАЮ"

Зав. кафедрой ВТ



М.А. Митрохин

« ____ » ____ 20__ г

ЗАДАНИЕ

на курсовое проектирование по курсу

Программирование

Студенту Аляеву Александру Олеговичу Группа 21ВВ1

Тема проекта: *Разработка программы сложной структуры методом нисходящего программирования.*

Исходные данные (технические требования) на проектирование

*Приложение инженерный калькулятор.
Функции: сложение, вычитание, умножение, деление,
квадратный корень, тригонометрические ф-ции.
Возможность сохранить результат в файл.
Вставка на языке Ассемблера.*

Обязательные требования к программе:

1. Многомодульность.
2. Использование сложных типов данных – структур, списков, файлов.
3. Режим работы видеосистемы – текстовый / графический.
4. Устройство ввода информации – клавиатура, мышь.
5. Пользовательский интерфейс должен быть построен на основе меню и панели инструментов.
6. Наличие заставки.
7. Операционная система MS Windows.
8. Язык программирования – Си и Ассемблер.
9. Среда разработки ПО – Microsoft Visual Studio.

Объем работы по курсу

1. Расчетная часть

Разработка программы.

2. Графическая часть

Схема данных, схема ресурсов системы, схема работы системы, иерархическая структура программы, схема взаимодействия программы.

3. Экспериментальная часть

Отладка программы.

Срок выполнения проекта по разделам

В соответствии с графиком.

Дата выдачи задания “22” 02.2022

Дата защиты проекта “ ”

Руководитель _____

Задание получил “22” 02 2022г.

Студент Ляев А.О. ф

Содержание

Введение.....	5
Постановка задачи	6
Выбор решения.....	7
Описание разработки программы	8
Отладка и тестирование.....	9
Описание программы	9
Руководство пользователя.....	20
Список используемых источников.....	27
Приложение А. Листинг программы	28
Приложение В. Результаты работы программы	50
Приложение С. Файл с результатами вычислений	51

ВВЕДЕНИЕ

На сегодняшний день без точных расчётов не обходится практически ни одна инженерная наука, будь то расчёт межпланетной орбиты или офисные расчёты. Калькуляторы во многом упрощают эти задачи. Именно они могут точно и главное быстро произвести операции любой сложности.

Человек создал первый прообраз аналогового калькулятора – Абак ещё в 6 веке, тогда он представлял из себя доску с бороздками. Данное устройство было популярно вплоть до 19 века. Первым цифровым калькулятором считается калькулятор 14-А фирмы Casio. Он мог выполнять действия над 14 разрядными десятичными цифрами.

В наше время жизнь без вычислительной техники невозможна. Несмотря на то, что эти механизмы создавались для сложных вычислений в инженерной области, медицине, бухгалтерии и финансах, они нашли свое широкое применение и в школе, университете и повседневной жизни, где удачно используются и сейчас.

Данная программа является вычислительной системой и позволяет пользователю выполнять основные математические операции над числами, а также просматривать историю операций.

ПОСТАНОВКА ЗАДАЧИ

Необходимо разработать программу калькулятор для проведения арифметических операций. Программа должна иметь интуитивно понятный интерфейс для пользователя. Необходимо определить набор арифметических операций, обработать возможные исключения в работе программы, изучить функции работы с файлами для сохранения истории операций.

Многомодульность программы. Программа должна быть поделена на логические модули. Это упростит поиск ошибок при отладке и тестировании программы, а также позволит легко расширять функционал программы.

Использование сложных типов данных – массивов, структур, файлов. Это необходимо для более простой и интуитивной обработки данных в коде программы.

Режим работы видеосистемы – текстовый/графический. Для начала необходимо определиться с типом интерфейса и с элементами управления, затем необходимо изучить способы их реализации.

Устройство ввода-вывода – клавиатура и мышь. Необходимо различать и идентифицировать действия, произведенные с их помощью, это облегчит использование программы.

Интерфейс должен быть создан в графической среде. Это необходимо для создания интуитивно понятного интерфейса для пользователя.

Заставка необходима для того, чтобы пользователь, запустивший программу, смог получить достаточную информацию о ней.

ВЫБОР РЕШЕНИЯ

При запуске программы открывается приветственный экран. Далее появляется графический интерфейс с кнопками:

1. Кнопки выбора цифр от 0 до 9.
2. Кнопки арифметических операций
 - 2.1) “+”-сложение.
 - 2.2) “-”-вычитание.
 - 2.3) “*”-умножение.
 - 2.4) “/”-деление.
 - 2.5) “√”-квадратный корень.
3. Кнопки тригонометрических операций
 - 3.1) “sin”-Синус.
 - 3.1) “cos”-Косинус.
 - 3.1) “tg”-Тангенс.
 - 3.1) “ctg”-Котангенс.
4. Служебные кнопки
 - 4.1) “☒”-Удаление справа. Удаляет крайнюю правую цифру.
 - 4.2) “C”-Удаление строки. Удаляет всю введенную строку.
 - 4.3) “History”-История. Открывает файл с историей операций.
 - 4.4) “,”-Запятая. Ставит в строку символ запятой, для работы с дробными числами.
 - 4.5) “=”-Вычисление. При нажатии вычисляет результат операции над введенными числами.
 - 4.6) “info”-Информация. При нажатии открывает окно с информацией о приложении.

ОПИСАНИЕ РАЗРАБОТКИ ПРОГРАММЫ

Разработанная программа состоит из нескольких модулей, которые реализовывались в следующем порядке:

1. Kursovaya.cpp – Основной файл программы.
2. Form1.h – Графический интерфейс программы. Содержит код вычисления некоторых функций.
3. Picture.h – файл заставки приложения
4. Plus.cpp – файл, с кодом выполнения операции сложения.
5. Plus.h – заголовочный файл для plus.cpp.
6. minus.cpp – файл, с кодом выполнения операции вычитания.
7. minus.h – заголовочный файл для minus.cpp.
8. division.cpp - файл, с кодом выполнения операции деления.
9. division.h - заголовочный файл для division.cpp.
10. FileAdd.cpp – файл, с кодом сохранения проведенных вычислений в текстовый документ.
11. FileAdd.h - заголовочный файл для FileAdd.cpp.
12. multiplication.cpp - файл, с кодом выполнения операции умножения.
13. multiplication.h - заголовочный файл для multiplication.cpp.

По порядку реализации модулей можно понять, что использовался нисходящий подход реализации программы. Этот метод основан на том, что 10 сначала проектируются основные компоненты программы, а затем уже дорабатываются ее мелкие детали. Это позволяет упростить ее тестирование, так как каждый модуль тестируется сразу же после его реализации, что помогает в создании более структурированных программ.

ОТЛАДКА И ТЕСТИРОВАНИЕ

В качестве среды разработки была выбрана программа Microsoft VisualStudio 2010. Программа обладает всеми средствами необходимыми при разработке и отладке программы. Для отладки использовались несколько возможностей VisualStudio: точка останова, трассировка, анализ содержимого переменных.

Тестирование проводилось во время разработки и также после завершения разработки. В ходе нее было выявлено определённое количество исключений, связанных с вычислительными операциями, заставкой, а также работой функций графического интерфейса.

ОПИСАНИЕ ПРОГРАММЫ

Приложение Kursova.exe является основным модулем программы. При запуске программы происходит вывод заставки, после чего пользователю представляется графический интерфейс приложения с кнопками ввода цифр, и кнопками математических операций. Описание состояний программы выполнено в таблице ниже.

Действия, вызывающие события	Действия пользователя	Действие программы
Нажатие на кнопку “1”, ввод с клавиатуры цифры “1”.	Нажатие на кнопку “1”, ввод с клавиатуры цифры “1”.	Выводит на экран цифру “1”.
Нажатие на кнопку “2”, ввод с клавиатуры цифры “2”.	Нажатие на кнопку “2”, ввод с клавиатуры цифры “2”.	Выводит на экран цифру “2”.
Нажатие на кнопку “3”, ввод с клавиатуры цифры “3”.	Нажатие на кнопку “3”, ввод с клавиатуры цифры “3”.	Выводит на экран цифру “3”.

Нажатие на кнопку “4”, ввод с клавиатуры цифры “4”.	Нажатие на кнопку “4”, ввод с клавиатуры цифры “4”.	Выводит на экран цифру “4”.
Нажатие на кнопку “5”, ввод с клавиатуры цифры “5”.	Нажатие на кнопку “5”, ввод с клавиатуры цифры “5”.	Выводит на экран цифру “5”.
Нажатие на кнопку “6”, ввод с клавиатуры цифры “6”.	Нажатие на кнопку “6”, ввод с клавиатуры цифры “6”.	Выводит на экран цифру “6”.
Нажатие на кнопку “7”, ввод с клавиатуры цифры “7”.	Нажатие на кнопку “7”, ввод с клавиатуры цифры “7”.	Выводит на экран цифру “7”.
Нажатие на кнопку “8”, ввод с клавиатуры цифры “8”.	Нажатие на кнопку “8”, ввод с клавиатуры цифры “8”.	Выводит на экран цифру “8”.
Нажатие на кнопку “9”, ввод с клавиатуры цифры “9”.	Нажатие на кнопку “9”, ввод с клавиатуры цифры “9”.	Выводит на экран цифру “9”.
Нажатие на кнопку “0”, ввод с клавиатуры цифры “0”.	Нажатие на кнопку “0”, ввод с клавиатуры цифры “0”.	Выводит на экран цифру “0”.
Нажатие на кнопку “,”, ввод с клавиатуры символа “,”.	Нажатие на кнопку “,”, ввод с клавиатуры символа “,”.	Выводит на экран символ “,”.

Таблица 1.

Действия, вызывающие события	Действия пользователя	Действие программы
Ввод в поле 1-го слагаемого и нажатие на кнопку “+”, ввод 2-го слагаемого, нажатие на кнопку “=”.	Ввод первого слагаемого, нажатие кнопки “+”, ввод второго слагаемого, нажатие на кнопку “=”.	Осуществляет операцию сложения.
Ввод в поле уменьшаемого и нажатие на кнопку “-”, ввод вычитаемого, нажатие на кнопку “=”.	Ввод уменьшаемого, нажатие кнопки “-”, ввод вычитаемого, нажатие на кнопку “=”.	Осуществляет операцию вычитания.
Ввод в поле 1-го множителя и нажатие на кнопку “*”, ввод 2-го множителя, нажатие на кнопку “=”.	Ввод первого множителя, нажатие кнопки “*”, ввод второго множителя, нажатие на кнопку “=”.	Осуществляет операцию умножения.
Ввод в поле делимого и нажатие на кнопку “/”, ввод делителя, нажатие на кнопку “=”.	Ввод делимого, нажатие кнопки “/”, ввод делителя, нажатие на кнопку “=”.	Осуществляет операцию деления.
Ввод в поле числа и нажатие на кнопку “√”.	Ввод в поле числа и нажатие на кнопку “√”.	Осуществляет вычисление квадратного корня.
Ввод в поле числа и нажатие на кнопку “sin”.	Ввод в поле числа и нажатие на кнопку “sin”.	Осуществляет вычисление синуса числа. (в радианах)

Ввод в поле числа и нажатие на кнопку “cos”.	Ввод в поле числа и нажатие на кнопку “cos”.	Осуществляет вычисление косинуса числа. (в радианах)
Ввод в поле числа и нажатие на кнопку “tan”.	Ввод в поле числа и нажатие на кнопку “tan”.	Осуществляет вычисление тангенса числа. (в радианах)
Ввод в поле числа и нажатие на кнопку “cot”.	Ввод в поле числа и нажатие на кнопку “cot”.	Осуществляет вычисление котангенса числа. (в радианах)

Таблица 2.

Действия, вызывающие события	Действия пользователя	Действие программы
Нажатие на кнопку “  ”.	Нажатие на кнопку “  ”.	Удаление последней цифры справа.
Нажатие на кнопку “C”.	Нажатие на кнопку “C”.	Удаление всего числа
Нажатие на кнопку “History”.	Нажатие на кнопку “History”.	Открытие текстового документа с историей вычислений.

Таблица 3.

На рисунке 1 представлена схема взаимодействия программы.

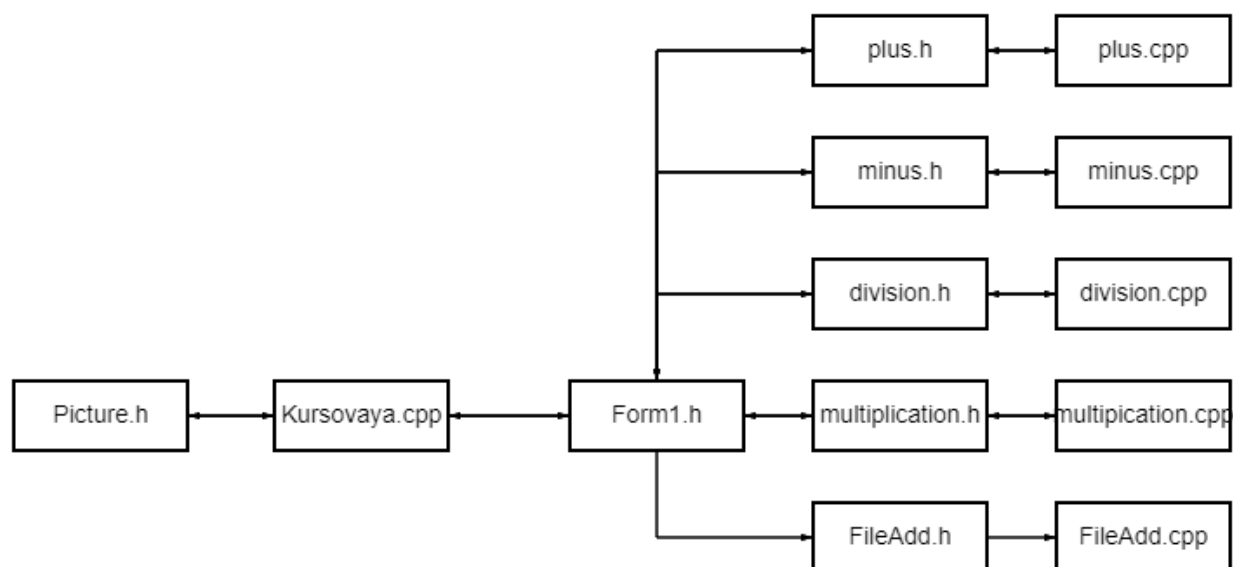


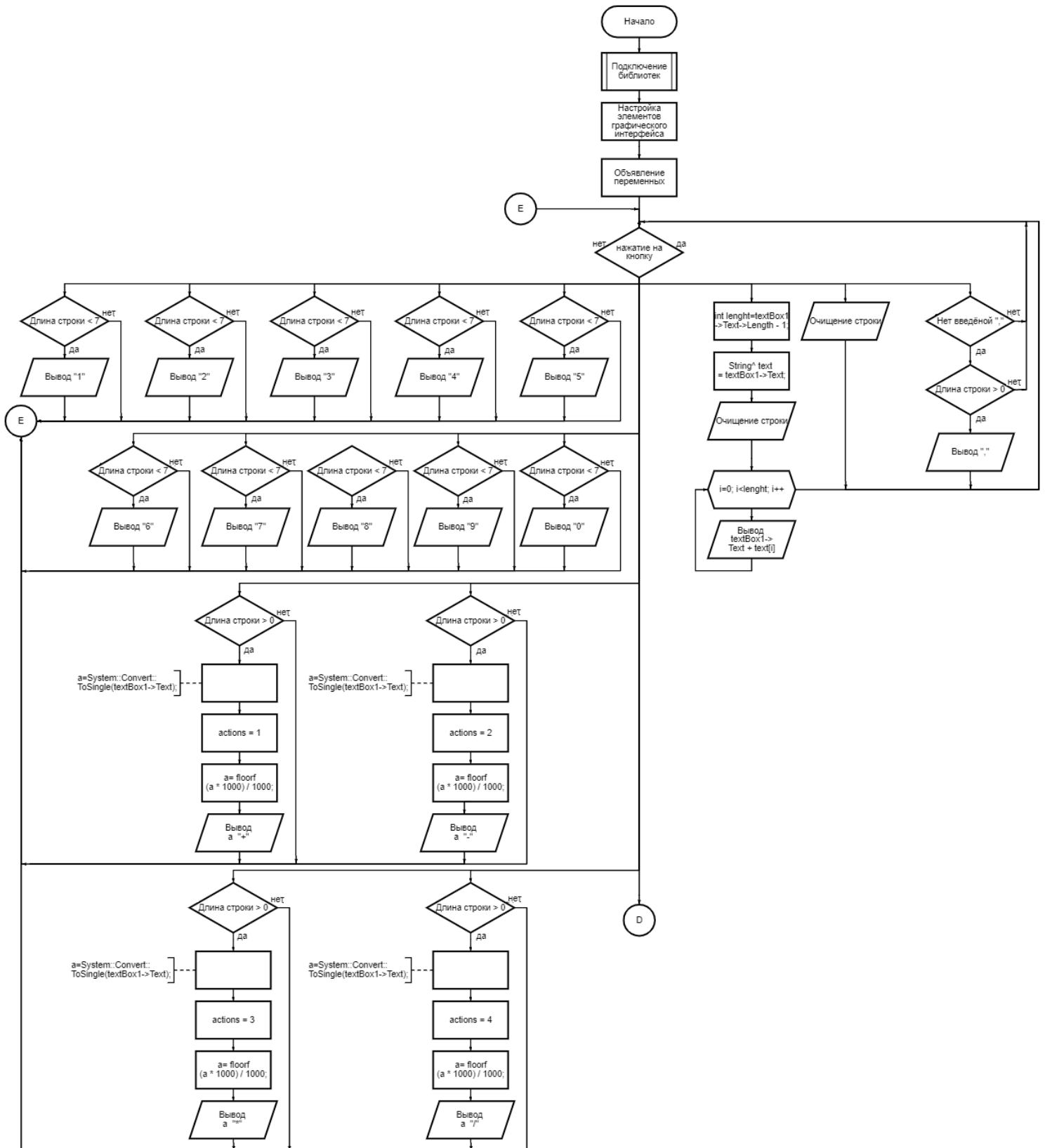
Рисунок 1 – схема взаимодействия программы

На рисунке 2 представлена схема данных



Рисунок 2 – схема данных

Схема программы. Класс Form1.



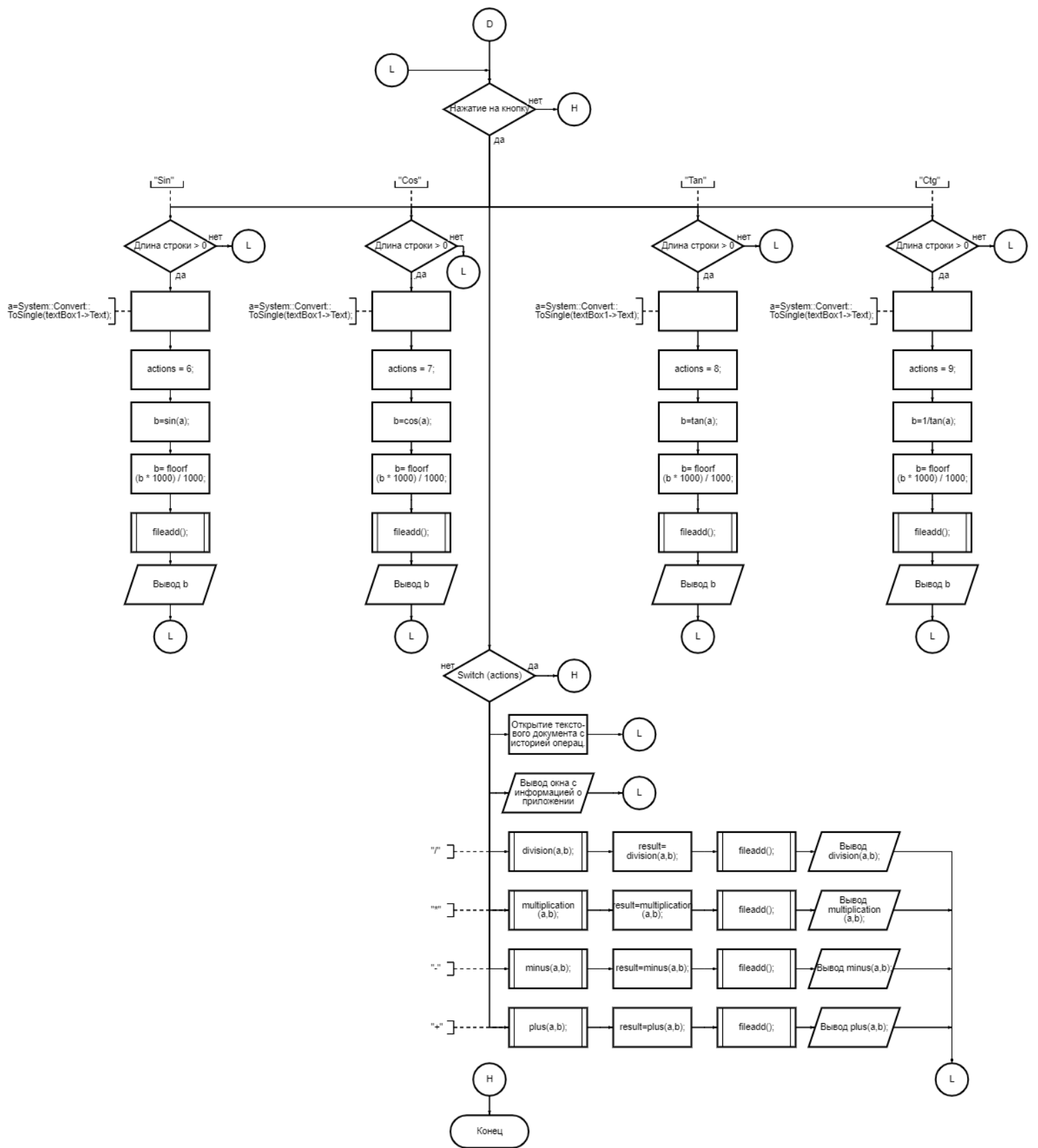


Рисунок – класс Form1

Схема программы. Функция plus.

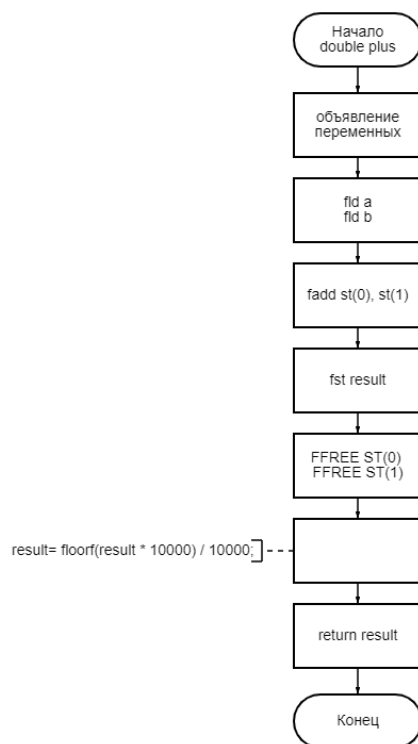


Рисунок – функция plus

Схема программы. Функция minus.

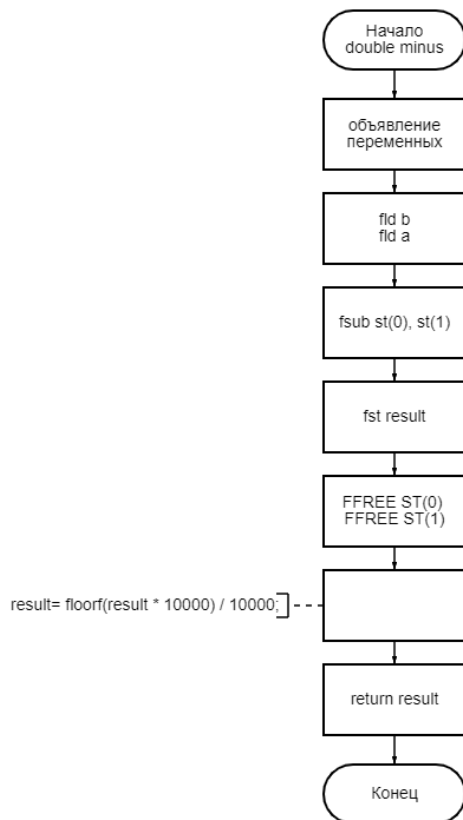


Рисунок – функция minus

Схема программы. Функция division.

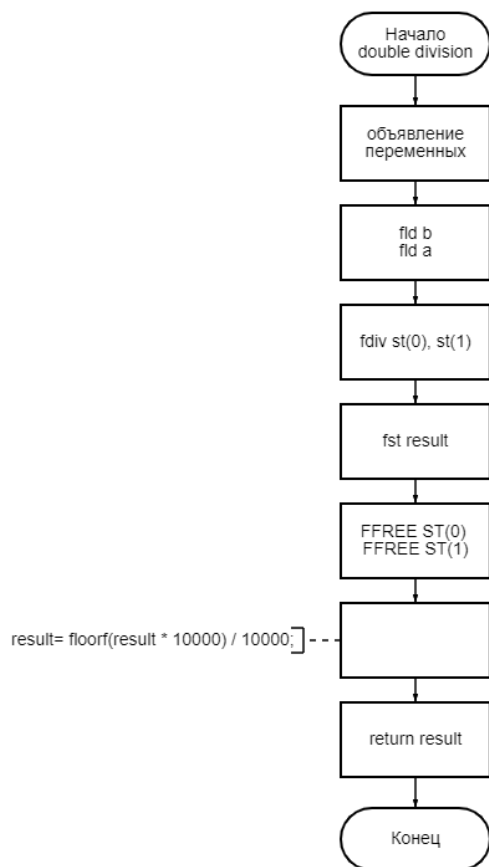


Рисунок – функция division

Схема программы. Функция multiplication.

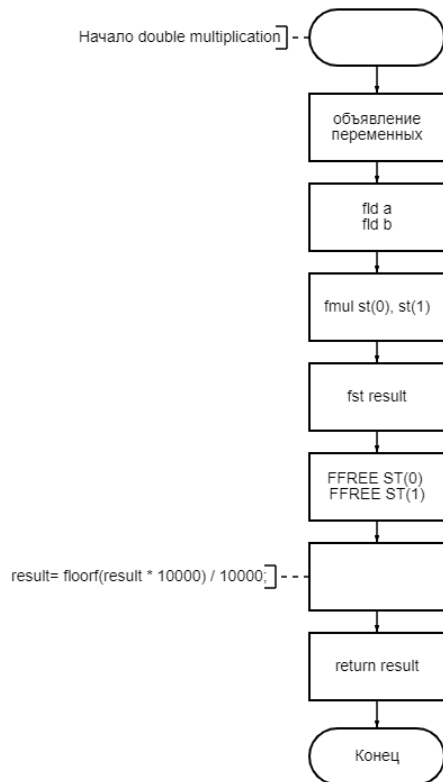


Рисунок – функция multiplication

Схема программы. Функция FileAdd.

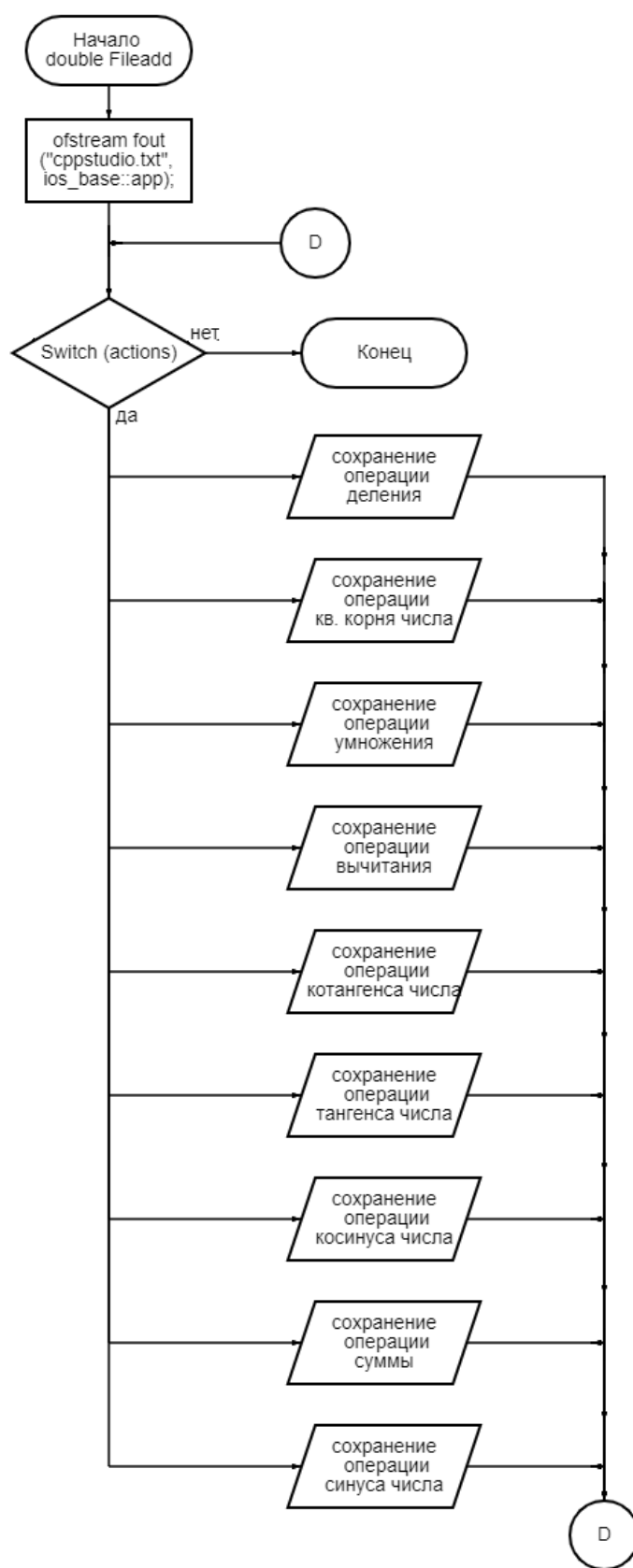


Рисунок – функция FileAdd

Схема программы. Класс Picture.

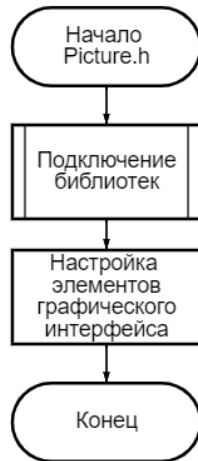


Рисунок – класс Picture

Схема программы. Функция Kursovaya.

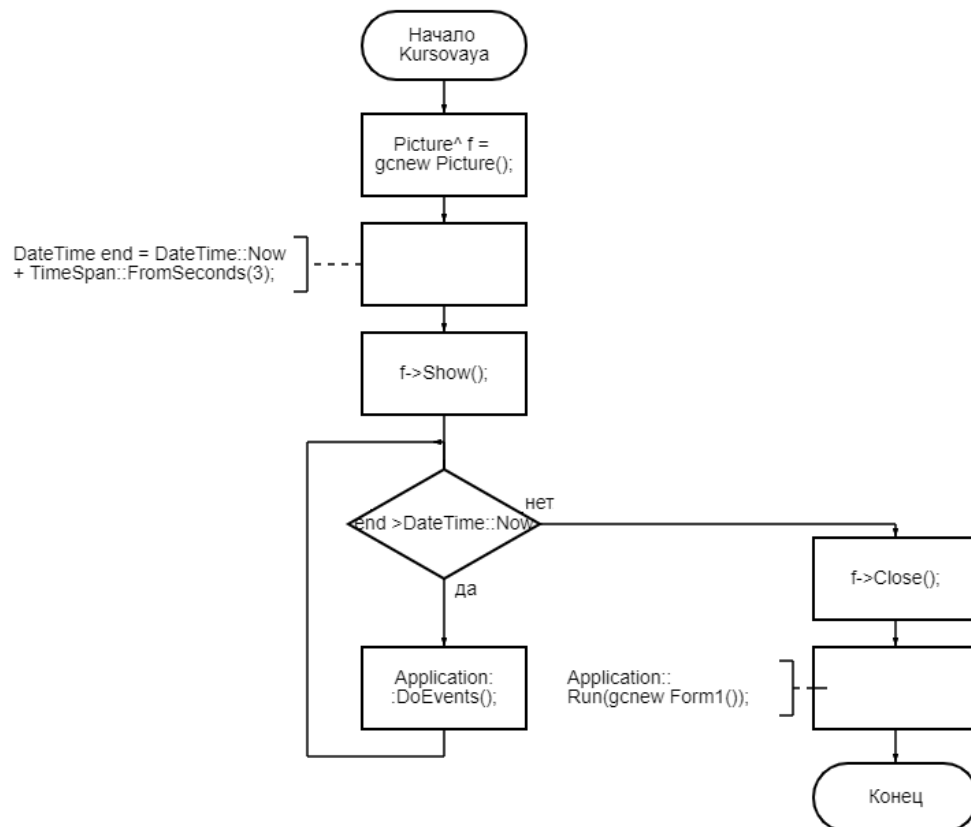


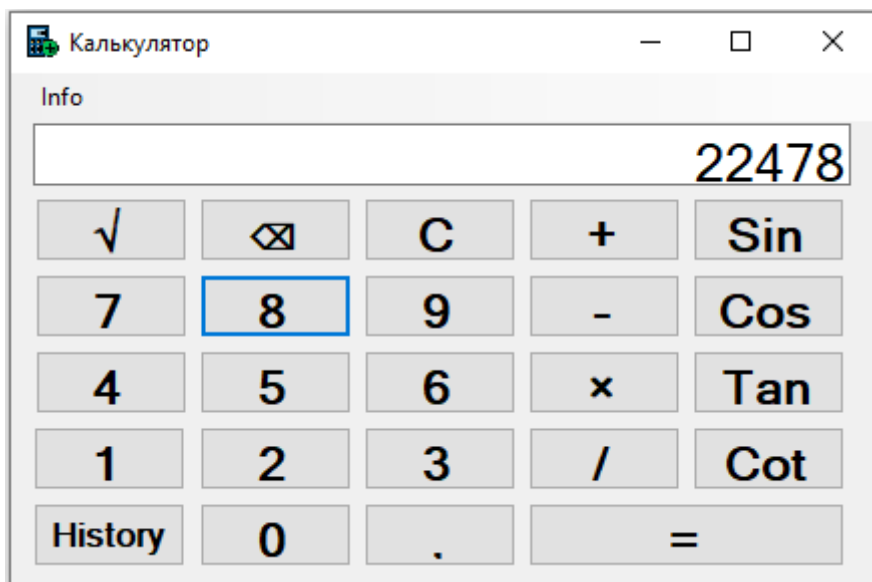
Рисунок – функция Kursovaya

РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

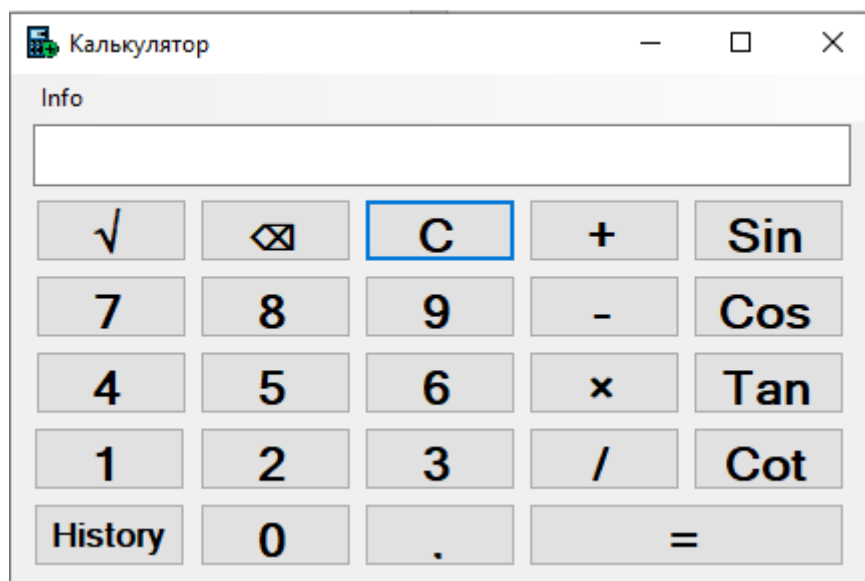
Программа предназначена для проведения простых математических операций. Программа имеет интуитивно понятный интерфейс и поддерживает такие операции как: сложение, вычитание, умножение, деление, тригонометрические функции, квадратный корень, просмотр истории, очистка последнего символа и строки целиком, просмотр информации о приложении.

После открытия приложения на экране появляется заставка, после которой открывается само приложение. Для работы ним пользователю предлагается использовать мышь или клавиатуру.

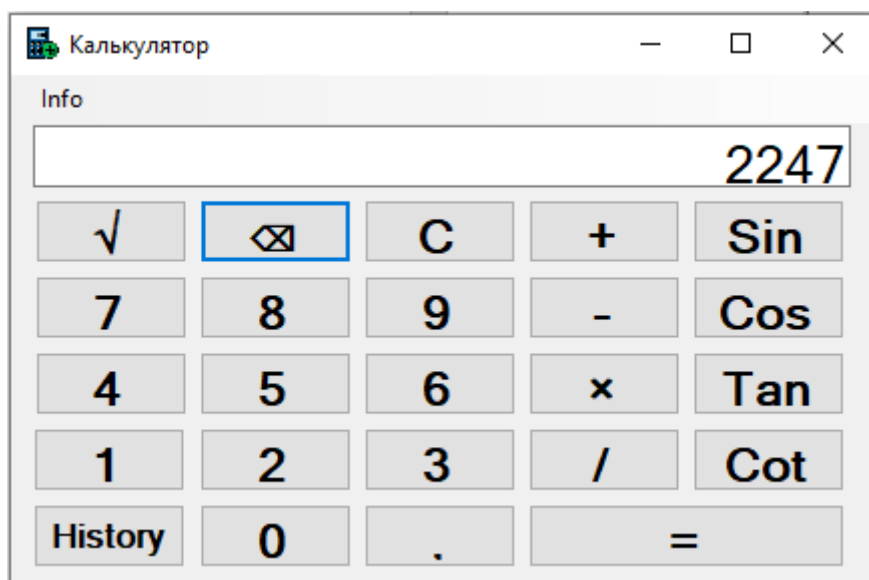
После нажатия на соответствующие кнопки цифр на интерфейсе, они появятся на текстовом поле приложения.



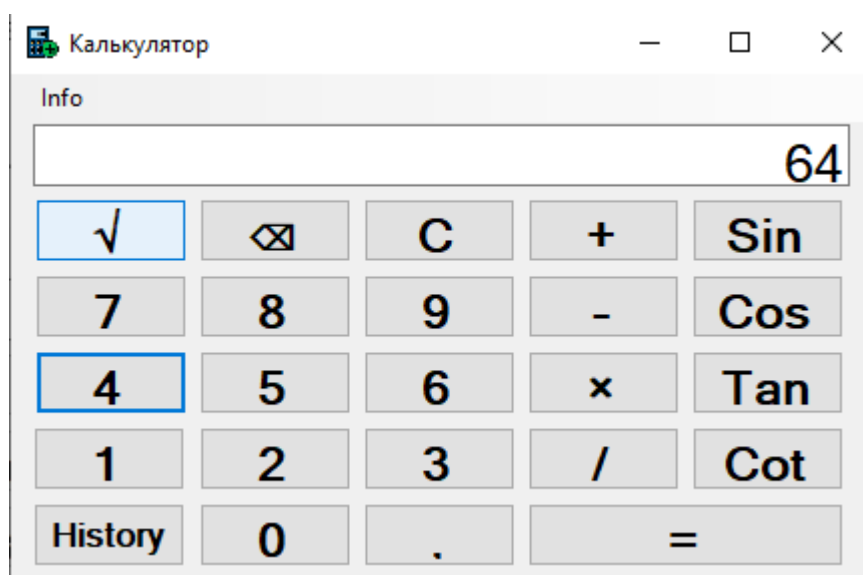
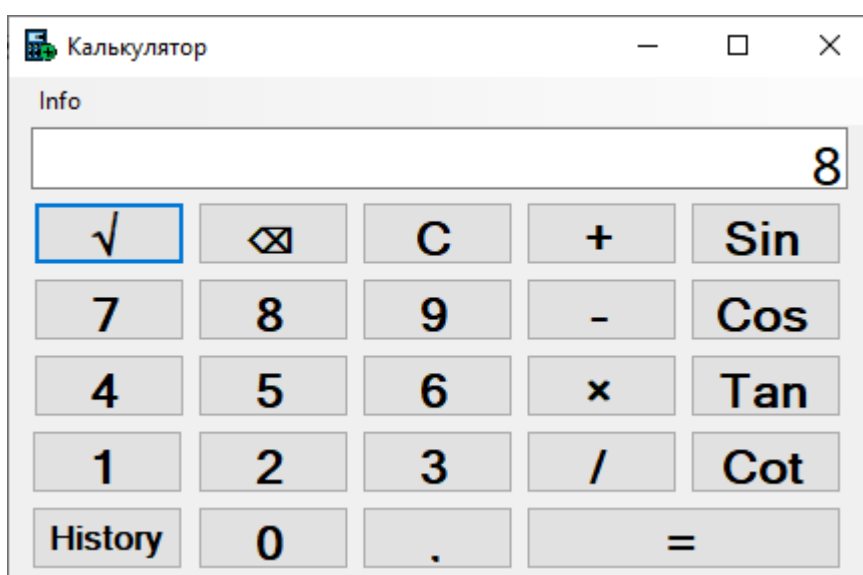
Для удаления строки целиком пользователю необходимо нажать на кнопку “C”.



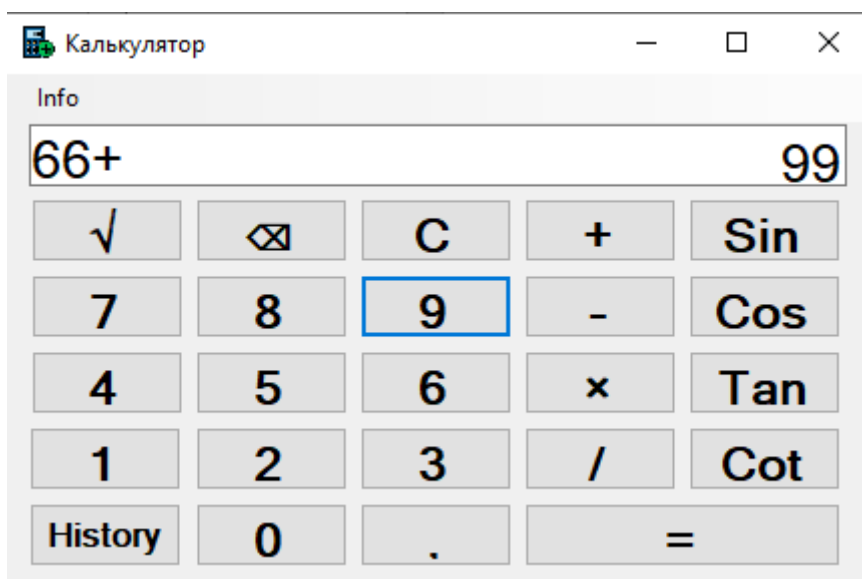
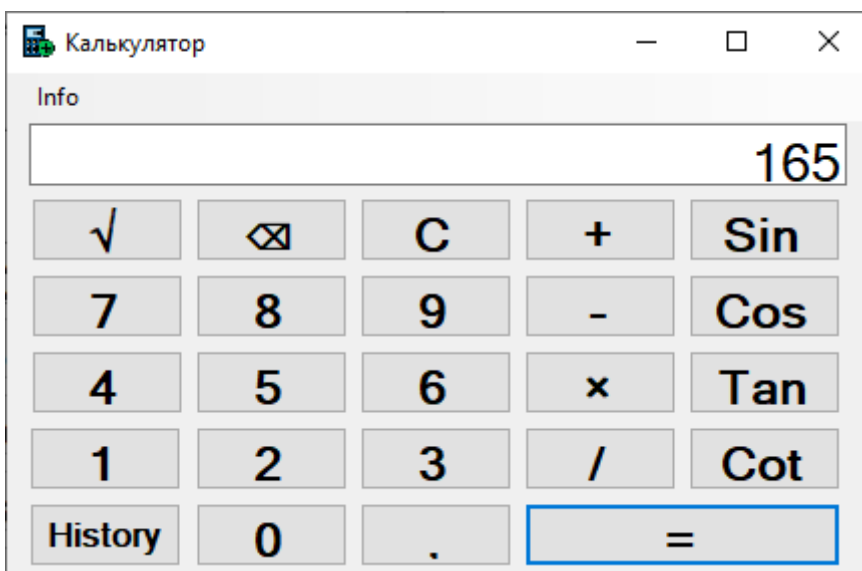
Для удаления последнего правого символа пользователь должен нажать на кнопку “ $\boxtimes$ ”.



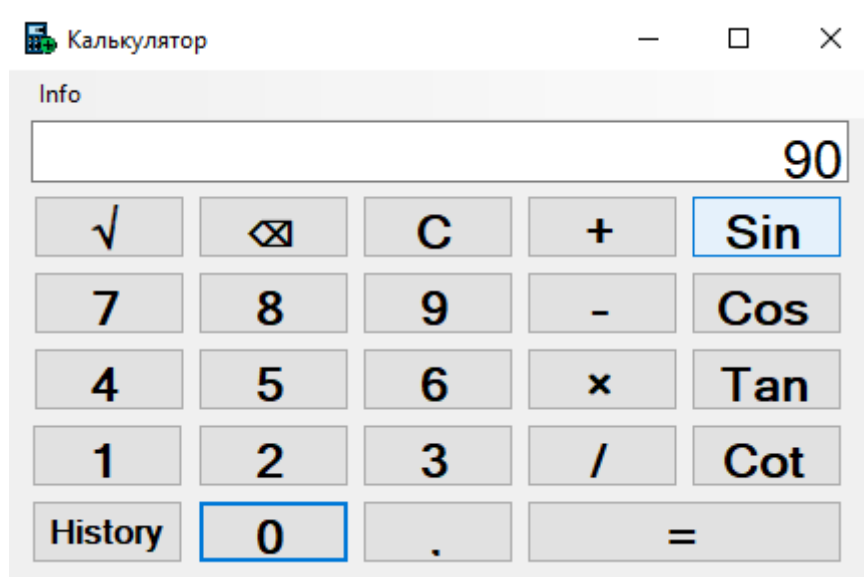
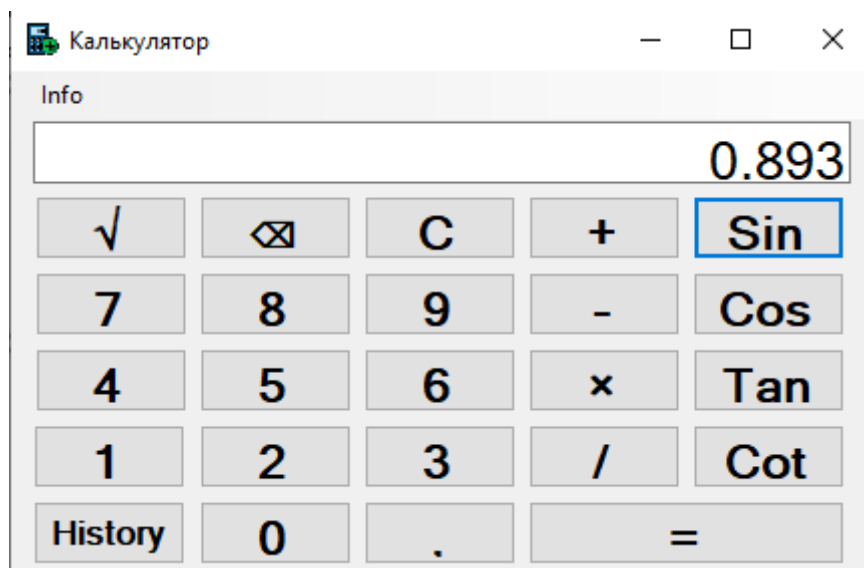
Для вычисления квадратного корня числа ($\sqrt{}$) пользователю необходимо ввести нужное число и нажать на соответствующую кнопку. Если число не будет введено, то действие кнопки заблокируется.



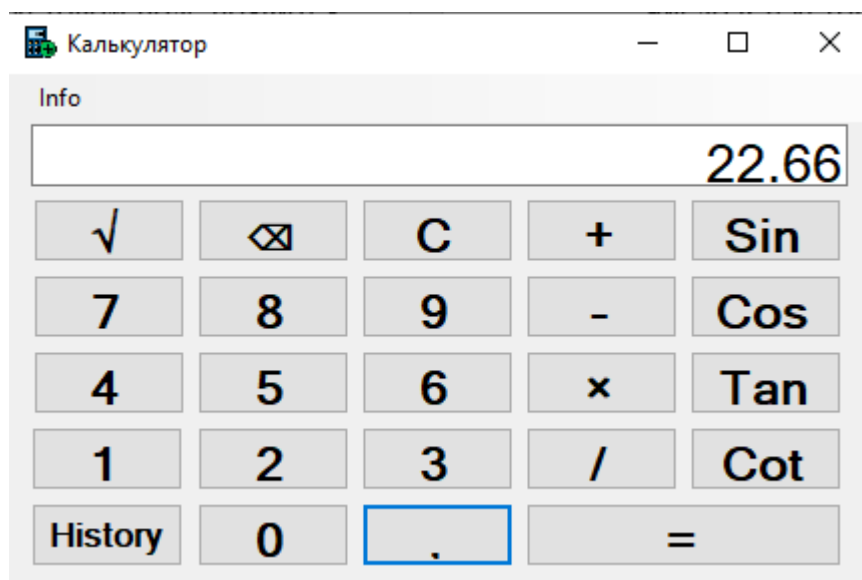
Для вычисления простейших математических операций (+, -, /, *) пользователю необходимо ввести первое число, затем нажать на кнопку необходимого действия. После нажатия, первое число переместится в левый край текстового поля. Затем, необходимо ввести второе число и нажать на кнопку проведения операции (“=”). В текстовом поле появится итоговый результат. Если первое число не будет введено, то кнопка будет заблокирована.



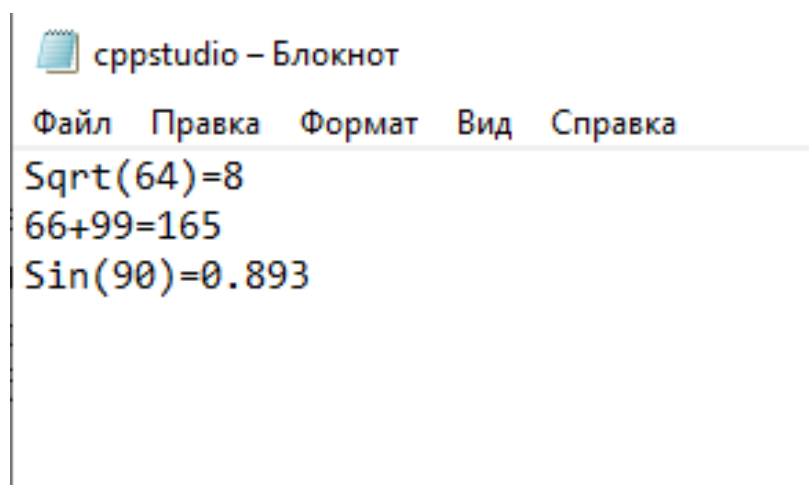
Для вычисления тригонометрических операций пользователю необходимо ввести число в текстовое поле (число вводится в радианах). После этого нажать на кнопку соответствующей операции. В текстовом поле появится результат вычисления. Если число не будет введено, то кнопка будет заблокирована.



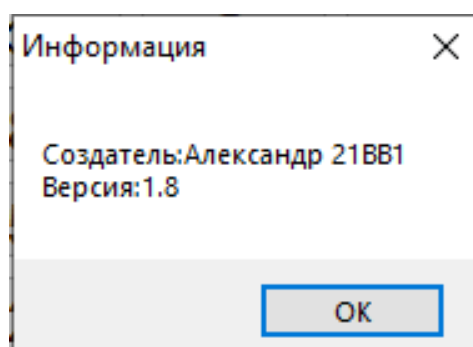
Для работы с дробными числами необходимо ввести целую часть числа, затем нажать на кнопку “,” и ввести дробную часть. Если не ввести число, или число в текстовом поле уже содержит знак дробной части, то действие кнопки будет заблокировано.



Для просмотра истории вычислений необходимо нажать на кнопку “History”. После нажатия откроется текстовый документ с историей операций.



Для получения дополнительной информации о приложении необходимо нажать на кнопку “Info”. После нажатия откроется окно с информацией о версии приложения и разработчике.



Заключение

При выполнении данной курсовой работы были получены навыки разработки многомодульных программ. Была освоена работа с графическим интерфейсом, изучены функции работы с файлами и математическим сопроцессором. Также были получены основные навыки отладки и тестирования программ и программирования в среде VisualStudio 2010 на языках C++ и Assembler.

В рамках курсовой работы была написана программа с графическим интерфейсом, выполняющая функции калькулятора.

В дальнейшем программу можно улучшить, путем добавления новых математических операций или вспомогательных функций, что даст пользователю ещё больше удобства при работе с приложением.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Осипов Н.А. Разработка Windows приложений на C#. Учебное пособие. 2012 г.
2. Герберт Шилдт. C++. Базовый курс. 2014 г.

ПРИЛОЖЕНИЕ А

ЛИСТИНГИ ПРОГРАММЫ

Файл Form1.h

```
#pragma once

#define _CRT_SECURE_NO_WARNINGS
#include <math.h>
#include "stdafx.h"
#include <stdio.h>
#include <string>
#include <locale.h>
#include <stdlib.h>
#include "plus.h"
#include "minus.h"
#include "multiplication.h"
#include "division.h"
#include "fileadd.h"
#include "Picture.h"
namespace Kursovaya {

    using namespace System::IO;
    using namespace System::Threading;
    using namespace System;
    using namespace System::ComponentModel;
    using namespace System::Collections;
    using namespace System::Windows::Forms;
    using namespace System::Data;
    using namespace System::Drawing;
    using namespace std;
    using namespace System::Diagnostics;

    /// <summary>
    /// Сводка для Form1
    /// </summary>
    public ref class Form1 : public System::Windows::Forms::Form
    {
    public:
        Form1(void)
        {
            InitializeComponent();
            //
            //TODO: добавьте код конструктора
            //
        }

    protected:
        /// <summary>
        /// Освободить все используемые ресурсы.
        /// </summary>
        ~Form1()
        {
            if (components)
            {
                delete components;
            }
        }
    }
```

```

private: System::Windows::Forms::TextBox^ textBox1;
private: System::Windows::Forms::Button^ button1;
private: System::Windows::Forms::Button^ button2;
private: System::Windows::Forms::Button^ button3;
private: System::Windows::Forms::Button^ button4;
private: System::Windows::Forms::Button^ button5;
private: System::Windows::Forms::Button^ button6;
private: System::Windows::Forms::Button^ button7;
private: System::Windows::Forms::Button^ button8;
private: System::Windows::Forms::Button^ button9;
private: System::Windows::Forms::Button^ button10;
private: System::Windows::Forms::Button^ button11;
private: System::Windows::Forms::Button^ button12;
private: System::Windows::Forms::Button^ button13;
private: System::Windows::Forms::Button^ button14;
private: System::Windows::Forms::Button^ button15;
private: System::Windows::Forms::Button^ button16;
private: System::Windows::Forms::Button^ button17;
private: System::Windows::Forms::Button^ button18;
private: System::Windows::Forms::Button^ button19;
private: System::Windows::Forms::Button^ button20;
private: System::Windows::Forms::Button^ button21;
private: System::Windows::Forms::TextBox^ textBox2;
private: System::Windows::Forms::Button^ button22;
private: System::Windows::Forms::Button^ button23;
private: System::Windows::Forms::Button^ button24;
private: System::Windows::Forms::ContextMenuStrip^ contextMenuStrip1;
private: System::Windows::Forms::ToolStripTextBox^ toolStripTextBox1;
private: System::Windows::Forms::ToolStripMenuItem^ infoToolStripMenuItem;
private: System::Windows::Forms::ContextMenuStrip^ contextMenuStrip2;
private: System::Windows::Forms::ContextMenuStrip^ contextMenuStrip3;
private: System::Windows::Forms::ToolStripTextBox^ toolStripTextBox2;
private: System::Windows::Forms::ContextMenuStrip^ contextMenuStrip4;
private: System::Windows::Forms::ContextMenuStrip^ contextMenuStrip5;
private: System::Windows::Forms::ContextMenuStrip^ contextMenuStrip6;
private: System::Windows::Forms::MenuStrip^ menuStrip1;
private: System::Windows::Forms::ToolStripMenuItem^ infoToolStripMenuItem1;

```

```

private: System::ComponentModel::IContainer^ components;

```

```

protected:

```

```

private:

```

```

    /// <summary>
    /// Требуется переменная конструктора.
    /// </summary>

```

```

#pragma region Windows Form Designer generated code

```

```

    /// <summary>
    /// Обязательный метод для поддержки конструктора - не изменяйте
    /// содержимое данного метода при помощи редактора кода.
    /// </summary>

```

```

void InitializeComponent(void)
{
    this->components = (gcnew System::ComponentModel::Container());
    System::ComponentModel::ComponentResourceManager^ resources = (gcnew
System::ComponentModel::ComponentResourceManager(Form1::typeid));
    this->textBox1 = (gcnew System::Windows::Forms::TextBox());
    this->button1 = (gcnew System::Windows::Forms::Button());
    this->button2 = (gcnew System::Windows::Forms::Button());
    this->button3 = (gcnew System::Windows::Forms::Button());
    this->button4 = (gcnew System::Windows::Forms::Button());
    this->button5 = (gcnew System::Windows::Forms::Button());
    this->button6 = (gcnew System::Windows::Forms::Button());
    this->button7 = (gcnew System::Windows::Forms::Button());
    this->button8 = (gcnew System::Windows::Forms::Button());
    this->button9 = (gcnew System::Windows::Forms::Button());
    this->button10 = (gcnew System::Windows::Forms::Button());
    this->button11 = (gcnew System::Windows::Forms::Button());
    this->button12 = (gcnew System::Windows::Forms::Button());
    this->button13 = (gcnew System::Windows::Forms::Button());
    this->button14 = (gcnew System::Windows::Forms::Button());
    this->button15 = (gcnew System::Windows::Forms::Button());
    this->button16 = (gcnew System::Windows::Forms::Button());
    this->button17 = (gcnew System::Windows::Forms::Button());
    this->button18 = (gcnew System::Windows::Forms::Button());
    this->button19 = (gcnew System::Windows::Forms::Button());
    this->button20 = (gcnew System::Windows::Forms::Button());
    this->button21 = (gcnew System::Windows::Forms::Button());
    this->textBox2 = (gcnew System::Windows::Forms::TextBox());
    this->button22 = (gcnew System::Windows::Forms::Button());
    this->button23 = (gcnew System::Windows::Forms::Button());
    this->button24 = (gcnew System::Windows::Forms::Button());
    this->contextMenuStrip1 = (gcnew System::Windows::Forms::ContextMenuStrip(this-
>components));
    this->toolStripTextBox1 = (gcnew System::Windows::Forms::ToolStripTextBox());
    this->infoToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
    this->contextMenuStrip2 = (gcnew System::Windows::Forms::ContextMenuStrip(this-
>components));
    this->contextMenuStrip3 = (gcnew System::Windows::Forms::ContextMenuStrip(this-
>components));
    this->toolStripTextBox2 = (gcnew System::Windows::Forms::ToolStripTextBox());
    this->contextMenuStrip4 = (gcnew System::Windows::Forms::ContextMenuStrip(this-
>components));
    this->contextMenuStrip5 = (gcnew System::Windows::Forms::ContextMenuStrip(this-
>components));
    this->contextMenuStrip6 = (gcnew System::Windows::Forms::ContextMenuStrip(this-
>components));
    this->menuStrip1 = (gcnew System::Windows::Forms::MenuStrip());
    this->infoToolStripMenuItem1 = (gcnew
System::Windows::Forms::ToolStripMenuItem());
    this->contextMenuStrip1->SuspendLayout();
    this->contextMenuStrip3->SuspendLayout();
    this->menuStrip1->SuspendLayout();
    this->SuspendLayout();
    //
    // textBox1
    //
    this->textBox1->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif",
20.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204)));
    this->textBox1->Location = System::Drawing::Point(11, 25);
    this->textBox1->Multiline = true;

```

```

        this->textBox1->Name = L"textBox1";
        this->textBox1->Size = System::Drawing::Size(408, 31);
        this->textBox1->TabIndex = 21;
        this->textBox1->TextAlign = System::Windows::Forms::HorizontalAlignment::Right;
        this->textBox1->TextChanged += gcnew System::EventHandler(this,
&Form1::textBox1_TextChanged);
        this->textBox1->KeyPress += gcnew
System::Windows::Forms::KeyPressEventHandler(this, &Form1::textBox1_KeyPress);
        //
        // button1
        //
        this->button1->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 18,
System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->button1->Location = System::Drawing::Point(11, 176);
        this->button1->Name = L"button1";
        this->button1->Size = System::Drawing::Size(76, 32);
        this->button1->TabIndex = 22;
        this->button1->Text = L"1";
        this->button1->UseVisualStyleBackColor = true;
        this->button1->Click += gcnew System::EventHandler(this, &Form1::button1_Click);
        //
        // button2
        //
        this->button2->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 18,
System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->button2->Location = System::Drawing::Point(94, 176);
        this->button2->Name = L"button2";
        this->button2->Size = System::Drawing::Size(76, 32);
        this->button2->TabIndex = 23;
        this->button2->Text = L"2";
        this->button2->UseVisualStyleBackColor = true;
        this->button2->Click += gcnew System::EventHandler(this,
&Form1::button2_Click_1);
        //
        // button3
        //
        this->button3->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 18,
System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->button3->Location = System::Drawing::Point(176, 176);
        this->button3->Name = L"button3";
        this->button3->Size = System::Drawing::Size(76, 32);
        this->button3->TabIndex = 24;
        this->button3->Text = L"3";
        this->button3->UseVisualStyleBackColor = true;
        this->button3->Click += gcnew System::EventHandler(this, &Form1::button3_Click);
        //
        // button4
        //
        this->button4->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 18,
System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->button4->Location = System::Drawing::Point(12, 138);
        this->button4->Name = L"button4";
        this->button4->Size = System::Drawing::Size(76, 32);
        this->button4->TabIndex = 25;
        this->button4->Text = L"4";
        this->button4->UseVisualStyleBackColor = true;
        this->button4->Click += gcnew System::EventHandler(this, &Form1::button4_Click);
        //

```

```

// button5
//
this->button5->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 18,
System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(204)));
this->button5->Location = System::Drawing::Point(94, 138);
this->button5->Name = L"button5";
this->button5->Size = System::Drawing::Size(76, 32);
this->button5->TabIndex = 26;
this->button5->Text = L"5";
this->button5->UseVisualStyleBackColor = true;
this->button5->Click += gcnew System::EventHandler(this, &Form1::button5_Click);
//
// button6
//
this->button6->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 18,
System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(204)));
this->button6->Location = System::Drawing::Point(176, 138);
this->button6->Name = L"button6";
this->button6->Size = System::Drawing::Size(76, 32);
this->button6->TabIndex = 27;
this->button6->Text = L"6";
this->button6->UseVisualStyleBackColor = true;
this->button6->Click += gcnew System::EventHandler(this, &Form1::button6_Click);
//
// button7
//
this->button7->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 18,
System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(204)));
this->button7->Location = System::Drawing::Point(12, 100);
this->button7->Name = L"button7";
this->button7->Size = System::Drawing::Size(76, 32);
this->button7->TabIndex = 28;
this->button7->Text = L"7";
this->button7->UseVisualStyleBackColor = true;
this->button7->Click += gcnew System::EventHandler(this, &Form1::button7_Click);
//
// button8
//
this->button8->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 18,
System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(204)));
this->button8->Location = System::Drawing::Point(94, 100);
this->button8->Name = L"button8";
this->button8->Size = System::Drawing::Size(76, 32);
this->button8->TabIndex = 29;
this->button8->Text = L"8";
this->button8->UseVisualStyleBackColor = true;
this->button8->Click += gcnew System::EventHandler(this, &Form1::button8_Click);
//
// button9
//
this->button9->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 18,
System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(204)));
this->button9->Location = System::Drawing::Point(176, 100);
this->button9->Name = L"button9";
this->button9->Size = System::Drawing::Size(76, 32);
this->button9->TabIndex = 30;
this->button9->Text = L"9";

```

```

        this->button9->UseVisualStyleBackColor = true;
        this->button9->Click += gcnew System::EventHandler(this, &Form1::button9_Click);
        //
        // button10
        //
        this->button10->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 18,
System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->button10->Location = System::Drawing::Point(12, 62);
        this->button10->Name = L"button10";
        this->button10->Size = System::Drawing::Size(76, 32);
        this->button10->TabIndex = 31;
        this->button10->Text = L"√";
        this->button10->UseVisualStyleBackColor = true;
        this->button10->Click += gcnew System::EventHandler(this,
&Form1::button10_Click);
        //
        // button11
        //
        this->button11->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 18,
System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->button11->Location = System::Drawing::Point(94, 62);
        this->button11->Name = L"button11";
        this->button11->Size = System::Drawing::Size(76, 32);
        this->button11->TabIndex = 32;
        this->button11->Text = L"□";
        this->button11->UseVisualStyleBackColor = true;
        this->button11->Click += gcnew System::EventHandler(this,
&Form1::button11_Click);
        //
        // button12
        //
        this->button12->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 18,
System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->button12->Location = System::Drawing::Point(176, 62);
        this->button12->Name = L"button12";
        this->button12->Size = System::Drawing::Size(76, 32);
        this->button12->TabIndex = 33;
        this->button12->Text = L"C";
        this->button12->UseVisualStyleBackColor = true;
        this->button12->Click += gcnew System::EventHandler(this,
&Form1::button12_Click);
        //
        // button13
        //
        this->button13->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 18,
System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->button13->Location = System::Drawing::Point(258, 62);
        this->button13->Name = L"button13";
        this->button13->Size = System::Drawing::Size(76, 32);
        this->button13->TabIndex = 34;
        this->button13->Text = L"+";
        this->button13->UseVisualStyleBackColor = true;
        this->button13->Click += gcnew System::EventHandler(this,
&Form1::button13_Click);
        //
        // button14
        //

```

```

        this->button14->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 18,
System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->button14->Location = System::Drawing::Point(258, 100);
        this->button14->Name = L"button14";
        this->button14->Size = System::Drawing::Size(76, 32);
        this->button14->TabIndex = 35;
        this->button14->Text = L"-";
        this->button14->UseVisualStyleBackColor = true;
        this->button14->Click += gcnew System::EventHandler(this,
&Form1::button14_Click);
        //
        // button15
        //
        this->button15->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 18,
System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->button15->Location = System::Drawing::Point(258, 138);
        this->button15->Name = L"button15";
        this->button15->Size = System::Drawing::Size(76, 32);
        this->button15->TabIndex = 36;
        this->button15->Text = L"x";
        this->button15->UseVisualStyleBackColor = true;
        this->button15->Click += gcnew System::EventHandler(this,
&Form1::button15_Click);
        //
        // button16
        //
        this->button16->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 18,
System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->button16->Location = System::Drawing::Point(258, 176);
        this->button16->Name = L"button16";
        this->button16->Size = System::Drawing::Size(76, 32);
        this->button16->TabIndex = 37;
        this->button16->Text = L"/";
        this->button16->UseVisualStyleBackColor = true;
        this->button16->Click += gcnew System::EventHandler(this,
&Form1::button16_Click);
        //
        // button17
        //
        this->button17->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 18,
System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->button17->Location = System::Drawing::Point(340, 62);
        this->button17->Name = L"button17";
        this->button17->Size = System::Drawing::Size(76, 32);
        this->button17->TabIndex = 38;
        this->button17->Text = L"Sin";
        this->button17->UseVisualStyleBackColor = true;
        this->button17->Click += gcnew System::EventHandler(this,
&Form1::button17_Click);
        //
        // button18
        //
        this->button18->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 18,
System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->button18->Location = System::Drawing::Point(340, 100);
        this->button18->Name = L"button18";
        this->button18->Size = System::Drawing::Size(76, 32);

```

```

        this->button18->TabIndex = 39;
        this->button18->Text = L"Cos";
        this->button18->UseVisualStyleBackColor = true;
        this->button18->Click += gcnew System::EventHandler(this,
&Form1::button18_Click);
        //
        // button19
        //
        this->button19->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 18,
System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->button19->Location = System::Drawing::Point(340, 138);
        this->button19->Name = L"button19";
        this->button19->Size = System::Drawing::Size(76, 32);
        this->button19->TabIndex = 40;
        this->button19->Text = L"Tan";
        this->button19->UseVisualStyleBackColor = true;
        this->button19->Click += gcnew System::EventHandler(this,
&Form1::button19_Click);
        //
        // button20
        //
        this->button20->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 18,
System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->button20->Location = System::Drawing::Point(340, 176);
        this->button20->Name = L"button20";
        this->button20->Size = System::Drawing::Size(76, 32);
        this->button20->TabIndex = 41;
        this->button20->Text = L"Cot";
        this->button20->UseVisualStyleBackColor = true;
        this->button20->Click += gcnew System::EventHandler(this,
&Form1::button20_Click);
        //
        // button21
        //
        this->button21->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 18,
System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->button21->Location = System::Drawing::Point(258, 214);
        this->button21->Name = L"button21";
        this->button21->Size = System::Drawing::Size(158, 32);
        this->button21->TabIndex = 42;
        this->button21->Text = L"=";
        this->button21->UseVisualStyleBackColor = true;
        this->button21->Click += gcnew System::EventHandler(this,
&Form1::button21_Click);
        //
        // textBox2
        //
        this->textBox2->BorderStyle = System::Windows::Forms::BorderStyle::None;
        this->textBox2->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif",
20.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->textBox2->Location = System::Drawing::Point(12, 26);
        this->textBox2->Multiline = true;
        this->textBox2->Name = L"textBox2";
        this->textBox2->Size = System::Drawing::Size(189, 29);
        this->textBox2->TabIndex = 43;
        this->textBox2->TextChanged += gcnew System::EventHandler(this,
&Form1::textBox2_TextChanged);

```

```

        this->textBox2->KeyPress += gcnnew
System::Windows::Forms::KeyPressEventHandler(this, &Form1::textBox2_KeyPress);
//
// button22
//
        this->button22->Font = (gcnnew System::Drawing::Font(L"Microsoft Sans Serif", 18,
System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->button22->Location = System::Drawing::Point(94, 214);
        this->button22->Name = L"button22";
        this->button22->Size = System::Drawing::Size(76, 32);
        this->button22->TabIndex = 44;
        this->button22->Text = L"0";
        this->button22->UseVisualStyleBackColor = true;
        this->button22->Click += gcnnew System::EventHandler(this,
&Form1::button22_Click);
//
// button23
//
        this->button23->Font = (gcnnew System::Drawing::Font(L"Microsoft Sans Serif", 18,
System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->button23->ImageAlign = System::Drawing::ContentAlignment::TopCenter;
        this->button23->Location = System::Drawing::Point(176, 214);
        this->button23->Name = L"button23";
        this->button23->Size = System::Drawing::Size(76, 32);
        this->button23->TabIndex = 45;
        this->button23->Text = L"";
        this->button23->UseVisualStyleBackColor = true;
        this->button23->Click += gcnnew System::EventHandler(this,
&Form1::button23_Click);
//
// button24
//
        this->button24->Font = (gcnnew System::Drawing::Font(L"Microsoft Sans Serif", 12,
System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->button24->Location = System::Drawing::Point(11, 214);
        this->button24->Name = L"button24";
        this->button24->Size = System::Drawing::Size(76, 32);
        this->button24->TabIndex = 46;
        this->button24->Text = L"History";
        this->button24->UseVisualStyleBackColor = true;
        this->button24->Click += gcnnew System::EventHandler(this,
&Form1::button24_Click);
//
// contextMenuStrip1
//
        this->contextMenuStrip1->Items->AddRange(gcnnew cli::array<
System::Windows::Forms::ToolStripItem^ >(2) {this->toolStripTextBox1,
        this->infoToolStripMenuItem});
        this->contextMenuStrip1->Name = L"contextMenuStrip1";
        this->contextMenuStrip1->Size = System::Drawing::Size(161, 51);
//
// toolStripTextBox1
//
        this->toolStripTextBox1->Name = L"toolStripTextBox1";
        this->toolStripTextBox1->Size = System::Drawing::Size(100, 23);
//
// infoToolStripMenuItem
//
        this->infoToolStripMenuItem->Name = L"infoToolStripMenuItem";

```

```

this->infoToolStripMenuItem->Size = System::Drawing::Size(160, 22);
this->infoToolStripMenuItem->Text = L"Info";
//
// contextMenuStrip2
//
this->contextMenuStrip2->Name = L"contextMenuStrip2";
this->contextMenuStrip2->Size = System::Drawing::Size(61, 4);
//
// contextMenuStrip3
//
this->contextMenuStrip3->Items->AddRange(gcnew cli::array<
System::Windows::Forms::ToolStripItem^ >(1) {this->toolStripTextBox2});
this->contextMenuStrip3->Name = L"contextMenuStrip3";
this->contextMenuStrip3->Size = System::Drawing::Size(161, 29);
//
// toolStripTextBox2
//
this->toolStripTextBox2->Name = L"toolStripTextBox2";
this->toolStripTextBox2->Size = System::Drawing::Size(100, 23);
//
// contextMenuStrip4
//
this->contextMenuStrip4->Name = L"contextMenuStrip4";
this->contextMenuStrip4->Size = System::Drawing::Size(61, 4);
//
// contextMenuStrip5
//
this->contextMenuStrip5->Name = L"contextMenuStrip5";
this->contextMenuStrip5->Size = System::Drawing::Size(61, 4);
//
// contextMenuStrip6
//
this->contextMenuStrip6->Name = L"contextMenuStrip6";
this->contextMenuStrip6->Size = System::Drawing::Size(61, 4);
//
// menuStrip1
//
this->menuStrip1->Items->AddRange(gcnew cli::array<
System::Windows::Forms::ToolStripItem^ >(1) {this->infoToolStripMenuItem1});
this->menuStrip1->Location = System::Drawing::Point(0, 0);
this->menuStrip1->Name = L"menuStrip1";
this->menuStrip1->Size = System::Drawing::Size(433, 24);
this->menuStrip1->TabIndex = 49;
this->menuStrip1->Text = L"menuStrip1";
this->menuStrip1->ItemClicked += gcnew
System::Windows::Forms::ToolStripItemClickedEventHandler(this, &Form1::menuStrip1_ItemClicked);
//
// infoToolStripMenuItem1
//
this->infoToolStripMenuItem1->Name = L"infoToolStripMenuItem1";
this->infoToolStripMenuItem1->Size = System::Drawing::Size(40, 20);
this->infoToolStripMenuItem1->Text = L"Info";
//
// Form1
//
this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);
this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;
this->ClientSize = System::Drawing::Size(433, 257);
this->Controls->Add(this->menuStrip1);
this->Controls->Add(this->button24);
this->Controls->Add(this->button23);
this->Controls->Add(this->button22);

```

```

        this->Controls->Add(this->textBox2);
        this->Controls->Add(this->button21);
        this->Controls->Add(this->button20);
        this->Controls->Add(this->button19);
        this->Controls->Add(this->button18);
        this->Controls->Add(this->button17);
        this->Controls->Add(this->button16);
        this->Controls->Add(this->button15);
        this->Controls->Add(this->button14);
        this->Controls->Add(this->button13);
        this->Controls->Add(this->button12);
        this->Controls->Add(this->button11);
        this->Controls->Add(this->button10);
        this->Controls->Add(this->button9);
        this->Controls->Add(this->button8);
        this->Controls->Add(this->button7);
        this->Controls->Add(this->button6);
        this->Controls->Add(this->button5);
        this->Controls->Add(this->button4);
        this->Controls->Add(this->button3);
        this->Controls->Add(this->button2);
        this->Controls->Add(this->button1);
        this->Controls->Add(this->textBox1);
        this->Icon = (cli::safe_cast<System::Drawing::Icon^ >(resources-
>GetObject(L"$this.Icon")));
        this->Name = L"Form1";
        this->StartPosition = System::Windows::Forms::FormStartPosition::CenterScreen;
        this->Text = L"Калькулятор";
        this->Load += gcnew System::EventHandler(this, &Form1::Form1_Load);
        this->contextMenuStrip1->ResumeLayout(false);
        this->contextMenuStrip1->PerformLayout();
        this->contextMenuStrip3->ResumeLayout(false);
        this->contextMenuStrip3->PerformLayout();
        this->menuStrip1->ResumeLayout(false);
        this->menuStrip1->PerformLayout();
        this->ResumeLayout(false);
        this->PerformLayout();

    }
#pragma endregion

```

```

    int actions;
    double a,b,result;
    String^ s;

```

```

private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e) {
    if (textBox1->Text->Length < 7){
        String^ s = textBox1->Text;
        textBox1->Text=s+"1";
    }
}

private: System::Void button2_Click_1(System::Object^ sender, System::EventArgs^ e) {
    if (textBox1->Text->Length < 7){
        String^ s = textBox1->Text;
        textBox1->Text=s+"2";
    }
}

private: System::Void button3_Click(System::Object^ sender, System::EventArgs^ e) {
    if (textBox1->Text->Length < 7){

```

```

        String^ s = textBox1->Text;
        textBox1->Text=s+"3";
    }
}
private: System::Void button4_Click(System::Object^ sender, System::EventArgs^ e) {
    if (textBox1->Text->Length < 7){
        String^ s = textBox1->Text;
        textBox1->Text=s+"4";
    }
}
private: System::Void button5_Click(System::Object^ sender, System::EventArgs^ e) {
    if (textBox1->Text->Length < 7){
        String^ s = textBox1->Text;
        textBox1->Text=s+"5";
    }
}
private: System::Void button6_Click(System::Object^ sender, System::EventArgs^ e) {
    if (textBox1->Text->Length < 7){
        String^ s = textBox1->Text;
        textBox1->Text=s+"6";
    }
}
private: System::Void button7_Click(System::Object^ sender, System::EventArgs^ e) {
    if (textBox1->Text->Length < 7){
        String^ s = textBox1->Text;
        textBox1->Text=s+"7";
    }
}
private: System::Void button8_Click(System::Object^ sender, System::EventArgs^ e) {
    if (textBox1->Text->Length < 7){
        String^ s = textBox1->Text;
        textBox1->Text=s+"8";
    }
}
private: System::Void button9_Click(System::Object^ sender, System::EventArgs^ e) {
    if (textBox1->Text->Length < 7){
        String^ s = textBox1->Text;
        textBox1->Text=s+"9";
    }
}
private: System::Void button22_Click(System::Object^ sender, System::EventArgs^ e) {
    if (textBox1->Text->Length < 7){
        String^ s = textBox1->Text;
        textBox1->Text=s+"0";
    }
}
private: System::Void button10_Click(System::Object^ sender, System::EventArgs^ e) {
    if (textBox1->Text->Length != 0){
        String^ s = textBox1->Text;
        a=System::Convert::ToSingle(textBox1->Text);
        actions = 5;
        b=sqrt(a);
        b= floorf(b * 1000) / 1000;
        fileadd(a,b,result,actions);
        textBox1->Text = System::Convert::ToString(b);
    }
}
private: System::Void button11_Click(System::Object^ sender, System::EventArgs^ e) {
    int lenght = textBox1->Text->Length - 1;
    String^ text = textBox1->Text;
    textBox1->Clear();
}

```

```

        for (int i = 0; i < lenght; i++)
        {
            textBox1->Text = textBox1->Text + text[i];
        }
    }
private: System::Void button12_Click(System::Object^ sender, System::EventArgs^ e) {
    textBox1->Text = "0";
    textBox2->Text = "0";
}

private: System::Void button13_Click(System::Object^ sender, System::EventArgs^ e) {
    while (textBox1->Text->Length != 0){
        a=System::Convert::ToSingle(textBox1->Text);
        textBox1->Clear();
        actions = 1;
        a= floorf(a * 1000) / 1000;
        textBox2->Text = System::Convert::ToString(a) + "+";
    }
}

private: System::Void button14_Click(System::Object^ sender, System::EventArgs^ e) {
    while (textBox1->Text->Length != 0){
        a=System::Convert::ToSingle(textBox1->Text);
        textBox1->Clear();
        actions = 2;
        a= floorf(a * 1000) / 1000;
        textBox2->Text = System::Convert::ToString(a) + "-";
    }
}

private: System::Void button15_Click(System::Object^ sender, System::EventArgs^ e) {
    while (textBox1->Text->Length != 0){
        a=System::Convert::ToSingle(textBox1->Text);
        textBox1->Clear();
        actions = 3;
        a= floor(a * 1000) / 1000;
        textBox2->Text = System::Convert::ToString(a) + ".";
    }
}

private: System::Void button16_Click(System::Object^ sender, System::EventArgs^ e) {
    while (textBox1->Text->Length != 0){
        a=System::Convert::ToSingle(textBox1->Text);
        textBox1->Clear();
        actions = 4;

        textBox2->Text = System::Convert::ToString(a) + "/";
    }
}

private: System::Void button17_Click(System::Object^ sender, System::EventArgs^ e) {
    if (textBox1->Text->Length != 0){
        String^ s = textBox1->Text;
        a=System::Convert::ToSingle(textBox1->Text);
        actions = 6;
        b=sin(a);
        b= floorf(b * 1000) / 1000;
        fileadd(a,b,result,actions);
        textBox1->Text = System::Convert::ToString(b);
    }
}

private: System::Void button18_Click(System::Object^ sender, System::EventArgs^ e) {
    if (textBox1->Text->Length != 0){
        String^ s = textBox1->Text;
        a=System::Convert::ToSingle(textBox1->Text);
    }
}

```

```

        actions = 7;
        b=cos(a);
        b= floorf(b * 1000) / 1000;
        fileadd(a,b,result,actions);
        textBox1->Text = System::Convert::ToString(b);
    }
}
private: System::Void button19_Click(System::Object^ sender, System::EventArgs^ e) {
    if (textBox1->Text->Length != 0){
        String^ s = textBox1->Text;
        a=System::Convert::ToSingle(textBox1->Text);
        actions = 8;
        b=tan(a);
        b= floorf(b * 1000) / 1000;
        fileadd(a,b,result,actions);
        textBox1->Text = System::Convert::ToString(b);
    }
}
private: System::Void button20_Click(System::Object^ sender, System::EventArgs^ e) {
    if (textBox1->Text->Length != 0){
        String^ s = textBox1->Text;
        a=System::Convert::ToSingle(textBox1->Text);
        actions = 9;
        b=1/tan(a);
        b= floorf(b * 1000) / 1000;
        fileadd(a,b,result,actions);
        textBox1->Text = System::Convert::ToString(b);
    }
}
private: System::Void button21_Click(System::Object^ sender, System::EventArgs^ e) {
    if ((textBox1->Text->Length != 0)&&(textBox2->Text->Length != 0)){
        b=System::Convert::ToSingle(textBox1->Text);

        switch(actions){
            case 1:
                result = plus(a,b);
                fileadd(a,b,result,actions);

                textBox1->Text = System::Convert::ToString(plus(a,b));
                textBox2->Clear();

                break;
            case 2:
                result = minus(a,b);
                fileadd(a,b,result,actions);

                textBox1->Text = System::Convert::ToString(minus(a,b));
                textBox2->Clear();

                break;
            case 3:
                result = multiplication(a,b);
                fileadd(a,b,result,actions);

                // result = a * b;
                textBox1->Text = System::Convert::ToString(multiplication(a,b));
                textBox2->Clear();

                break;
            case 4:
                result = division(a,b);
                fileadd(a,b,result,actions);

                // result = a / b;
                textBox1->Text = System::Convert::ToString(division(a,b));
                textBox2->Clear();

                break;
        }
    }
}

```

```

        default:
            break;
        }
    }
}

private: System::Void textBox1_TextChanged(System::Object^ sender, System::EventArgs^ e) {
}

private: System::Void textBox2_TextChanged(System::Object^ sender, System::EventArgs^ e) {
}

private: System::Void button23_Click(System::Object^ sender, System::EventArgs^ e) {
    if (textBox1->Text->IndexOf(',') == -1){
        if (textBox1->Text->Length != 0){

            b=System::Convert::ToSingle(textBox1->Text);

            String^ s = textBox1->Text;
            textBox1->Text=s+",";
        }
        else{
        }
    }
}

private: System::Void button24_Click(System::Object^ sender, System::EventArgs^ e) {
    Process::Start("notepad.exe", "cppstudio.txt");
}

private: System::Void Form1_Load(System::Object^ sender, System::EventArgs^ e) {
}

private: System::Void menuStrip1_ItemClicked(System::Object^ sender,
System::Windows::Forms::ToolStripItemClickedEventArgs^ e) {
    MessageBox::Show("Создатель:Александр 21ВВ1\nВерсия:1.8","Информация");
}

private: System::Void textBox1_KeyPress(System::Object^ sender,
System::Windows::Forms::KeyPressEventArgs^ e) {
    if ((e->KeyChar <= 47 || e->KeyChar >= 58) && e->KeyChar != 8)
        e->Handled = true;
}

private: System::Void textBox2_KeyPress(System::Object^ sender,
System::Windows::Forms::KeyPressEventArgs^ e) {
    e->Handled = true;
}

};
}

```

Файл Kursovaya.cpp

```
#include "stdafx.h"
#include "Form1.h"
#include "Picture.h"

#include <stdio.h>
#include "locale.h"

using namespace Kursovaya;

void main(array<System::String ^> ^argv)
{
    Application::SetCompatibleTextRenderingDefault(false);
    Picture^ f = gcnew Picture();

    // Включение визуальных эффектов Windows XP до создания каких-либо элементов
    // управления
    Application::EnableVisualStyles();

    // Создание главного окна и его запуск

    DateTime end = DateTime::Now + TimeSpan::FromSeconds(3);
    f->Show();
    while(end > DateTime::Now){
        Application::DoEvents();
    }
    f->Close();

    Application::Run(gcnew Form1());
}
```

Файл Picture.h

```
#pragma once
#include "windows.h"
namespace Kursovaya {

    using namespace System;
    using namespace System::Threading;
    using namespace System::ComponentModel;
    using namespace System::Collections;
    using namespace System::Windows::Forms;
    using namespace System::Data;
    using namespace System::Drawing;

    /// <summary>
    /// Сводка для Picture
    /// </summary>
    public ref class Picture : public System::Windows::Forms::Form
    {
    }
```

```

public:
    Picture(void)
    {
        InitializeComponent();
        timer1->Start();

        //
        //TODO: добавьте код конструктора
        //
    }

protected:
    /// <summary>
    /// Освободить все используемые ресурсы.
    /// </summary>
    ~Picture()
    {
        if (components)
        {
            delete components;
        }
    }

private: System::Windows::Forms::Timer^ timer1;
private: System::Windows::Forms::PictureBox^ pictureBox1;

private: System::ComponentModel::IContainer^ components;
protected:

protected:

private:
    /// <summary>
    /// Требуется переменная конструктора.
    /// </summary>

#pragma region Windows Form Designer generated code
    /// <summary>
    /// Обязательный метод для поддержки конструктора - не изменяйте
    /// содержимое данного метода при помощи редактора кода.
    /// </summary>
    void InitializeComponent(void)
    {
        this->components = (gcnew System::ComponentModel::Container());
        System::ComponentModel::ComponentResourceManager^ resources = (gcnew
System::ComponentModel::ComponentResourceManager(Picture::typeid));
        this->timer1 = (gcnew System::Windows::Forms::Timer(this->components));
        this->pictureBox1 = (gcnew System::Windows::Forms::PictureBox());
        (cli::safe_cast<System::ComponentModel::ISupportInitialize^ >(this->pictureBox1))-
>BeginInit();

        this->SuspendLayout();
        //
        // timer1
        //
        this->timer1->Interval = 10;
        this->timer1->Tick += gcnew System::EventHandler(this, &Picture::timer1_Tick);
        //
    }

```

```

        // pictureBox1
        //
        this->pictureBox1->BackColor = System::Drawing::Color::Transparent;
        this->pictureBox1->BackgroundImageLayout =
System::Windows::Forms::ImageLayout::None;
        this->pictureBox1->Image = (cli::safe_cast<System::Drawing::Image^ >(resources-
>GetObject(L"pictureBox1.Image")));
        this->pictureBox1->Location = System::Drawing::Point(35, 12);
        this->pictureBox1->Name = L"pictureBox1";
        this->pictureBox1->Size = System::Drawing::Size(529, 524);
        this->pictureBox1->SizeMode =
System::Windows::Forms::PictureBoxSizeMode::Zoom;
        this->pictureBox1->TabIndex = 0;
        this->pictureBox1->TabStop = false;
        this->pictureBox1->UseWaitCursor = true;
        //
        // Picture
        //
        this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);
        this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;
        this->BackColor = System::Drawing::Color::White;
        this->BackgroundImageLayout = System::Windows::Forms::ImageLayout::Center;
        this->ClientSize = System::Drawing::Size(584, 584);
        this->ControlBox = false;
        this->Controls->Add(this->pictureBox1);
        this->Cursor = System::Windows::Forms::Cursors::WaitCursor;
        this->ForeColor = System::Drawing::SystemColors::Control;
        this->FormBorderStyle = System::Windows::Forms::FormBorderStyle::FixedDialog;
        this->Name = L"Picture";
        this->RightToLeft = System::Windows::Forms::RightToLeft::Yes;
        this->RightToLeftLayout = true;
        this->StartPosition = System::Windows::Forms::FormStartPosition::CenterScreen;
        this->UseWaitCursor = true;
        this->Load += gcnew System::EventHandler(this, &Picture::Picture_Load);
        (cli::safe_cast<System::ComponentModel::ISupportInitialize^ >(this->pictureBox1))-
>EndInit();
        this->ResumeLayout(false);

    }
#pragma endregion
    private: System::Void Picture_Load(System::Object^ sender, System::EventArgs^ e) {
    }
    private: System::Void timer1_Tick(System::Object^ sender, System::EventArgs^ e) {

    }

};
}

```

Файл Fileadd.cpp

```
#include "stdafx.h"
#include <fstream>
#include <math.h>

using namespace std;

double fileadd(double a, double b, double result, int actions)
{
    ofstream fout("cppstudio.txt", ios_base::app);
    switch(actions){
        case 1:
            fout << a << "+" << b << "=" << result << "\n"; // запись строки в файл
            fout.close(); // закрываем файл
            break;
        case 2:
            fout << a << "-" << b << "=" << result << "\n"; // запись строки в файл
            fout.close(); // закрываем файл
            break;
        case 3:
            fout << a << "*" << b << "=" << result << "\n"; // запись строки в файл
            fout.close(); // закрываем файл
            break;
        case 4:
            fout << a << "/" << b << "=" << result << "\n"; // запись строки в файл
            fout.close(); // закрываем файл
            break;
        case 5:
            fout << "Sqrt(" << a << ")" << "=" << b << "\n"; // запись строки в файл
            fout.close(); // закрываем файл
            break;
        case 6:
            fout << "Sin(" << a << ")" << "=" << b << "\n"; // запись строки в файл
            fout.close(); // закрываем файл
            break;
        case 7:
            fout << "Cos(" << a << ")" << "=" << b << "\n"; // запись строки в файл
            fout.close(); // закрываем файл
            break;
        case 8:
            fout << "Tan(" << a << ")" << "=" << b << "\n"; // запись строки в файл
            fout.close(); // закрываем файл
            break;
        case 9:
            fout << "Cot(" << a << ")" << "=" << b << "\n"; // запись строки в файл
            fout.close(); // закрываем файл
            break;

        default:
            break;
    }

    return 0;
}
```

Файл Fileadd.h

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>
#ifndef FILEADD_H
#define FILEADD_H
double fileadd(double a, double b, double result, int actions);
#endif FILEADD_H
```

Файл plus.cpp

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>
#include "StdAfx.h"
#include <cmath>

double plus(double a, double b) {
    double result, res;

    __asm{
        fld a
        fld b
        fadd st(0),st(1)
        fst result
        FFREE ST(0)
        FFREE ST(1)
    }
    result= floorf(result * 10000) / 10000;
    return result;
}
```

Файл plus.h

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>
#ifndef PLUS_H
#define PLUS_H
double plus(double a, double b);
#endif PLUS_H
```

Файл minus.cpp

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>
#include "StdAfx.h"
#include <cmath>

double minus(double a, double b) {
    double result;
    __asm{
```

```

        fld b
    fld a
        fsub st(0),st(1)
        fst result
        FFREE ST(0)
        FFREE ST(1)
    }
    result= floorf(result * 10000) / 10000;
    return result;
}

```

Файл minus.h

```

#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>
#ifndef MINUS_H
#define MINUS_H
double minus(double a, double b);
#endif MINUS_H

```

Файл division.cpp

```

#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>
#include "StdAfx.h"
#include <cmath>

double division(double a, double b) {
    double result;
    __asm{
        fld b
    fld a
        fdiv st(0),st(1)
        fst result
        FFREE ST(1)
        FFREE ST(0)
    }
    result= floorf(result * 10000) / 10000;
    return result;
}

```

Файл division.h

```

#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>
#ifndef DIVISION_H
#define DIVISION_H
double division(double a, double b);
#endif DIVISION_H

```

Файл multiplication.cpp

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>
#include "StdAfx.h"
#include <cmath>

double multiplication(double a, double b) {
    double result;
    __asm{

        fld a
        fld b
        fmul st(0),st(1)
        fst result
        FFREE ST(0)
        FFREE ST(1)
    }
    result= floorf(result * 10000) / 10000;
    return result;
}
```

Файл multiplication.h

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>
#ifndef MULTIPLICATION_H
#define MULTIPLICATION_H
double multiplication(double a, double b);
#endif MULTIPLICATION_H
```

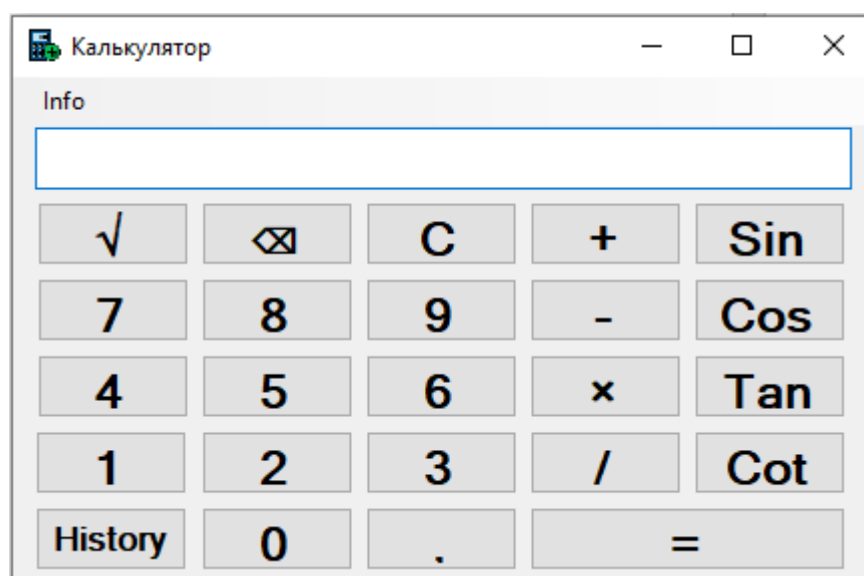
ПРИЛОЖЕНИЕ В

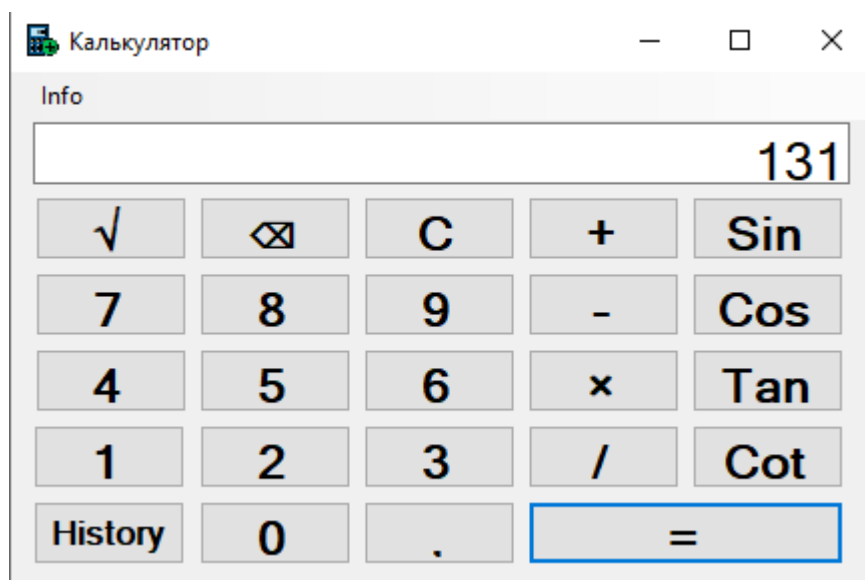
РЕЗУЛЬТАТЫ РАБОТЫ ПРОГРАММЫ

Приложение В.1 – Информационная заставка



Приложение В.2 – Интерфейс программы





ПРИЛОЖЕНИЕ С

ФАЙЛ С РЕЗУЛЬТАТАМИ ВЫЧИСЛЕНИЙ

