|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  Калужский филиал  федерального государственного бюджетного  образовательного учреждения высшего образования  ***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»***  ***(КФ МГТУ им. Н.Э. Баумана)*** |

**ФАКУЛЬТЕТ** \_ ***ИУК «Информатика и управление»\_\_\_\_\_\_\_\_\_***

**КАФЕДРА** \_\_ ***ИУК5 «Системы обработки информации»***

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**к курсовой работе на тему:**

***Калькулятор с расширенными возможностями***

по дисциплине ***Системное программирование***

Студент гр. ИУК5-42Б \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Кузнецов Р.С. )

(подпись) (Ф.И.О.)

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

(подпись) (Ф.И.О.)

Оценка руководителя \_\_\_\_\_ баллов \_\_\_\_\_\_\_\_\_\_\_

30-50 (дата)

Оценка защиты \_\_\_\_\_ баллов \_\_\_\_\_\_\_\_\_\_\_

30-50 (дата)

Оценка проекта \_\_\_\_\_ баллов \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(оценка по пятибалльной шкале)

Комиссия: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

(подпись) (Ф.И.О.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

(подпись) (Ф.И.О.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

(подпись) (Ф.И.О.)

Калуга, 2021

Калужский филиал   
федерального государственного бюджетного образовательного учреждения высшего образования

***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»   
(КФ МГТУ им. Н.Э. Баумана)***

УТВЕРЖДАЮ

Заведующий кафедрой **\_\_ИУК5\_\_\_**

\_\_\_\_\_\_\_\_\_\_\_(Е.В. Вершинин)

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_г.

**ЗАДАНИЕ**

**на выполнение курсовой работы**

по дисциплине ***Системное программирование***

Студент\_\_\_\_\_Кузнецов Р.С. ИУК5-42Б\_\_\_\_\_\_\_\_\_\_\_ \_

(фамилия, инициалы, индекс группы)

Руководитель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, инициалы)

График выполнения проекта: 25% к\_4\_нед., 50% к\_7\_нед., 75% к\_10\_нед., 100% к\_14\_нед.

***1. Тема курсового проекта***

***Калькулятор с расширенными возможностями***

***2. Техническое задание***

*Разработать калькулятор с использованием функций Windows API для вычисления арифметических и тригонометрических выражений*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***3. Оформление курсового проекта***

3.1. Расчетно-пояснительная записка на\_\_\_\_\_\_\_\_ листах формата А4.

3.2. Перечень графического материала КП (плакаты, схемы, чертежи и т.п.)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата выдачи задания «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_г.

Руководитель курсового проекта \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

(подпись) (Ф.И.О.)

Задание получил\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_20\_\_г.

(подпись) (Ф.И.О.)

Примечание:

Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

**СОДЕРЖАНИЕ**

1. Техническое задание…………………………………………………………5
   1. Наименование………………………………………………………….5
   2. Основание для разработки……………………………………………5
   3. Исполнитель…………………………………………………………...5
   4. Цель разработки………………………………………………………5
   5. Содержание работы…………………………………………………...6
      1. Задачи, подлежащие решению……………………………………6
      2. Требования к архитектуре АСОИ………………………………...7
      3. Требования к составу программных компонентов……………...7
      4. Требования к прикладным программам………………………….7
      5. Требования к входным/выходным данным……………………...7
      6. Требования к временных характеристикам……………………..8
      7. Требования к составу технических средств……………………..8
   6. Этапы разработки……………………..……………………………...8
   7. Техническая документация, предъявляемая по окончании работы…………………………………………………………………7
   8. Дополнительные условия……………………..……………………...8
2. Научно-исследовательская часть…………………………………………..10
   1. Постановка задачи проектирования………………………………...10
   2. Описание предметной области……………………………………...10
   3. Анализ аналогов и прототипов……………………..……………….13
   4. Перечень задач, подлежащих решению в процессе разработки….17
   5. Обоснование выбора инструментов и платформы для разработки………………………………………………………….....17
3. Проектно-конструкторская часть………………………………………….24
   1. Разработка структуры приложения…………………………………24
   2. Используемые функции Win32API…………………………………25
   3. Разработка архитектуры приложения………………………………30
   4. Разработка алгоритмов приложения………………………………..31
   5. Разработка интерфейса взаимодействия пользователя с системой……………………………………………………………...32
4. Проектно-технологическая часть………………………………………....34
   1. Тестирование и отладка макета рабочей программы………….….34
   2. Разработка руководства администратора… ………………………39
   3. Разработка руководства пользователя …………………………….42

Заключение…………………………………………………………………….48

Литература……………………………………………………………………..49

# ТЕХНИЧЕСКОЕ ЗАДАНИЕ

* 1. **Наименование**

Калькулятор с расширенными возможностями на основе Windows API

* 1. **Основание для разработки**

Практически во всех сферах разработки необходимо производить быстрые арифметические и тригонометрические вычисления. Но в большинстве видов простых калькуляторов пользователь не может производить вычисления над тригонометрическими функциями, хотя зачастую это необходимо сделать в кротчайшие сроки. Вместо того, чтобы сразу начать вычислять на главном экране калькулятора, пользователь вынужден искать другой вид калькулятора или возиться в настройках, чтобы появилась возможность взаимодействия с тригонометрическими функциями. Но не хотелось бы делать лишних действий в условиях временных ограничений. Поэтому необходимо реализовать такой калькулятор, в котором пользователь сможет, не теряя времени, производить вычисления как обычные, так и тригонометрические.

Для того, чтобы реализовать наиболее эффективный, продуктивный и удобный калькулятор, необходимо использовать интерфейсы взаимодействия с операционной системой через системные библиотеки и WinAPI функции.

* 1. **Исполнитель**

Студент группы ИУК5-42Б Кузнецов Р.С.

* 1. **Цель разработки**

Целью курсовой работы является формирование практических навыков по разработке и реализации программного приложения с использованием интерфейса прикладного программирования (АРI) операционных систем.

Задачи проектирования:

1. овладение первичными навыками ведения научно-исследовательской, проектной и производственно-технологической деятельности, развитие творческих способностей индивидуально для каждого студента;
2. подготовка к выполнению выпускной квалификационной работы;
3. усвоение методов грамотного ведения, оформления и редактирования технической документации.

Целью работы является разработка калькулятора с расширенными возможностями на основе системных библиотек и WinAPI функций.

* 1. **Содержание работы**
     1. **Задачи, подлежащие решению:**

1. исследование предметной области;

изучение принципов системного программирования и работы WinAPI функций;

разработка калькулятора;

тестирование разработанного программного обеспечения;

1. анализ полученных результатов;
2. подготовка расчетно-пояснительной записки и графических листов;
3. подготовка презентации и речи для защиты курсовой работы;
4. защита курсовой работы.

**1.5.2 Требования к архитектуре АСОИ**

К архитектуре предъявляются следующие требования:

каждая функция на калькуляторе должна быть представлена в виде отдельного класса;

приложение должно представлять собой многопоточное приложение. Работа пользовательского интерфейса должна быть реализована в основном потоке, а все остальные функции калькулятора в отдельных потоках;

**1.5.3 Требования к составу программных компонентов**

Программный комплекс должен состоять из исполняемого файла PE формата с расширением .exe, реализующего калькулятор с расширенными возможностями.

**1.5.4 Требования к прикладным программам**

Для работы программного комплекса необходимы:

1. Microsoft Windows 8/10;
2. Microsoft Visual Studio 2019;
3. минимальный набор драйверов, обеспечивающих   
   работоспособность ПК.

**1.5.5 Требования к входным/выходным данным**

Входные данные:

1. арифметические выражения в текстовом формате;
2. тригонометрические выражения в текстовом формате.

Выходные данные:

1. Результат арифметических или тригонометрических операции в текстовом формате.

**1.5.6 Требования к временным характеристикам**

Требования к временным характеристикам программы   
не предъявляются.

**1.5.7 Требования к составу технических средств**

Для функционирования системы необходимы:

* процессор: любой 64- или 32- битный процессор
* RAM: 1 Гбайт;
* HDD: 50 Мбайт;
* видеокарта: поддержка Microsoft DirectX;
* монитор;
* клавиатура;
* мышь.
  1. **Этапы разработки**

разработка пользовательского интерфейса с помощью Windows API функций;

реализация функционала калькулятора с расширенными возможностями в отдельных потоках;

тестирование разработанного программного обеспечения;

исправление выявленных ошибок.

* 1. **Техническая документация, предъявляемая по окончании работы**

По окончанию работы предъявлена расчетно-пояснительная записка в состав которой входят:

* техническое задание;
* научно-исследовательская часть;
* проектно-конструкторская часть;
* проектно-технологическая часть.

Также должна быть предоставлена графическая часть работы, выполненная формате А1 на 2 листах, в которую входят:

* демонстрационные чертежи;
* алгоритмические схемы.

**1.8 Дополнительные условия**

язык программирования С/С++;

тип приложения – оконное;

1. интерфейс должен предоставлять пользователю удобный и эффективный калькулятор с расширенными возможностями;

# НАУЧНО ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ

## **Постановка задачи проектирования**

Практически во всех сферах разработки необходимо производить быстрые арифметические и тригонометрические вычисления. В большинстве видов простых калькуляторов пользователь не может производить вычисления над тригонометрическими функциями, хотя зачастую это необходимо сделать в кротчайшие сроки. Вместо того, чтобы сразу начать вычислять на главном экране калькулятора, пользователь вынужден искать другой вид калькулятора или разбираться в настройках, чтобы появилась возможность взаимодействия с тригонометрическими функциями. Во время усердной работы не хотелось бы делать лишних действий в условиях временных ограничений. Поэтому необходимо реализовать такой калькулятор, в котором пользователь сможет, не теряя времени, производить вычисления как обычные, так и тригонометрические.

Для того, чтобы реализовать наиболее эффективный, производительный и удобный калькулятор, необходимо использовать интерфейсы взаимодействия с операционной системой через системные библиотеки и WinAPI функции.

* 1. **Описание предметной области.**

Калькуля́тор (лат. calculātor «счётчик») — электронное вычислительное устройство для выполнения операций над числами или алгебраическими формулами.

Калькулятор заменил механические вычислительные устройства, такие, как абаки, счёты, логарифмические линейки, механические или электромеханические арифмометры, а также математические таблицы (прежде всего — таблицы логарифмов).

В зависимости от возможностей и целевой сферы применения калькуляторы делятся на простейшие, бухгалтерские, инженерные (научные), финансовые. В отдельные классы обычно выделяют программируемые калькуляторы, дающие возможность выполнения сложных вычислений по предварительно заложенной программе, а также графические — поддерживающие построение и отображение графиков. Специализированные калькуляторы предназначены для выполнения вычислений в достаточно узкой сфере (финансовые, строительные и т. п.)

По исполнению калькуляторы могут быть настольными или компактными (карманными). Отдельные модели имеют интерфейсы для подключения персонального компьютера, печатающего устройства, внешнего модуля памяти или иных внешних устройств. Современные персональные компьютеры, сотовые телефоны, КПК и даже наручные часы могут иметь программы, выполняющие функции калькулятора.

Калькуляторы можно разделить на следующие виды:

* карманные — калькуляторы небольшого размера, которые можно брать с собой;
* настольные — калькуляторы чуть большего размера, которые удобнее использовать, например, на рабочем месте, людям, производящим большое количество расчетов;
* калькуляторы с печатью — настольные калькуляторы со встроенным печатным устройством, которое выводит производимые вычисления, промежуточные итоги, графики на бумажную ленту;
* онлайн калькуляторы.

***Простые калькуляторы****:*

Простые калькуляторы выполняют обычные арифметические расчеты (сложение, вычитание, деление и умножение) и, как правило, несколько дополнительных функций, таких как расчет процентов и извлечение из квадратного корня. Такие калькуляторы обычно небольшого размера и веса.

***Бухгалтерские калькуляторы****:*

Как следует из названия этого типа калькуляторов, они предназначены для использования бухгалтерами и кассирами. В целом же основная их функция — это профессиональные расчеты с денежными суммами.

Бухгалтерские калькуляторы преимущественно выполнены в настольном варианте, оснащены крупными клавишами, большего размера дисплеем, могут иметь клавиши типа «000», поддерживают большее, чем у других калькуляторов, число знаков. Такие калькуляторы зачастую имеют функции округления, а также дополнительные бухгалтерские функции: «проверка и коррекция» (позволяет не только просмотреть выполненные действия, но и внести в них изменения), «покупка-продажа-прибыль» (вычисление себестоимости, цены или маржи по двум параметрам), вычисление надбавок, расчет и добавление/ вычитание НДС, подсчет итога по всем операциям, конвертация валюты.

***Инженерные калькуляторы****:*

Более сложный тип калькуляторов, разработанный для различных по сложности инженерных и научных расчетов. Такие калькуляторы способны делать расчеты с приоритетами операций и скобками, иногда позволяют делать расчеты с дробями, делают вычисления элементарных функций, а также поддерживают множество других расчетов (статистические, тригонометрические и пр.).

Инженерный калькулятор может поддерживать более сотни функций, из-за чего обычно содержит большее количество клавиш, зачастую двойного или тройного значения.

***Программируемые калькуляторы****:*

Программируемые калькуляторы по их возможностям можно назвать сложными инженерными калькуляторами. Они способны выполнять те же функции, а также дополнительно делать повторные сложные вычисления, выполнять создаваемые пользователями программы. Такие калькуляторы имеют более 10 регистров памяти, зачастую имеют интерфейсы для подключения к внешним устройствам, таким как компьютер. Также оснащены внешней памятью, исполнительными устройствами и аппаратными датчиками. Наиболее функциональные программируемые калькуляторы можно даже отнести к простым портативным компьютерам, но их основное отличие от последних заключается в узкой специализации выполняемых действий.

***Графические калькуляторы****:*

Все графические калькуляторы относятся к программируемым, но их отличает наличие графического экрана. Такие калькуляторы способны поддерживать команды, отображающие графики функций, а также могут выводить на экран рисунки.

Таким образом, современный рынок калькуляторов богат на устройства с различной формой и функциональностью, среди которых каждый сможет найти для себя наиболее подходящий вариант.

* 1. **Анализ аналогов и прототипов**

Для того, чтобы сделать хороший проект, необходимо проанализировать и сравнить уже существующие, самые часто используемые виды калькуляторов, и выделить их основные преимущества и недостатки. Воспользовавшись недочётами других разработчиков, не повторим их в своем проекте и сделаем более удобный и правильный калькулятор.

Посмотрим на преимущества и недостатки обычных(простых) калькуляторов.



Рисунок 1 - Интерфейс простого калькулятора на IPhone

Преимущества:

* простой и понятный интерфейс, не требующий дополнительных(прикладных) знаний;

Недостатки:

* невозможность подсчитывания более сложных выражений, таких как: квадратный корень, возведение в степень и др.

Далее проанализируем инженерный калькулятор Windows.

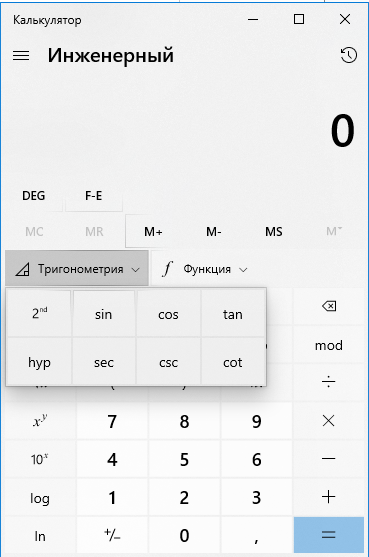


Рисунок 2 - Интерфейс инженерного калькулятора Windows

Преимущества:

* приятный и удобный интерфейс;
* возможность работы с тригонометрическими функциями;
* вычисление остатка от деления.

Недостатки:

* сложность написания программного кода для выполнения всех необходимых функций;
* при необходимости работы с тригонометрическими функциями, не интуитивная и неудобная работа с интерфейсом калькулятора. (Например, если необходимо подсчитать Sin5°, то машинально пользователь захочет сначала нажать на кнопку Sin, далее на цифру 5, но при таком вводе вы не получим желаемый результат. Чтобы получить корректный ответ, необходимо сначала нажать на цифру 5, потом на кнопку Sin. Еще один пример неудобства и неэффективности работы с тригонометрией: если необходимо подсчитать Sin(π/2), то пользователь должен выполнить следующий ряд действий:

1. Перевести калькулятор в режим работы с радианами, нажав на кнопку DEG;
2. Нажать на кнопку открывающейся скобки (;
3. Нажать на кнопку π;
4. Нажать на кнопку деления ÷;
5. Нажать на цифру 2;
6. Нажать на кнопку закрывающейся скобки );
7. Нажать на раскрывающееся меню с тригонометрическими функциями;
8. Нажать на кнопку Sin.

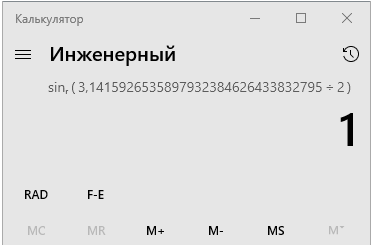


Рисунок 3 - Неудобная работа с интерфейсом калькулятора Windows

8 действий для подсчитывания элементарной тригонометрической функции)

Калькулятор-виджет Glossy Calculator.

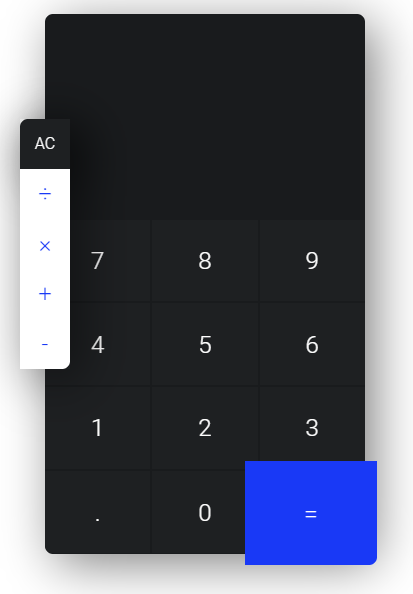


Рисунок 4 - Интерфейс калькулятора Glossy Calculator

Преимущества:

* стильный и красивый интерфейс;
* простота написания программного кода.

Недостатки:

* малая функциональность калькулятора.

Калькулятор для Windows – Aero Calculator.

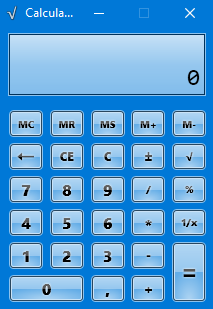


Рисунок 5 - Интерфейс калькулятора Aero Calculator

Преимущества:

* арифметические вычисления и работа с простыми функциями;
* сохранение результата при случайном закрытии;
* сохранение значений в памяти.

Недостатки:

* сайт и дистрибутив на английском;
* для некоторых пользователей, раздражающий, ярко-синий цвет интерфейса.

Проанализировав аналоги и прототипы, можно сделать вывод, что существующие калькуляторы не позволяют удобно и быстро решать поставленные задачи, поэтому необходимо разработать собственный калькулятор, который бы эффективно вычислял арифметические и тригонометрические выражения.

## **Перечень задач, подлежащих решению в процессе разработки.**

* разработка пользовательского интерфейса с использованием WinApi;
* разработка базового функционала калькулятора;
* разработка функций для решения уравнений;
* привязка функций калькулятора к пользовательскому интерфейсу;
* тестирование разработанного приложения;
* исправление выявленных ошибок.
  1. **Обоснование выбора инструментов и платформы для разработки.**

Для успешной разработки проекта, используется операционная система Windows 10. Windows 10 — [операционная система](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) для [персональных компьютеров](https://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D1%81%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80) и [рабочих станций](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B1%D0%BE%D1%87%D0%B0%D1%8F_%D1%81%D1%82%D0%B0%D0%BD%D1%86%D0%B8%D1%8F), разработанная корпорацией [Microsoft](https://ru.wikipedia.org/wiki/Microsoft" \o "Microsoft) в рамках семейства [Windows NT](https://ru.wikipedia.org/wiki/Windows_NT).

Данная платформа предоставляет широкий набор инструментов для работы с ОС Windows, например, язык программирования Си с использованием базовых функций интерфейсов программирования приложений операционных систем Microsoft Windows – Windows API. WinApi – это набор функций, работающих под управлением ОС Windows. С помощью WinAPI можно создавать различные оконные процедуры, диалоговые окна, программы и даже игры, а также работать с операционной системой на прямую.

Согласно Statcounter.com, общий рейтинг операционных систем, показывает, что лидером является Windows, который установлен на 76,58% всех компьютеров, OS X — 18,93% и Linux — 1,62%. Платформа Windows является самой популярной и доступной, также для этой платформы создано много удобных средств, для облегчения выполнения проекта.

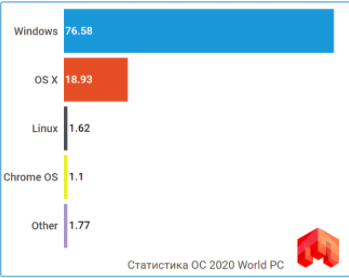


Рисунок 6 - Статистика использования операционных систем в мире

В качестве среды разработки используется Visual Studio 2019. Интегрированная среда разработки Visual Studio — это стартовая площадка для написания, отладки и сборки кода, а также последующей публикации приложений. Интегрированная среда разработки (IDE) представляет собой многофункциональную программу, которую можно использовать для различных аспектов разработки программного обеспечения. Помимо стандартного редактора и отладчика, которые существуют в большинстве сред IDE, Visual Studio включает в себя компиляторы, средства автозавершения кода, графические конструкторы и многие другие функции для упрощения процесса разработки.

Ниже перечислены некоторые популярные возможности Visual Studio, которые помогут вам повысить продуктивность разработки программного обеспечения.

* волнистые линии и [быстрые действия](https://docs.microsoft.com/ru-ru/visualstudio/ide/quick-actions?view=vs-2019);

Волнистые линии обозначают ошибки или потенциальные проблемы кода прямо во время ввода. Эти визуальные подсказки позволяют устранять проблемы немедленно и не ждать, пока ошибка будет обнаружена во время сборки или запуска программы. Если навести указатель мыши на волнистую линию, на экран будут выведены дополнительные сведения об ошибке. Кроме того, в поле слева может появляться значок лампочки с быстрыми действиями по устранению ошибки.

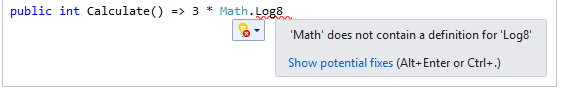


Рисунок 7 - Волнистые линии и быстрые действия

* [поиск в Visual Studio](https://docs.microsoft.com/ru-ru/visualstudio/ide/visual-studio-search?view=vs-2019).

Среда Visual Studio может показаться сложной, ведь там столько разных меню, параметров и свойств. Чтобы быстро находить функции интегрированной среды разработки и элементы кода, в Visual Studio представлен единый компонент поиска (**CTRL**+**Q**).

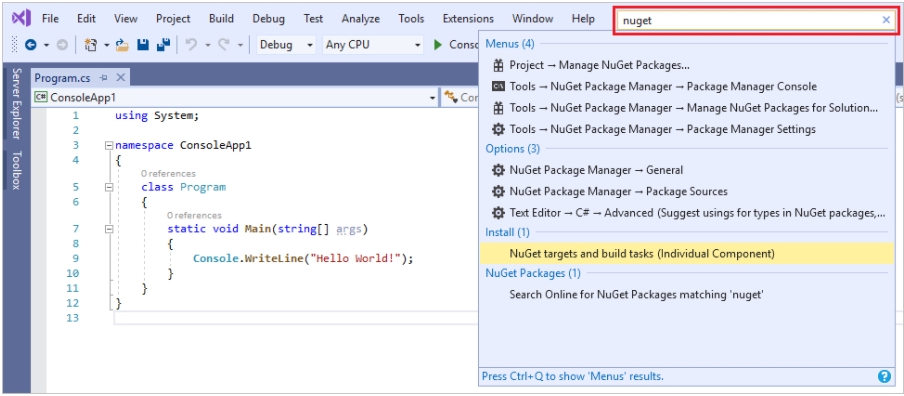


Рисунок 8 - Поиск в Visual Studio

Согласно опросу, проведенного среди 3240 разработчиков в феврале 2018 организацией Standard C++, разработчики на C++ выделили такие средства разработки, как наиболее часто используемые:

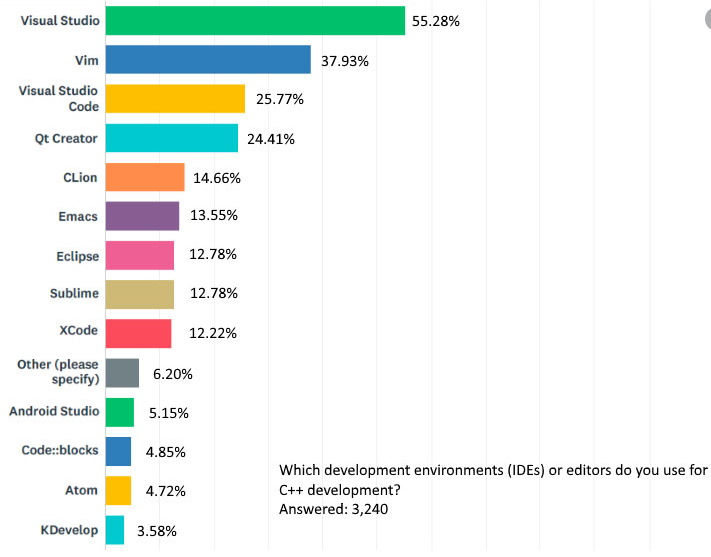


Рисунок 9 - Статистика использования платформ для разработки систем в мире

Первое место среди IDE у Visual Studio с 55.28% использующих его разработчиков.

Также, Visual Studio 2019 обладает хорошим отладчиком, с помощью которого можно легко отследить ошибки в коде и без проблем их исправить, позволяет удобно и быстро создавать проекты и работать с ними.

По сравнению со средой разработки Eclipse, Visual Studio имеет более простой интерфейс в использовании.

Для разработки проекта можно использовать следующий инструмент:

* язык программирования C++.

C++ - компилируемый, статически типизированный язык программирования общего назначения.

Поддерживает такие [парадигмы программирования](https://ru.wikipedia.org/wiki/%D0%9F%D0%B0%D1%80%D0%B0%D0%B4%D0%B8%D0%B3%D0%BC%D1%8B_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F), как [процедурное программирование](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D0%B4%D1%83%D1%80%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5), [объектно-ориентированное программирование](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5), [обобщённое программирование](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D0%BE%D0%B1%D1%89%D1%91%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5). Язык имеет богатую стандартную библиотеку, которая включает в себя распространённые контейнеры и алгоритмы, ввод-вывод, регулярные выражения, поддержку многопоточности и другие возможности. C++ сочетает свойства как [высокоуровневых](https://ru.wikipedia.org/wiki/%D0%92%D1%8B%D1%81%D0%BE%D0%BA%D0%BE%D1%83%D1%80%D0%BE%D0%B2%D0%BD%D0%B5%D0%B2%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F), так и [низкоуровневых языков](https://ru.wikipedia.org/wiki/%D0%9D%D0%B8%D0%B7%D0%BA%D0%BE%D1%83%D1%80%D0%BE%D0%B2%D0%BD%D0%B5%D0%B2%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F). В сравнении с его предшественником — языком [C](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8_(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)) — наибольшее внимание уделено поддержке [объектно-ориентированного](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) и [обобщённого программирования](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D0%BE%D0%B1%D1%89%D1%91%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5).

C++ широко используется для разработки программного обеспечения. Область его применения включает создание [операционных систем](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0), разнообразных прикладных программ, [драйверов](https://ru.wikipedia.org/wiki/%D0%94%D1%80%D0%B0%D0%B9%D0%B2%D0%B5%D1%80) устройств, приложений для встраиваемых систем, высокопроизводительных серверов, а также игр. Существует множество реализаций языка C++, как бесплатных, так и коммерческих и для различных платформ. Например, на платформе [x86](https://ru.wikipedia.org/wiki/X86) это [GCC](https://ru.wikipedia.org/wiki/GNU_Compiler_Collection), [Visual C++](https://ru.wikipedia.org/wiki/Visual_C%2B%2B), [Intel C++ Compiler](https://ru.wikipedia.org/wiki/Intel_C%2B%2B_Compiler), [Embarcadero (Borland) C++ Builder](https://ru.wikipedia.org/wiki/Embarcadero_C%2B%2B_Builder) и другие. C++ оказал огромное влияние на другие языки программирования, в первую очередь на [Java](https://ru.wikipedia.org/wiki/Java" \o "Операционная система) и [C#](https://ru.wikipedia.org/wiki/C_Sharp).

Достоинства языка программирования С++:

* эффективность. Язык спроектирован так, чтобы дать программисту максимальный контроль над всеми аспектами структуры и порядка исполнения программы. Ни одна из языковых возможностей, приводящая к дополнительным накладным расходам, не является обязательной для использования — при необходимости язык позволяет обеспечить максимальную эффективность программы;
* имеется возможность работы на низком уровне с памятью, адресами;
* пользовательские функции-операторы позволяют кратко и ёмко записывать выражения над пользовательскими типами в естественной алгебраической форме;
* высокая совместимость с языком Си, позволяющая использовать весь существующий Си-код.

Недостатки:

* операция присваивания обозначается как =, а операция сравнения как ==, их легко спутать;
* некоторые преобразования типов неинтуитивны;
* операции присваивания (=), инкрементации (++), декрементации (--) и другие возвращают значение. В сочетании с обилием операций это позволяет, хотя и не обязывает, создавать трудночитаемые выражения.

1. **ПРОЕКТНО-КОНСТРУКТОРСКАЯ ЧАСТЬ**
2. **Разработка структуры приложения**

Для разработки приложения использовалась среда Microsoft Visual Studio 2019, включающая в себя широкий набор функциональных инструментов.

Для представления принципа работы программы ниже приведена ее

UML диаграмма вариантов использования (рисунок 1).

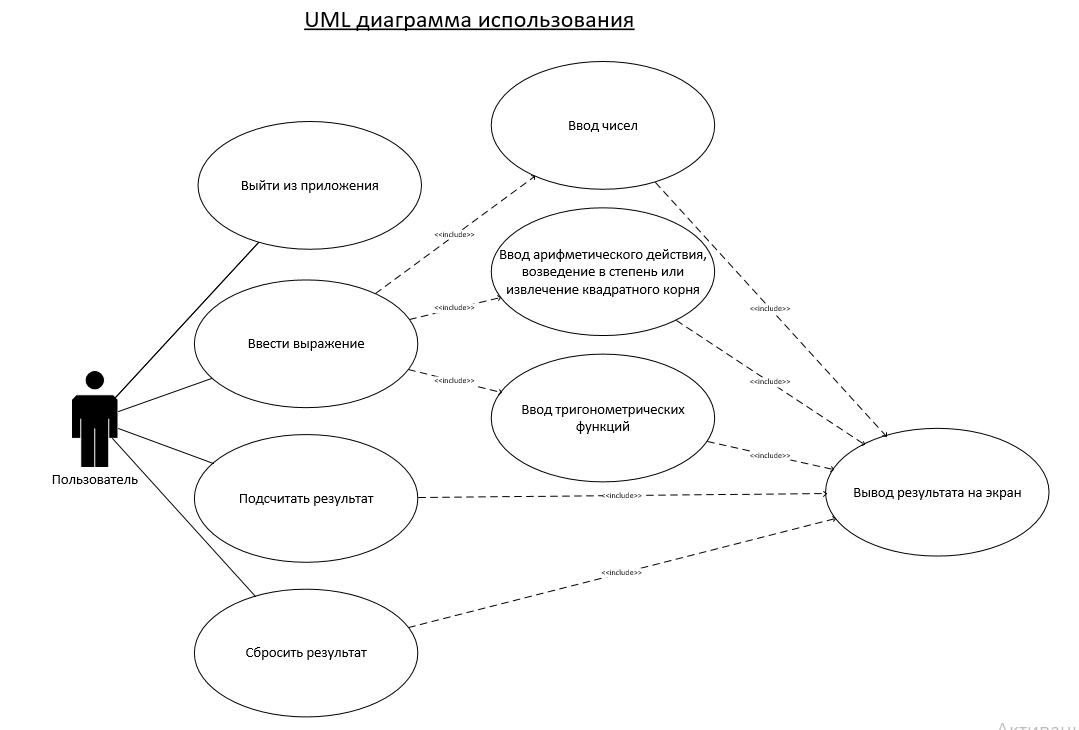


Рисунок 1 - Диаграмма использования программы «Калькулятор»

При запуске приложения у пользователь может выбрать, какое действие ему совершить.

В любой момент работы с приложением, пользователь может выйти из него.

Также, пользователь может ввести необходимо для вычисления выражение. Ввод состоит из чисел, арифметических действий, возведения в степень, извлечения квадратного корня, а также тригонометрических функций. Если возникла ошибка при вводе, то пользователь может сбросить результат.

После ввода, пользователь может подсчитать результат выражения и продолжить работу с полученным результатом или сбросить его.

В процессе работы с приложением, действия (ввод, вычисление, сброс), которые совершает пользователь, будут выводится на экран.

Общее назначение программного средства - выполнение арифметических и тригонометрических операций для использования в производственном, учебном процессах и повседневной жизни.

Реализуемая задача состоит в том, чтобы производилась корректная работа с тригонометрическими функциями, также чтобы при выборе действия выполнялась определенная операция, и имелась возможность сбросить полученный результат или же продолжить расчеты с данным результатом.

1. **Используемые функции Win32API**

Для создания программы «Калькулятор с расширенными возможностями», необходимо реализовать алгоритмы, позволяющие работать с арифметическими действиями, а также с тригонометрическими функциями. Необходимо организовать возможность сброса полученных результатов.

Блок-схемы алгоритмов представлены в Приложении 1.

Основные функции, реализуемые в программе из библиотеки User32.lib:

* LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam) - производит обработку сообщений для окна (получает сообщения, которые Windows будет отсылать окну) :
  + hWnd – главное окно приложения,
  + message – сообщение, посылаемое операционной системе,
  + wParam, lParam – параметры сообщений, для последующего обращения.
* HCURSOR LoadCursor(NULL, IDC\_ARROW) - загружает заданный ресурс курсора из исполняемого файла:
  + NULL - дескриптор экземпляра модуля, исполняемый файл которого содержит загружаемый курсор.
  + IDC\_ARROW - cтандартная стрелка.
* HICON LoadIcon(GetModuleHandle(NULL), MAKEINTRESOURCE(MAIN\_ICON)) - загружает указанный ресурс значка из исполняемого файла:
  + GetModuleHandle - извлекает дескриптор указанного модуля;
  + MAKEINTRESOURCE - преобразует целочисленное значение в тип ресурса
  + MAIN\_ICON – предопределенная константа, служит для обозначения уникальности созданной иконки (.ico)
* ATOM RegisterClass(&wc) - регистрирует класс окна для последующего использования при вызове функции CreateWindow:
  + wc – структура, содержащая атрибуты окна.
* HWND CreateWindow(WMAIN\_CLASSNAME, WMAIN\_CAPTION, WS\_OVERLAPPED | WS\_CAPTION | WS\_SYSMENU | WS\_MINIMIZEBOX | WS\_MAXIMIZEBOX | WS\_VISIBLE, CW\_USEDEFAULT, CW\_USEDEFAULT, 320, 419, HWND\_DESKTOP, NULL, hInstance, 0) – создает основное окно приложения:
  + WMAIN\_CLASSNAME – зарегистрированное имя класса (Calculator),
  + WMAIN\_CAPTION – имя окна (Калькулятор),
  + WS\_OVERLAPPED – определяет базовый стиль окна - перекрывающееся окно,
  + WS\_CAPTION – заголовок,
  + WS\_SYSMENU – системное меню,
  + WS\_MINIMIZEBOX – кнопка минимизации размера окна,
  + WS\_MAXIMIZEBOX – кнопка максимального увеличения размера окна,
  + WS\_VISIBLE – первоначально, окно не отображается,
  + CW\_USEDEFAULT – горизонтальная позиция окна,
  + CW\_USEDEFAULT – вертикальная позиция окна,
  + 320 – ширина окна,
  + 419 – высота окна,
  + HWND\_DESKTOP – дескриптор родительского окна,
  + NULL - дескриптор меню или ID дочернего окна,
  + hInstance - дескриптор экземпляра приложения (это некий код оконной процедуры, идентификатор, по которой ОС будет отличать её от остальных окон)
  + 0 - указатель на данные создания окна (указывает на значение, переданное в параметре lParam сообщения WM\_CREATE).
* BOOL ShowWindow (hWnd, nCmdShow) - устанавливает состояние показа определяемого окна:
  + hWnd - главное окно приложения,
  + nCmdShow - состояние показа окна.
* BOOL UpdateWindow(hWnd) - обновляет рабочую область заданного окна, отправляя сообщение WM\_PAINT окну, если регион обновления окна не пуст:
  + hWnd - главное окно приложения.
* BOOL GetMessage(&msg, NULL, 0, 0) - извлекает сообщение из очереди сообщений вызывающего потока и помещает его в заданную структуру:
  + msg - указатель на структуру MSG, которая принимает информацию из очереди сообщений потока,
  + NULL – дескриптор окна,
  + 0 - определяет целочисленную величину самого маленького значения сообщения, которое будет извлечено,
  + 0 - определяет целочисленную величину самого большого значения сообщения, которое будет извлечено

Обе величины нулевые, следовательно, функция GetMessage возвращает все доступные сообщения (то есть никакой фильтрации в диапазоне значений не выполняется).

* BOOL TranslateMessage(&msg) - переводит сообщения виртуальных клавиш в символьные сообщения:
  + &msg - указатель на структуру MSG, которая содержит информацию о сообщении извлеченную из очереди сообщений вызывающего потока при помощи использования функции GetMessage.
* LRESULT DispatchMessage(&msg) - распределяет сообщение оконной процедуре, используется, чтобы доставить сообщение, извлеченное функцией GetMessage:
  + &msg - указатель на структуру MSG, которая содержит сообщение.
* LRESULT SendMessage(button[21], BM\_SETCHECK, BST\_CHECKED, 0) - отправляет заданное сообщение окну:
  + button[21] - дескриптор окна, оконная процедура которого примет сообщение (в данном случае радио - кнопка «RAD»);
  + BM\_SETCHECK - определяет сообщение, которое будет отправлено (возвращает состояние флажка для переключателя);
  + BST\_CHECKED – определяет дополнительную конкретизирующую сообщение информацию (устанавливает состояние флажка как отмеченное);
  + 0 - определяет дополнительную конкретизирующую сообщение информацию.
* WORD LOWORD(wParam) - выделяет из 32-битового целочисленного значения младшее слово:
  + wParam – преобразуемое значение;
* WORD HIWORD(wParam) - выделяет из 32-битового целочисленного значения старшее слово:
  + wParam – преобразуемое значение;
* int GetWindowText(hEdit, buf, sizeof(buf)) - копирует текст окна в буфер:
  + hEdit - дескриптор окна управления с текстом (окно ввода-вывода);
  + buf - адрес буфера для текста;
  + sizeof(buf) - максимальное число символов для копирования;
* BOOL SetWindowText(hEdit, buf) - изменяет текст окна:
  + hEdit - дескриптор окна, текст которого должен быть изменен (окно ввода-вывода);
  + buf – адрес буфера для текста.
* void PostQuitMessage(0) - указывает системе, что поток сделал запрос на то, чтобы завершить свою работу (выйти). Это используется в ответ на сообщение WM\_DESTROY:
  + 0 - определяет код завершения прикладной программы.
* LRESULT DefWindowProc(hWnd, message, wParam, lParam) - вызывается оконной процедурой по умолчанию, чтобы обеспечить обработку по умолчанию любого сообщения окна, которые приложение не обрабатывает. Эта функция гарантирует то, что обрабатывается каждое сообщение:
  + hWnd – главное окно приложения, которое получает сообщение;
  + message – передаваемое сообщение;
  + wParam, lParam - определяют дополнительную информацию о сообщении.

1. **Разработка архитектуры приложения**

Приложение разрабатывалась по архитектуре многоуровневого шаблона.

Данное приложение использует многоуровневый архитектурный шаблон. Данный шаблон используется для структурирования программ, которые можно разложить на группы неких подзадач, находящихся на определенных уровнях абстракции. Каждый слой представляет службы для следующего, более высокого слоя. На рисунке 2 представлена схема приложения.

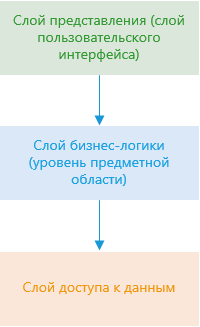


Рисунок 2 - Схема построения приложения по многоуровневому шаблону

Слой пользовательского интерфейса в разрабатываемом приложении представлен в файле Calculator.cpp, в котором указаны следующие параметры:

* главное окно. Расположение кнопок, поля ввода-вывода и параметрами главного окна;
* иконка приложения при сборке проекта.

Слой бизнес-логики описан основным исполняемым файлом Calculator.cpp и файлом заголовков resource.h. Исполняемый файл Calculator.cpp состоит из функции обработки нажатия клавиш. Файл заголовков состоит из логики вычислений и вывода результата на экран.

Слой данных представляется в виде символьных массивов, в которых хранятся введенные и вычисленные выражения.

1. **Разработка алгоритмов приложения**

Основной алгоритм приложения – непосредственное вычисление арифметических и тригонометрических выражений и последующий вывод на экран результата вычислений.

Для этого используется функция TCHAR\* Calculate(TCHAR\* buf) , где buf – это глобальный строчный массив символов, который был заполнен пользователем числами и знаками с помощью предыдущей функции.

Сперва, определяются приоритеты знаков, с помощью встроенной функции “map” из библиотеки “map”.

После определения приоритетов, необходимо перенести данные из символьного массива buf[256] в строку basic\_string, а после этого в строку потока basic\_stringstream.

Далее, используя basic\_stringstream, символы поочередно переводятся в числа типа double. После получения чисел, данная функция определяет какой знак действия ввел пользователь, и в зависимости от этого выполняются необходимые вычисления.

Последний шаг в функции – это вывод на экран вычисленного выражения.

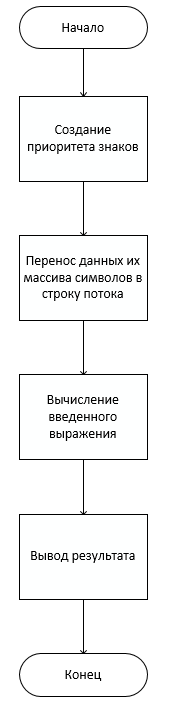


Рисунок 3 - Блок-схема функции Calculate

1. **Разработка интерфейса взаимодействия пользователя с системой**

Основная и самая главная часть приложения – создание окна и дальнейшее взаимодействие с ним.

Для этого используется функция

LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam) , где

hWnd – дескриптор окна,

message – сообщение, посылаемое операционной системе,

wParam, lParam – параметры сообщений, для последующего обращения.

В данной функции взаимодействие с окном происходит с помощью множественного условного оператора Switch.

В оператор передается переменная message, в зависимости от значения которой будут выполняться определенные действия.

При message = WM\_CREATE, создается интерфейс взаимодействия пользователя с приложением (окно, кнопки) с помощью встроенной функции CreateWindow.

Message = WM\_COMMAND становится при нажатии пользователем на любую кнопку интерфейса, далее, в зависимости от того, какая именно кнопка была нажата, в программе выполняются определенные действия, например, ввод чисел, знаков, подсчитывание результата.

Если пользователь нажмет на крестик в правом верхнем углу программы, то переменной message присвоится значение WM\_DESTROY и произойдет удаление окна с экрана (программа закроется).

Message = default – это обработчик программы по умолчанию, для того чтобы обеспечить обработку любого сообщения окна, которое приложение не обрабатывает.

Таким образом, для того чтобы производилась корректная работа с тригонометрическими функциями, при выборе действия выполнялась определенная операция, и имелась возможность сбросить полученный результат или же продолжить расчеты с данным результатом необходимо реализовать определенные алгоритмы: алгоритм непосредственного вычисления выражений и алгоритм взаимодействия пользователя с интерфейсом приложения. Для правильной работы приложения, необходимо использовать многоуровневый архитектурный шаблон.

1. **ПРОЕКТНО-ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ**

**4.1. Тестирование и отладка макета рабочей программы.**

Одной из самых важных функций любой программы является ввод и вывод данных.

Выводимые данные — это то, что сообщается пользователю. Входные данные — это то, что пользователь сообщает программе.

Выводимые данные в программе представлены в виде графического отображения окна программы (рисунок 1):

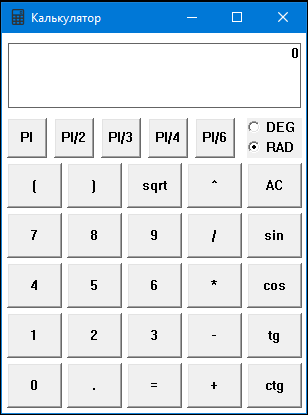


Рисунок 1 - Окно программы.

Входные данные представлены в виде программного кода, который необходимо выполнить при определенных действиях пользователя, а именно:

* нажатие клавиш клавиатуры;
* работа пользователя с кнопками в интерфейсе.

Тестирование приложения. Для начала необходимо запустить приложение «Калькулятор». После запуска приложения, открывается окно в виде калькулятора (рисунок 1). По умолчанию калькулятор настроен на режим работы с тригонометрией в градусах (при запуске приложения сразу включена радио - кнопка «RAD»).

Далее нужно ввести входные данные, например, выражение, которые проверит корректную работу со скобками: (2+2) \* 2 (рисунок 2)

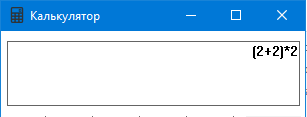


Рисунок 2 - Пример №1.

Если приложение обрабатывает выражения со скобками правильно, то после нажатия на кнопку «=», должен вывестись результат равный 8 (сначала выполняется действие в скобках, после производится умножение).

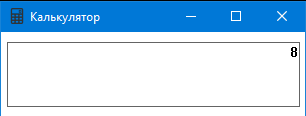


Рисунок 3 - Результат вычисления примера №1

Можно увидеть, что результат правильный и, следовательно, калькулятор корректно работает со скобками (рисунок 3.)

Также следует проверить работу калькулятора с исключительными ситуациями. Для этого нужно ввести 9 разделить на 0 (рисунок 4).



Рисунок 4 - Пример №2

Деление на ноль невозможно, поэтому калькулятор должен как-то отреагировать на ошибку в веденном выражении. На экран приложения выведется сообщение о неправильном вводе (рисунок 5).

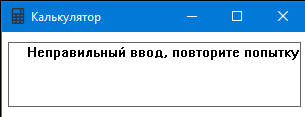


Рисунок 5 - Результат вычисления примера №2

Еще один пример исключительной ситуации. На ввод подается следующее выражение: sqrt(-9) (рисунок 6).

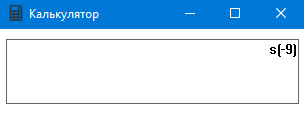


Рисунок 5 - Пример №3

На данный пример калькулятор снова выведет ошибку (рисунок 6).

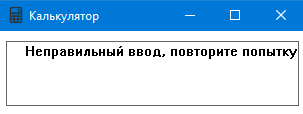


Рисунок 6 - Результат вычисления примера №3

Основная функция калькулятора – возможность работы с тригонометрическими функциями. Следовательно, необходимо проверить работу калькулятора с тригонометрией. Сперва нужно провести проверку в режиме работы в радианах (по умолчанию калькулятор работает в данном режиме). На ввод подается выражение Sin(PI/2) + Cos(PI) (рисунок 7), где S и C – соответственно синус и косинус, 1.57… и 3.14… - PI/2 и PI соответственно.

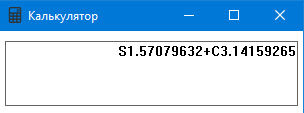


Рисунок 7 - Пример №4

В качестве результата калькулятор должен вывести 0, т.к. Sin(PI/2) равен 1, а Cos(PI) равен -1. Можно увидеть, что калькулятор провел корректные вычисления (рисунок 8).

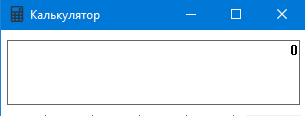


Рисунок 8 - Результат вычисления примера №4

Также следует проверить калькулятор в режиме работы в градусах. Пусть вводится следующий пример: 5 \* Tg(45°), также нужно переключить радио - кнопку «DEG» (рисунок 9).

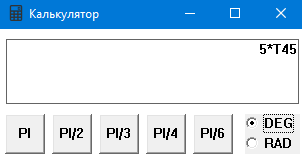


Рисунок 9 - Пример №5

В качестве результата, должно получится 5 (Tg(45°) равен единице). В данном режиме калькулятор также вычисляет правильно (рисунок 10).

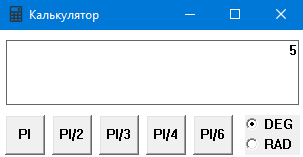


Рисунок 10 - Результат вычисления примера №5

Также следует проверить, сможет ли калькулятор вычислить сложное выражение, состоящее из нескольких действий и чисел. На ввод подается: Sin(PI/2) \* 9^2 \* 6/3 + 4 – 5 (рисунок 11).



Рисунок 11 - Пример №6

Результат должен быть таков: Sin(PI/2) равен 1, умножить на 9^2 это 81, умножить на 6/3 это 162, прибавить 4 это 166, вычесть 5 это 161. Даже сложные примеры калькулятор подсчитывает верно (рисунок 12).

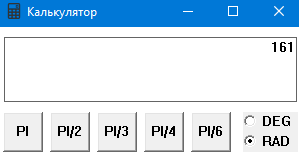


Рисунок 12 - Результат вычисления примера №6

Протестировав работу приложения используя разнообразные примеры, можно сделать вывод, что калькулятор работает корректно как с арифметическими, так и с тригонометрическими выражениями, а также с исключительными ситуациями.

**4.2. Разработка руководства администратора**

Для того, чтобы пользоваться программой «Калькулятор», необходимо сделать распаковку архива с калькулятором (рисунок 13).

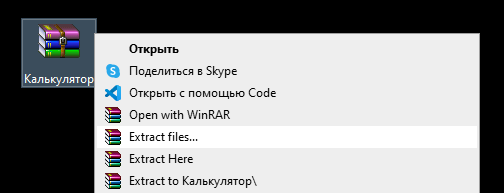


Рисунок 13 - Установка шаг №1

Если вы до этого ни разу не пользовались Visual Studio, то вам следует открыть папку «Install», которая находится внутри основной папки «Калькулятор» и установить необходимые драйвера для успешной работы с приложением (рисунок 14).

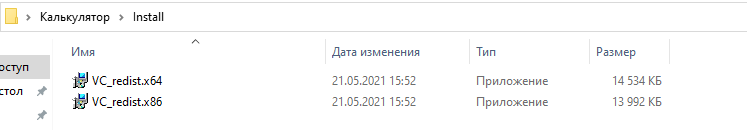


Рисунок 14 - Установка шаг №2

Приложение готово к работе! Все что осталось сделать – это открыть исполняемый файл с расширением .exe (Calculator.exe) (рисунок 15).

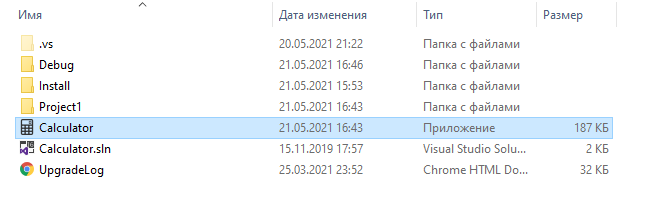


Рисунок 15 - Установка шаг №3

Если вы планируете пользоваться калькулятором на постоянной основе, то вам будет удобнее создать ярлык на рабочий стол, чтобы быстрее производить запуск приложения. Для этого необходимо нажать правой кнопкой мыши по исполняемому фалу Calculator.exe, после этого выбрать пункт «Создать ярлык» (рисунок 16).

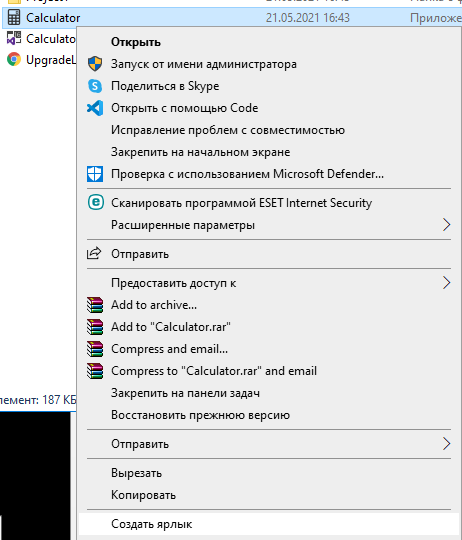


Рисунок 16 - Создание ярлыка калькулятора

После этого можно перенести ярлык на рабочий стол, и при двойном нажатии на него, будет открываться приложение (рисунок 17).

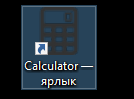


Рисунок 17 - Ярлык приложения

**4.3. Разработка руководства пользователя**

Порядок работы:

Запуск программы производится с помощью двойного щелчка мыши по файлу Calculator.exe или по его ярлыку.

После запуска открывается окно программы (рисунок 18).

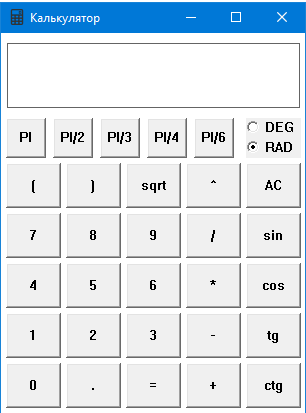


Рисунок 18 - Приложение после открытия

Далее в поле данных мы должны ввести операнд (постоянную величину) (рисунок 19),



Рисунок 19 - Ввод первого операнда

затем операцию, которую мы хотим выполнить (рисунок 20),

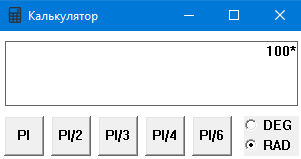


Рисунок 20 - Ввод знака действия

а затем второй операнд (рисунок 21).

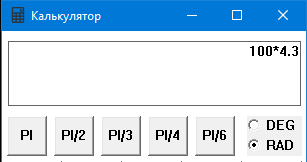


Рисунок 21 - Ввод второго операнда

При нажатии кнопки «=» получается результат вычисленного арифметического или тригонометрического выражения (рисунок 22).

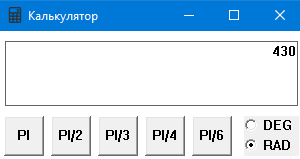


Рисунок 22 - Результат вычисления

Результат можно сбросить нажатием кнопки «AC» (рисунок 23),

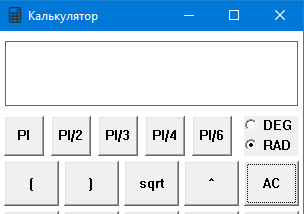
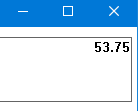
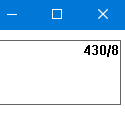


Рисунок 23 - Сброс результата

либо продолжить работу с ним (рисунки 24,25).



Рисунки 24 и 25 - Продолжение работы с ранее полученным результатом

Перечень элементов интерфейса:

* «AC» – очистка поля;
* «1», «2», «3», «4», «5», «6», «7», «8», «9», «0» – цифры;
* «.» – десятичный разделитель;
* «/» – деление;
* «\*» – умножение;
* «+» – сложение;
* «-» – вычитание;
* «DEG» – переключение режима калькулятора для работы в градусах;
* «RAD» – переключение режима калькулятора для работы в радианах;
* «PI» – постоянная величина (численно равна ≈ 3,14);
* «PI/2» – постоянная величина (численно равна ≈ 1,57);
* «PI/3» – постоянная величина (численно равна ≈ 1,05);
* «PI/4» – постоянная величина (численно равна ≈ 0,79);
* «PI/6» – постоянная величина (численно равна ≈ 0,52);
* «sin» – синус;
* «cos» – косинус;
* «tg» – тангенс;
* «ctg» – котангенс;
* «sqrt» – квадратный корень;
* «^» – возведение в степень;

По умолчанию, калькулятор находится в режиме работы с радианами (для тригонометрии) (рисунок 26).

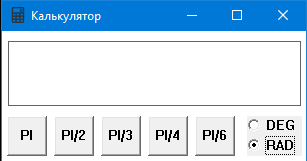


Рисунок 26 - Режим работы по умолчанию

Пользователь может в любой момент поменять режим работы калькулятора, нажав на соответствующую радио – кнопку с подписью «DEG» (рисунок 27).

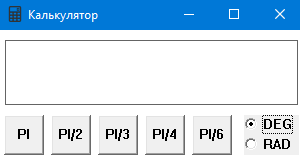


Рисунок 27 - Смена режима работы калькулятора

После этого калькулятор перейдет в режим работы с градусами.

Приложение разработано для обеспечения удобной и быстрой работы пользователя, предоставляя возможность выполнения арифметических и тригонометрических расчетов.

С точки зрения эргономики, в программе создан такой пользовательский интерфейс, который сделает работу комфортной и максимально упрощенной, чем обеспечит удовлетворенность пользователя от работы с программой.

Интерфейс калькулятора разработан исходя из принципа наименьшего возможного количества действий со стороны пользователя.

Основное поле калькулятора выполнено в белом цвете. Кнопки калькулятора выделены от основного фона серым цветом, что повышает зрительное удобство использования. Кнопки калькулятора крупные, по ним легко попасть при использовании мыши. Поле ввода выполнено в белом цвете. Цифры выводятся в темно-сером цвете и в крупном размере, что также удобно для использования. Надписи на кнопках выполнены в черном цвете.

Элементы интерфейса сгруппированы по значимости. Основные арифметические знаки расположены в одном ряду по вертикали. Математические функции упорядочены в один блок. Кнопка удаления данных «AС» расположена в верхнем правом углу, что позволяет избежать ошибочных нажатий.

Приложение допускает возможность ввода данных как с помощью мыши, так и с помощью клавиатуры. На иконке выполнен макет калькулятора, что позволяет легко найти приложение на рабочем столе.



Рисунок 28 - Иконка приложения

Таким образом, приложение может корректно вычислять арифметические и тригонометрические выражения. После проведения тестирования можно увидеть правильную обработку исключительных ситуаций. Провести успешную установку и дальнейшую работу с данным приложением не составит труда даже новым пользователям после прочтения руководств администратора и пользователя.

# ЗАКЛЮЧЕНИЕ

В ходе выполнения курсовой работы был разработан калькулятор с расширенными возможностями на основе WinAPI.

Работа выполнялась в несколько этапов: была разработана структура системы и реализовано прикладное оконное приложение на основе WinAPI.

Были сформированы навыки по разработке и реализации программного приложения с использованием интерфейса прикладного программирования (АРI) операционных систем.

В данной работе выполнены все поставленные задачи.

В будущем можно усовершенствовать разработанное приложение путем добавления дополнительного функционала (например, добавление кнопок памяти).

# ЛИТЕРАТУРА

1. Павловская Т.А. C/C++ . Процедурное и объектно-ориентированное программирование. Стандарт 3-го поколения. СПб.: Питер, 2015.

2. Платонов В.В. Программно-аппаратные средства защиты информации М.: Академия, 2014.

3. Бречка Д.М. Операционные системы: в 3 ч. Ч. 1. Пакетные файлы и управление компьютером: учебно-методическое пособие. Омск: Омский государственный университет им. Ф.М. Достоевского, 2012 [https://e.lanbook.com/book/75382?category\_pk=1554#book\_name](https://e.lanbook.com/book/75382?category_pk=1554%23book_name)

4. Вирт Н., Гуткнехт Ю. Разработка операционной системы и компилятора. Проект Оберон. М.: ДМК Пресс, 2012 [https://e.lanbook.com/book/39992?category\_pk=1554#book\_name](https://e.lanbook.com/book/39992?category_pk=1554%23book_name)

5. Сидоров В.Н., Сломинская Е.Н., Полникова Т.В., Макарова О.Ю. Оформление графической части выпускной квалификационной работы. Учебное пособие. М.: МГТУ им. Н.Э. Баумана, 2016.