

Министерство науки и высшего образования
Пензенский государственный университет
Кафедра «Вычислительная техника»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе
по курсу «Программирование»
на тему «Калькулятор»

Выполнил(а):
студентка группы 21ВВ2
Алёшина А.В.

Принял:
Голотенков Н.О.


07.06.22

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 Фейминой Ане Васильевне Группа ЗИЗ

Тема проекта: Разработка программы сложной структуры методом нисходящего программирования.

Исходные данные (технические требования) на проектирование

Приложение: универсальный калькулятор
Функции приложения: вычисления арифметических действий (сложение, вычитание, умножение, деление), возведение в степень, извлечение квадратного корня, вычисления основных тригонометрических функций, сохранение истории вычислений в файл, вывод данных на экран

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 " февраля

Дата защиты проекта " 06 " 2022 г.

Руководитель Голоженков Н.О.

Задание получил " 22 " февраля 2022 г.

Студент Аймиша Арина Васильевна

Содержание

Введение	5
1 Постановка задачи.....	7
2 Выбор решения.....	8
3 Описание разработки программы.....	9
4 Отладка и тестирование	11
5 Описание программы	12
6 Руководство пользователя	13
Заключение	17
Список используемых источников.....	18
Приложение А Листинги программы.....	19
Приложение В Схемы программы	26

Введение

В развитии электронно-вычислительной техники принято условно выделять несколько поколений. Последние обычно рассматривают во взаимосвязи с применяемой элементной базой.

Последняя информационная революция, связанная с изобретением микропроцессорной технологии и появлением персонального компьютера, выдвигает на первый план новую отрасль - информационную индустрию. Бурное развитие компьютерной техники и информационных технологий послужило толчком к развитию общества, построенного на использование различной информации и получившего название информационного общества.

Деятельность отдельных людей, групп, коллективов и организаций сейчас всё в большей степени начинает зависеть от их информированности и способности эффективно использовать имеющуюся информацию. Прежде чем провести какие-то действия, необходимо провести большую работу по сбору и переработке информации, её осмыслению и анализу. Отыскание рациональных решений в любой сфере требует обработки больших объёмов информации, что подчас невозможно без привлечения специальных технических средств.

Возрастание объёма информации особенно стало заметным в середине XX в. в ежедневно появляющемся потоке информации ориентироваться становилось всё труднее. Подчас выгоднее стало создать новый материальный или интеллектуальный продукт, нежели вести розыск аналога, сделанного ранее. Информационный кризис поставил общество перед необходимостью поиска путей выхода создавшегося положения. Внедрение ЭВМ, современных средств переработки и передачи информации в различные сферы деятельности послужило началом нового эволюционного процесса, называемого информатизацией, в развитии человеческого общества.

Универсальным техническим средством обработки любой информации является компьютер и разработанные для конкретных целей программные продукты. К таким продуктам обработки информации относятся различные информационные системы, позволяющие систематизировать и обрабатывать огромные потоки информации. Для персональных компьютеров было разработано множество различных программ, предназначенных для решения различных потребностей человека. Данная программа является математической системой, позволяющей любому человеку, быстро и практично провести любые расчёты.

1 Постановка задачи

Необходимо разработать программу «Калькулятор», которая должна отвечать следующим требованиям:

1. Многомодульность. Программа должна быть поделена на логические модули. Это упростит дальнейшее совершенствование программы, а также позволит локализовать ошибки.

2. Использование сложных типов данных: структур, списков, файлов. Для разработки данной программы необходимо подробно изучить принципы работы с структурами, списками и файлами для создания калькулятора.

3. Режим работы видеосистемы: текстовый/графический. Необходимо определиться с типом управления, видом интерфейса.

4. Устройство ввода информации: клавиатура/мышь. Необходимо определиться с механикой взаимодействия пользователя с программой и её интерфейсом.

5. Пользовательский интерфейс должен быть построен на основе меню и панели инструментов.

6. Наличие заставки. Заставка-приветствие должна сообщать пользователю необходимую и достаточную информацию о программе. Заставка сообщает пользователю информацию о программе.

6. Операционная система: MS Windows.

7. Язык программирования: Си и Ассемблер.

8. Среда разработки ПО - MS Visual Studio.

2 Выбор решения

Приложение «Калькулятор» состоит из трёх модулей:

1. kurs.cpp
2. kur.cpp
3. kur.h

Заголовочный файл «kur.h» — библиотека функций.

Файл «kur.cpp» содержит определение функций, объявленных в «kur.h.»

Файл «kurs.cpp» — главный файл. В нём объявляются основные переменные и массивы программы. Функция «main» — главная функция, в которой вызываются другие функции. Структура данных программы показана на рисунке 1.

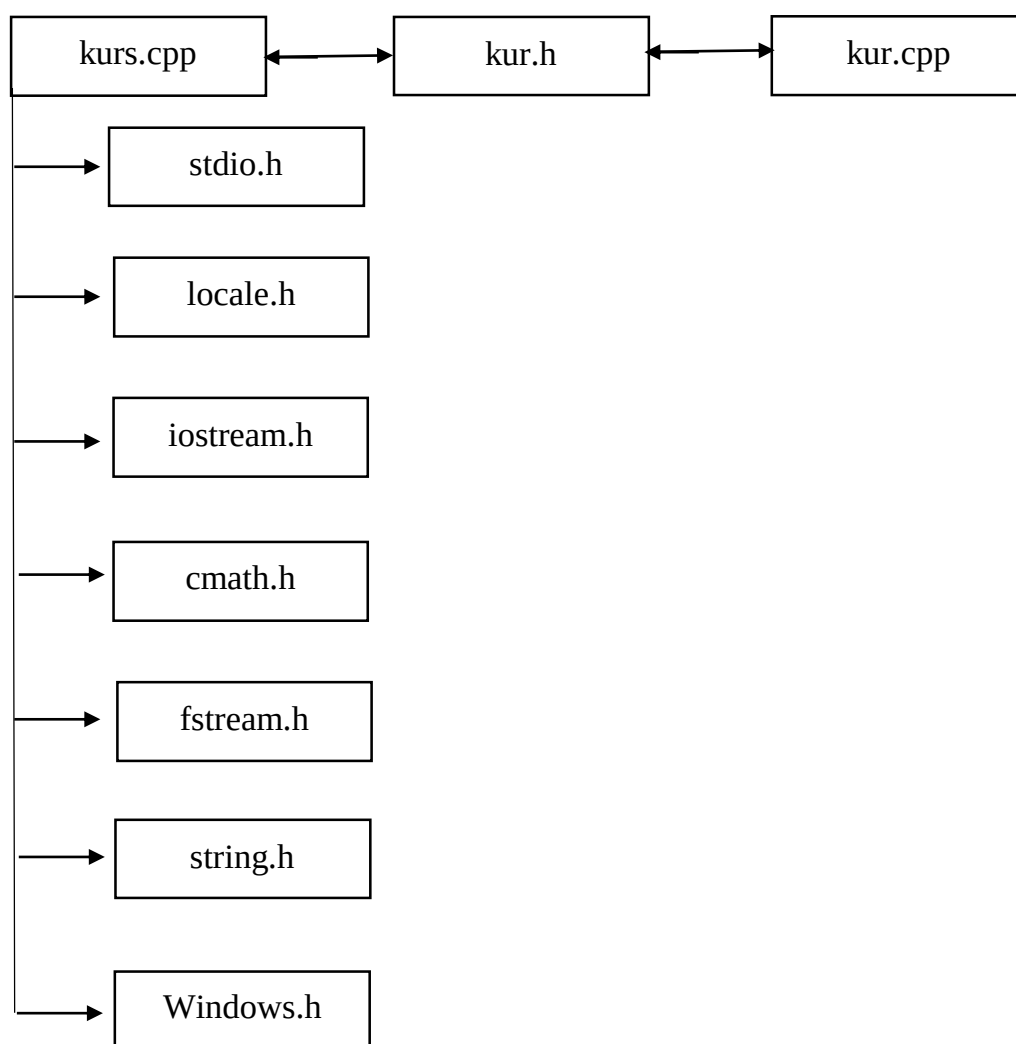


Рисунок 1. Схема данных программы

3 Описание разработки программы

Для написания данной программы будет использован язык программирования Си. Этот язык является распространённым языком программирования. При разработке языка Си был принят компромисс между низким уровнем языка ассемблера и высоким уровнем других языков. В языке Си предусмотрено много операций, непосредственно исполняемых большинством микропроцессоров; в то же время он дает программисту возможность выражать свои алгоритмы и данные наиболее подходящими средствами. Язык Си обеспечивает возможности структурирования данных. Он позволяет разрабатывать большие, но структурированные программы, предоставляя возможность отдельной разработки подпрограмм.

Целью создателей языка Си была разработка мобильного языка, который можно было бы использовать для разработки системного программного обеспечения. Программы, написанные на языке высокого уровня, мобильны, поскольку язык должен быть одним и тем же независимо от того, на каком компьютере и в какой операционной системе он используется. Дополнительное свойство языка Си, называемое условной компиляцией, позволяет программисту изолировать машинно-зависимые операторы и контролировать их компиляцию в другой среде. Это дополнительно повышает мобильность программного обеспечения, написанного на языке Си.

В качестве среды программирования был выбран программный продукт Visual Studio 2019.

Интерфейс информационной системы будет представлен в виде консоли, т.к. она является наиболее удобной и обеспечивает наименьшее время доступа к данным. Выбор по пунктов меню будет осуществляться с помощью клавиш управления.

При запуски приложения на экран будет выводиться информационная заставка.

Исходя из того, что при первом знакомстве с программой пользователю нужно дать краткую информацию о работе с ней, поэтому пользователю будет дан выбор функций, а также в программе будет предоставлен образец уравнений.

Так же после каждого выполненного расчёта уравнения программа будет автоматически сохранять историю вычисления в файл «history».

Для удобного использования калькулятора после каждого вычисления пользователю будет дан выбор: продолжить работу с калькулятором, вывести историю вычислений или выйти из программы.

При неправильном вводе данных или некорректном выборе функций, программа будет выводить «Ошибка».

4 Отладка и тестирование

В качестве среды разработки была выбрана программа Visual Studio 2019. Программа предоставляет все средства необходимые при разработке и отладке разрабатываемых модулей и программ. Для отладки использовались такие инструменты как точка останова, выполнение кода по шагам, анализ содержимого локальных и глобальных переменных, анализ содержимого памяти.

Тестирование проводилось в рабочем порядке, в процессе разработки, после завершения написания программы. В ходе тестирования было выявлено и исправлено множество проблем связанных с перерисовкой заставки, с перерисовкой структуры, с освобождением выделенной памяти и многое другое.

5 Описание программы

Разработка функции на языке Assembler

Как известно, функции, реализованные на языке ассемблера, работают быстрее и занимают меньше места. В качестве разрабатываемой функции были выбраны функции сложения и вычитания.

Описание функции выполнено на языке C, а тело функции представляет собой ассемблерную вставку, компилятор, входящий в состав среды разработки VS2019, позволяет это делать. Для этого перед ассемблерной вставкой необходимо написать ключевое слово `_asm`, а сам код поместить в фигурные скобки.

В своей работе я использовала ассемблер для удобного вычисления функций сложения и вычитания:

```
//c = a + b;
_asm
{
    mov eax, a_copy
    mov ebx, b_copy
    add eax, ebx
    mov c_copy, eax
}
//c = a - b;
_asm
{
    mov eax, a_copy
    mov ebx, b_copy
    sub eax, ebx
    mov c_copy, eax
}
```

Для реализации функций сложения и вычитания были использованы следующие ассемблерные инструкции:

- инструкция `mov` выполняет копирования содержимого источника в приемник;
- инструкция `add` выполняет суммирование содержимого источника и приемника;
- инструкция `sub` выполняет вычитание второго значения из первого;

6 Руководство пользователя

Программа «Калькулятор» предназначена для удобного расчёта различных функций. Программа имеет понятный интерфейс и поддерживает такие операции как вычисление арифметических действий, возведение в степень, извлечение корня, вычисление тригонометрических функций, сохранение истории вычисления в файл, а также возможность вывода истории на экран.



Рисунок 2.1 — Информационная заставка

Выбираем один из трёх действий. Если хотим использовать калькулятор вводим цифру 1 и нажимаем Enter.

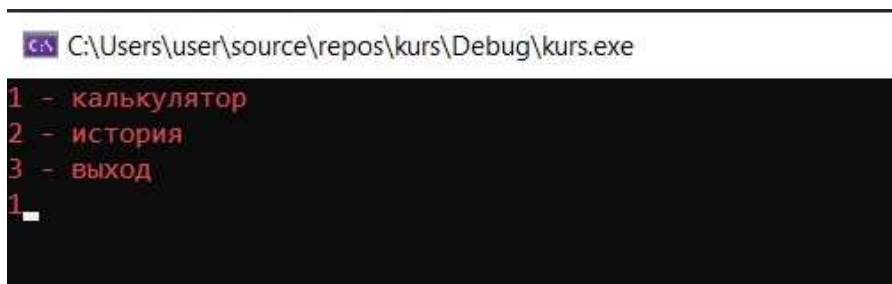


Рисунок 2.2 — Выбор действия

Далее выбираем функцию, которую хотим использовать. Например, возведение в степень. Для этого надо ввести цифру 5. При некорректном вводе данных будет выводиться «Ошибка».

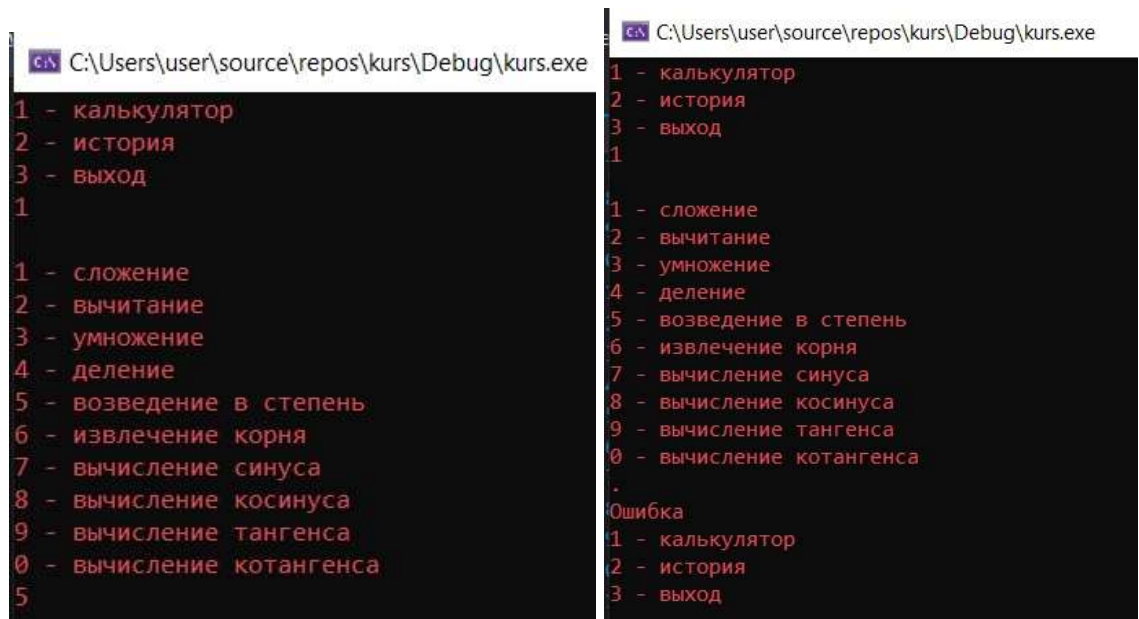


Рисунок 2.3 — Выбор функции

Потом надо ввести две переменных a и b. Например, a = 3, b = 4.

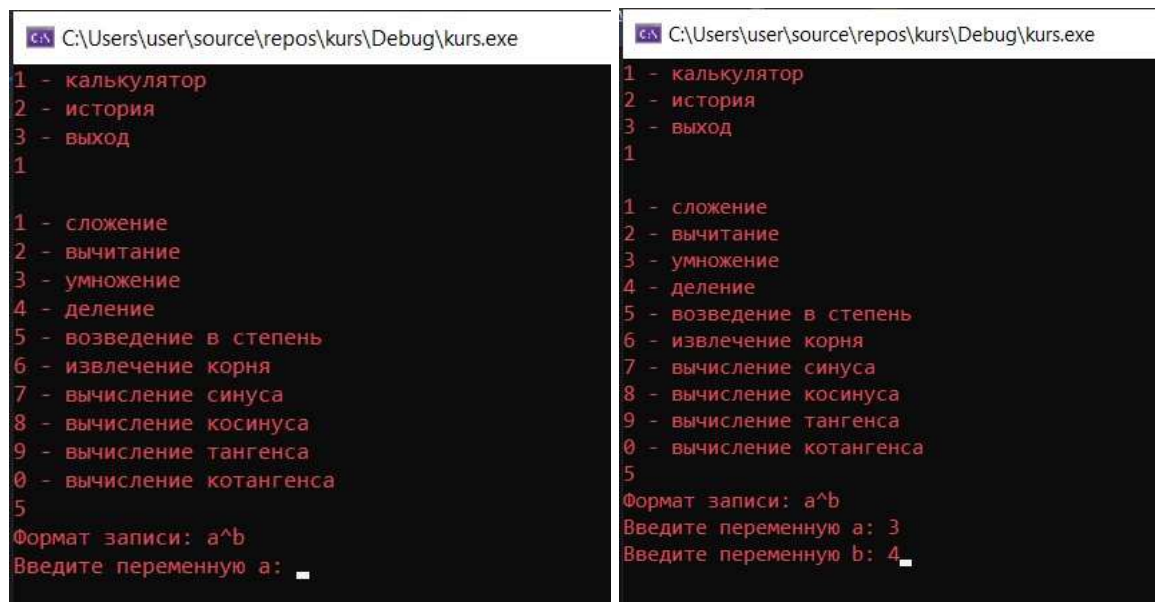
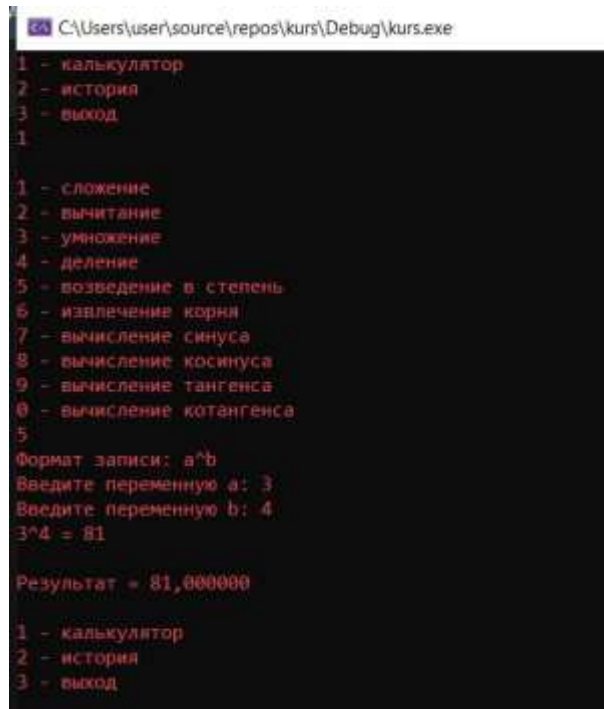


Рисунок 2.4 — Ввод переменных

На экран выводится результат. Далее можно снова использовать калькулятор, вызвать историю вычислений или закрыть программу.



```
C:\Users\user\source\repos\kurs\Debug\kurs.exe
1 - калькулятор
2 - история
3 - выход
1

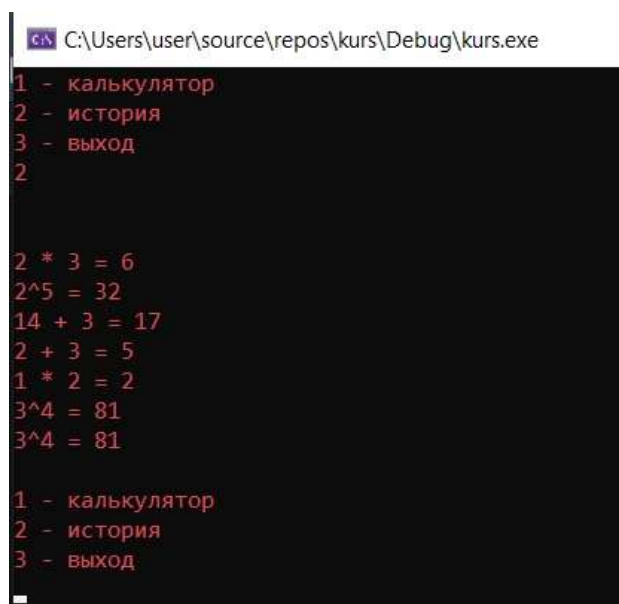
1 - сложение
2 - вычитание
3 - умножение
4 - деление
5 - возведение в степень
6 - извлечение корня
7 - вычисление синуса
8 - вычисление косинуса
9 - вычисление тангенса
0 - вычисление котангенса
5
Формат записи: a^b
Введите переменную a: 3
Введите переменную b: 4
3^4 = 81

Результат = 81,000000

1 - калькулятор
2 - история
3 - выход
```

Рисунок 2.5 — Вывод результата

Если пользователь захочет вывести историю всех вычислений на экран, ему следует ввести цифру 2 при выборе действия.



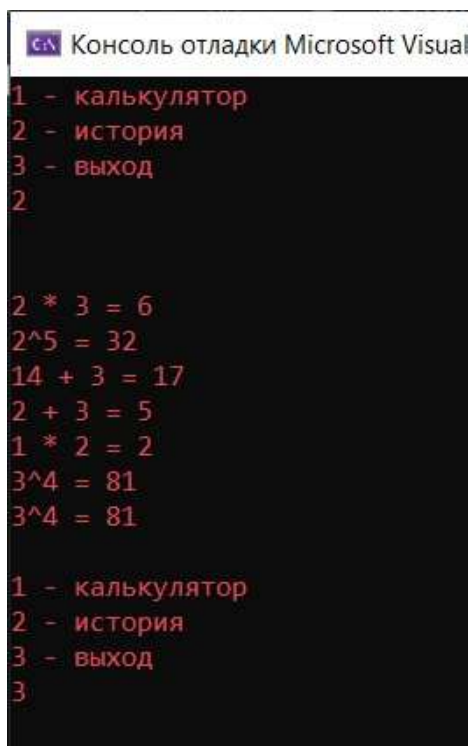
```
C:\Users\user\source\repos\kurs\Debug\kurs.exe
1 - калькулятор
2 - история
3 - выход
2

2 * 3 = 6
2^5 = 32
14 + 3 = 17
2 + 3 = 5
1 * 2 = 2
3^4 = 81
3^4 = 81

1 - калькулятор
2 - история
3 - выход
```

Рисунок 2.6 — Просмотр истории вычислений

Для выхода из программы необходимо выбрать пункт «Выход», для этого необходимо выбрать цифру 3 при выборе действия.



```
cs Консоль отладки Microsoft Visual  
1 - калькулятор  
2 - история  
3 - выход  
2  
  
2 * 3 = 6  
2^5 = 32  
14 + 3 = 17  
2 + 3 = 5  
1 * 2 = 2  
3^4 = 81  
3^4 = 81  
  
1 - калькулятор  
2 - история  
3 - выход  
3
```

Рисунок 2.5 — Выход из программы

Если выйти из программы, то все вычисления будут сохранены в файл и не будут удалены при повторном запуске программы.

Заключение

При выполнении данной курсовой работы были получены навыки разработки многомодульных программ. Изучены способы обработки событий с клавиатуры. Получены знания и навыки по работе со структурами, списками и файлами. Получены базовые навыки программирования на языке Си и Ассемблера, а также некоторых функций из Си++. Получены навыки отладки и тестирования программ. В рамках данной курсовой была написана программа «Калькулятор». Данная программа не требует большого количества системных ресурсов, и имеет управление с помощью клавиатуры. В дальнейшем, есть идеи по усовершенствованию программы: графический интерфейс, добавить логарифмические функции, сложные функции, производные и первообразные, пределы и матрицы, различные константы.

Список используемых источников

1. Керниган Б. Ритчи Д. Язык программирования С. 1985 г. -
https://www.r-5.org/files/books/computers/languages/c/kr/Brian_Kernighan_Dennis_Ritchie-The_C_Programming_Language-RU.pdf
2. Рудольф Марек Ассемблер на примерах 2005 г. -
https://sharpened.ucoz.ru/lib/assembler_na_primerah.pdf
3. К. Джамса. Учимся программировать на языке C++. 1997 г. -
<https://www.for-stydents.ru/informatika/cc/uchebniki/uchimsya-programmirovat-na-yazyke-s.html>
4. В. Г. Давыдов. Программирование и основы алгоритмизации. 2003 г.-
https://replace.org.ua/files/books/c/Programmirovanie_i_osnovyi_algo.pdf

Приложение А

Листинги программы

Файл «kurs.cpp»

```
#define _CRT_SECURE_NO_WARNINGS
#define _USE_MATH_DEFINES
#define PI 3.14159265
#include <stdio.h>
#include <locale.h>
#include <iostream>
#include <cmath>
#include <fstream> // для работы с файлами
#include <string>
#include <Windows.h>
#include "kur.h"

using namespace std;

int main()
{
    setlocale(LC_ALL, "Rus");
    vvod(); //заставка

    double a = 0, b = 0, c = 0;
    char d;
    int p = 1, a_copy = 0, b_copy = 0, c_copy = 0;

    //ofstream file("history.txt"); // поток для перезаписи
    (удаление содержимого)
    //file.close(); // закрытие потока на перезапись

    while (p != 3) {

        printf("1 - калькулятор\n2 - история\n3 - выход\n");
        scanf("%d", &p);
        printf("\n");

        if (p == 1) {

            printf("1 - сложение\n2 - вычитание\n3 -
умножение\n4 - деление\n5 - возведение в степень\n6 - извлечение
корня\n7 - вычисление синуса\n8 - вычисление косинуса\n9 -
вычисление тангенса\n0 - вычисление котангенса\n");
            scanf(" %c", &d);

            //while (getchar() != '\n');
            double sin_x;
            double cos_x;
            double tg_x;
            double ctg_x;
```

```

        ofstream file("history.txt", ios::app); // поток
для записи (дописывает в конец файла)
        switch (d)
        {
        case '1':
            printf("Введите первую переменную: ");
            scanf("%lf", &a);
            printf("Введите вторую переменную: ");
            scanf("%lf", &b);
            a_copy = a;
            b_copy = b;
            c_copy = c;
            //c = a + b;
            _asm
            {
                mov eax, a_copy
                mov ebx, b_copy
                add eax, ebx
                mov c_copy, eax
            }
            c = c_copy;
            cout << a << " + " << b << " = " << c << endl
<< endl; // Вывод на экран

            if (file.is_open()) {
                file << a << " + " << b << " = " << c <<
endl; // Запись в файл
            }
            break;

        case '2':
            printf("Введите первую переменную: ");
            scanf("%lf", &a);
            printf("Введите вторую переменную: ");
            scanf("%lf", &b);
            a_copy = a;
            b_copy = b;
            c_copy = c;
            //c = a - b;
            _asm
            {
                mov eax, a_copy
                mov ebx, b_copy
                sub eax, ebx
                mov c_copy, eax
            }
            c = c_copy;
            cout << a << " - " << b << " = " << c << endl
<< endl; // Вывод на экран

            if (file.is_open()) {

```

```

        file << a << " - " << b << " = " << c <<
endl; // Запись в файл
    }
    break;

    case '3':
        printf("Введите первую переменную: ");
        scanf("%lf", &a);
        printf("Введите вторую переменную: ");
        scanf("%lf", &b);

        c = a * b;

        cout << a << " * " << b << " = " << c << endl
<< endl; // Вывод на экран

        if (file.is_open()) {
            file << a << " * " << b << " = " << c <<
endl; // Запись в файл
        }
        break;

    case '4':
        printf("Введите первую переменную: ");
        scanf("%lf", &a);
        printf("Введите вторую переменную: ");
        scanf("%lf", &b);

        c = a / b;

        cout << a << " : " << b << " = " << c << endl
<< endl; // Вывод на экран

        if (file.is_open()) {
            file << a << " : " << b << " = " << c <<
endl; // Запись в файл
        }
        break;

    case '5':
        printf("Формат записи: a^b\n");
        printf("Введите переменную a: ");
        scanf("%lf", &a);
        printf("Введите переменную b: ");
        scanf("%lf", &b);

        c = pow(a, b);

        cout << a << "^" << b << " = " << c << endl
<< endl; // Вывод на экран

```

```

        if (file.is_open()) {
            file << a << "^" << b << " = " << c <<
endl; // Запись в файл
        }
        break;

    case '6':
        printf("Формат записи: a^(1/b)\n");
        printf("Введите переменную a: ");
        scanf("%lf", &a);
        printf("Введите переменную b: ");
        scanf("%lf", &b);

        c = pow(a, 1 / b);

        cout << a << "^" << "(1/" << b << ")" << " = "
<< c << endl << endl; // Вывод на экран

        if (file.is_open()) {
            file << a << "^" << "(1/" << b << ")" <<
" = " << c << endl; // Запись в файл
        }
        break;

    case '7':
        printf("Формат записи: sin(b)\n");
        printf("Введите переменную b: ");
        scanf("%lf", &b);
        sin_x = sin(b);
        c = sin_x;

        cout << "sin(" << b << ")" << " = " << c <<
endl << endl; // Вывод на экран

        if (file.is_open()) {
            file << "sin(" << b << ")" << " = " << c
<< endl; // Запись в файл
        }
        break;

    case '8':
        printf("Формат записи: cos(b)\n");
        printf("Введите переменную b: ");
        scanf("%lf", &b);
        cos_x = cos(b);

        c = cos_x;

```

```

        cout << "cos(" << b << ")" << " = " << c <<
endl << endl; // Вывод на экран

        if (file.is_open()) {
            file << "cos(" << b << ")" << " = " << c
<< endl; // Запись в файл
        }
        break;

    case '9':
        printf("Формат записи: tg(b)\n");
        printf("Введите переменную b: ");
        scanf("%lf", &b);
        tg_x = tan(b);

        c = tg_x;

        cout << "tg(" << b << ")" << " = " << c <<
endl << endl; // Вывод на экран

        if (file.is_open()) {
            file << "tg(" << b << ")" << " = " << c
<< endl; // Запись в файл
        }
        break;

    case '0':
        printf("Формат записи: ctg(b)\n");
        printf("Введите переменную b: ");
        scanf("%lf", &b);
        ctg_x = 1 / tan(b);

        c = ctg_x;

        cout << "ctg(" << b << ")" << " = " << c <<
endl << endl; // Вывод на экран

        if (file.is_open()) {
            file << "ctg(" << b << ")" << " = " << c
<< endl; // Запись в файл
        }
        break;

    default:
        printf("Ошибка\n");
        continue;
    }
    printf("Результат = %lf\n\n", c);
    file.close(); // закрытие потока на запись
}

```

```

        else
            if (p == 2) {
                string line;
                ifstream file("history.txt"); // открываем
                файл для чтения из него
                if (file.is_open()) {
                    while (getline(file, line))
                    {
                        cout << line << endl;
                    }
                    cout << endl;
                }
                file.close(); // закрытие потока на чтение
            }

            else
                if (p == 3) break;

        }
    }

```

файл «kur.cpp»

```

#include <iostream>
#include <Windows.h>
#include "kur.h"
void vvod()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    system("color 0C");
    printf("\n\n\n\n");
    printf("\n
||||||||||||||||||||||||||||| ");
    printf("\n
----- || ");
    printf("\n
| || ");
    printf("\n
----- || ");
    printf("\n
///    /// /// || ");
    printf("\n
|| ");
    printf("\n
/x/ || ");
    printf("\n
|| ");
    printf("\n
|sin| |cos| |tg| |ctg| || ");

```

```

        printf("\n
|| ");
        printf("\n
|| ");
|7| |8| |9| |/| || ");
        printf("\n
|| ");
        printf("\n
|| ");
|4| |5| |6| |*| || ");
        printf("\n
|| ");
        printf("\n
|| ");
|1| |2| |3| |+| || ");
        printf("\n
|| ");
        printf("\n
|| ");
|0| |.| |-| |=| || ");
        printf("\n
|| ");
        printf("\n
||||||||||||||||||||||||||||| ");
        Sleep(2500);
        system("cls");
}

```

файл «kur.h»

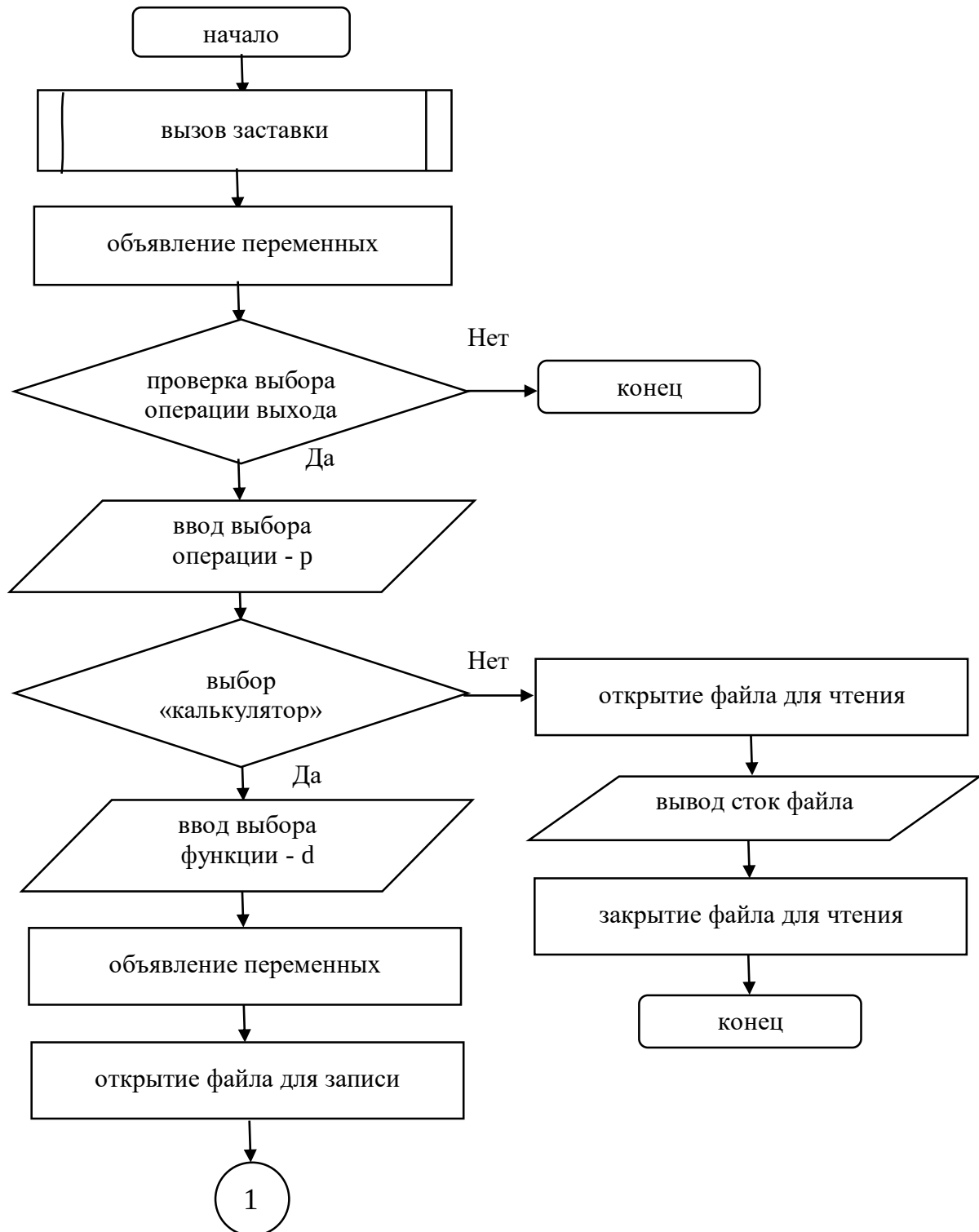
```

#pragma once
void vvod();

```

Приложение В

Схемы программы



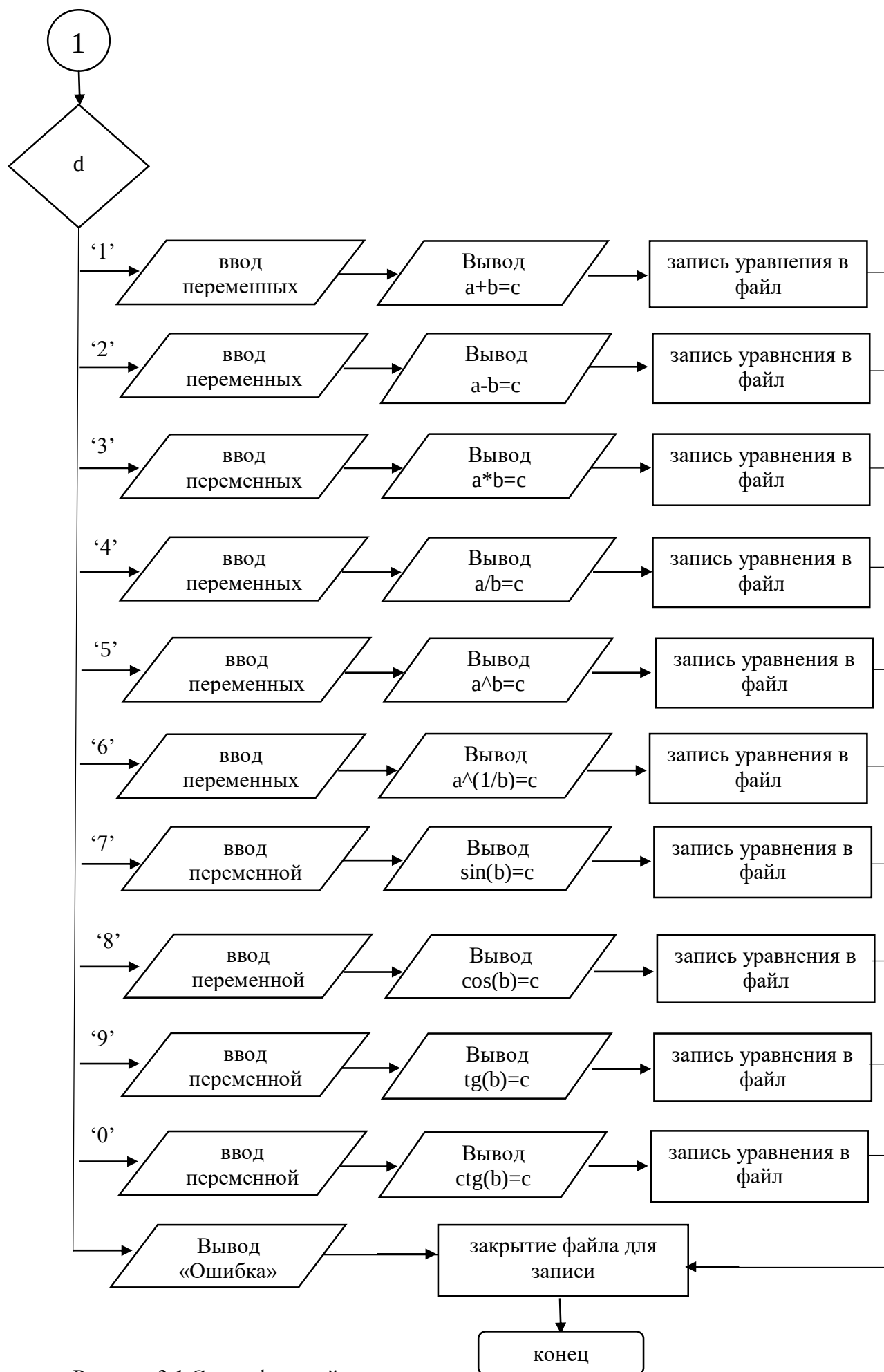


Рисунок 3.1 Схема функций программы

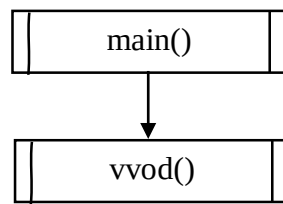


Рисунок 3.2 Иерархическая структура программы

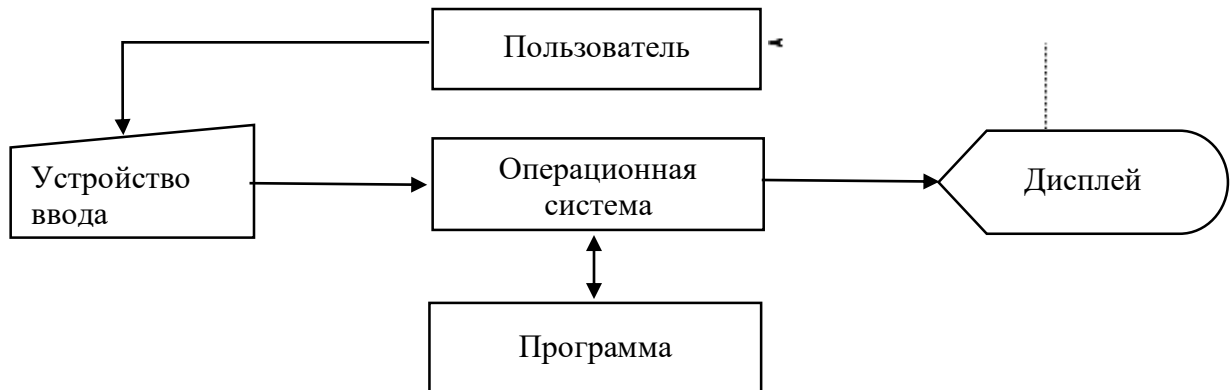


Рисунок 3.3 Схема взаимодействия программ