|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **Компьютерные системы и сети (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.03 Прикладная информатика**

**ОТЧЕТ ОБ УЧЕБНОЙ ПРАКТИКЕ №2**

|  |  |
| --- | --- |
| Тип практики | Проектно-технологическая практика |

|  |  |
| --- | --- |
| Название  предприятия | НУК ИУ МГТУ им. Н.Э. Баумана |

|  |  |  |  |
| --- | --- | --- | --- |
| Студент группы ИУ6-25 Б |  |  | Дулина И.А. |
|  |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |
| Руководитель практики |  |  |  |
|  |  | (Подпись, дата) | (И.О. Фамилия) |

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*2023 г.*

**Оглавление**

Техническое задание3

Код программы3

Пример работающей версии программы8

Диаграмма состояний интерфейса12

Заключение12

**Создание программной системы с элементарным интерфейсом консольного режима на С++**

**Условие:** Выполнить структурную декомпозицию, разработать структурную схему, содержащую не менее 3 подпрограмм, и алгоритмы этих подпрограмм. Реализовать на С++ в консольном режиме. Предусмотреть примитивный интерфейс типа меню, позволяющий выбирать нужную подпрограмму.

Разработать программу, которая выполняет простейшее исследование функций одной переменной. Реализовать следующие операции: ввод функции, преобразование ее во внутреннее представление (дерево), вычисление значения функции от заданного аргумента, а также вывод результатов на экран.

**Цель:** получение навыков создания программной системы на языке Pascal с помощью среды Lazarus

*Код основной программы:*

#include <iostream>

#include <math.h>

#include <cmath>

using namespace std;

const int N = 256;

struct core {

char oper[5]; //знак

int value; //константа

core\* left;

core\* right;

};

//прототипы функций

void print();

void tree(core\* r, char str[]);

int position(char st[], char ops[]);

float solve(core\* r, bool&key);

char function[N];

core\* root;

bool key = true;

int x;

**int main ()**

{

int n; float y;

cout << "Choose the function:\n";

print();

cin >> n;

switch (n) {

case 1: strcpy\_s(function, "2\*x"); break;

case 2: strcpy\_s(function, "4\*sin(-x)+(x^2-3)"); break;

case 3: strcpy\_s(function, "cos(7\*x+2)"); break;

case 4: strcpy\_s(function, "(4-x)\*(1+x)"); break;

case 5: strcpy\_s(function, "1/x"); break;

}

root = new core;

tree(root, function);

cout << "\nInput the value of x:";

cin >> x;

y = solve(root, key);

if (key != false)cout<<y;

else cout << "Impossible";

return 0;

}

**void print()** {

cout << "1)y=2\*x\n2)y=4\*sin(-x)+(x^2-3)\n3)y=cos(7\*x+2)\n4)y=(4-x)\*(1+x)\n5)y=1/x\n";

}

//есть ли "главная" операция

**int position(char st[], char ops[]) {**

int i = 0, j = 0, k = 0; int p;

p = 0;

while (i < strlen(st) && p == 0) {

if (st[i] == '(') j++;

else if (st[i] == ')') k++;

else if (j == k && strchr(ops, st[i]) != nullptr) p = i;

i++;

}

return p;

}

**void tree(core\* r, char str[]) {**

char op[6], o[2] = { ' ', '\0' }, o2[3]{ ' ', ' ', '\0' }, strr[N], strl[N],str1[N], \* ptr1;

int nmain, nnn;

op[0] = '+'; op[1] = '-'; op[2] = '\0';

nmain = position(str, op);

op[0] = '\*'; op[1] = '/'; op[2] = '\0';

if (nmain == 0) nmain = position(str, op);

op[0] = '^'; op[1] = '\0';

if (nmain == 0) nmain = position(str, op);

op[0] = '\*'; op[1] = '/'; op[2] = '+';

op[3] = '-'; op[4] = '^'; op[5] = '\0';

if (nmain != 0) {

o[0] = str[nmain];

strcpy\_s(r->oper, o);

strncpy\_s(strl, str, nmain);

strl[nmain] = '\0';

if (strl[0] == '(' && position(strl, op) == NULL) {

strcpy\_s(strl, &strl[0]+1);

strl[strlen(strl)-1] = '\0';

}

ptr1 = &str[nmain] + 1;

strcpy\_s(strr, ptr1);

if (strr[0] == '(' && position(strr, op) == NULL) {

strcpy\_s(strr, &strr[0] + 1);

strr[strlen(strr)-1] = '\0';

}

r->left = new core;

tree(r->left, strl);

r->right = new core;

tree(r->right, strr);

}

else

if (str[0] == 'x'|| strcmp(str, "-x")==0 ){ //переменная

if (str[0] == 'x') {

o[0] = 'x';

strcpy\_s(r->oper, o);

r->left = nullptr;;

r->right = nullptr;

}

else {

o[0] = '-';

strcpy\_s(r->oper, o);

strcpy\_s(strl, "0");

strcpy\_s(strr, "x");

r->left = new core;

tree(r->left, strl);

r->right = new core;

tree(r->right, strr);

}

}

else

if (atoi(str)!=0 || strcmp(str, "0") == 0) { //константа

o[0] = 'n';

strcpy\_s(r->oper, o);

r->left = nullptr;

r->right = nullptr;

r->value = atoi(str);

}

else { //функция

ptr1 = strchr(str, '(');

nnn = strlen(str) - strlen(ptr1);

strncpy\_s(str1, str, nnn);

str1[nnn] = '\0';

strcpy\_s(r->oper, str1);

r->right = nullptr;

strcpy\_s(strl, ptr1 + 1);

strl[strlen(strl) - 1] = '\0';

r->left = new core;

tree(r->left, strl);

}

}

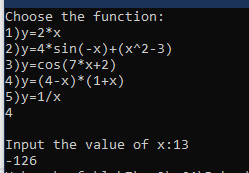


Рисунок 1 – работающая версия программы

Блок-схема алгоритма:

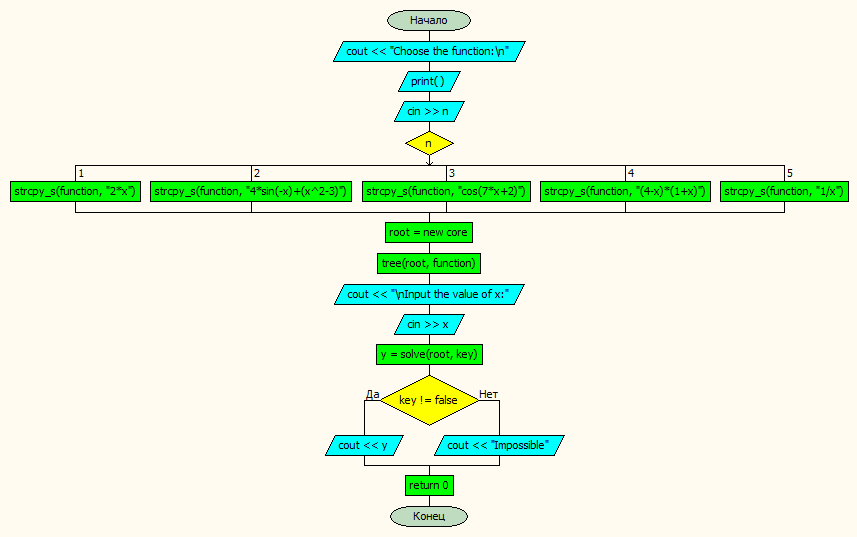
****

Рисунок 2 – основная программа

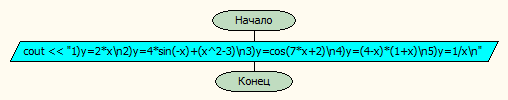
****

Рисунок 3 – печать меню функций

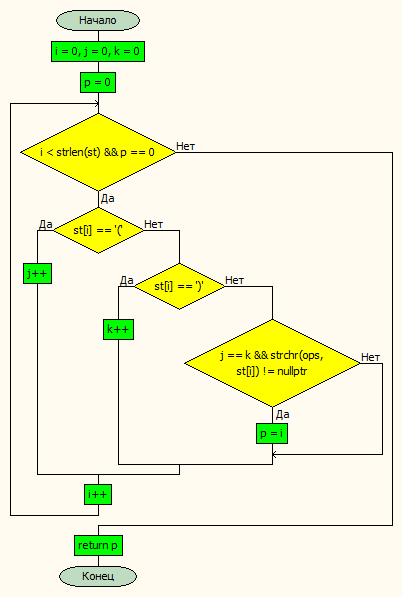
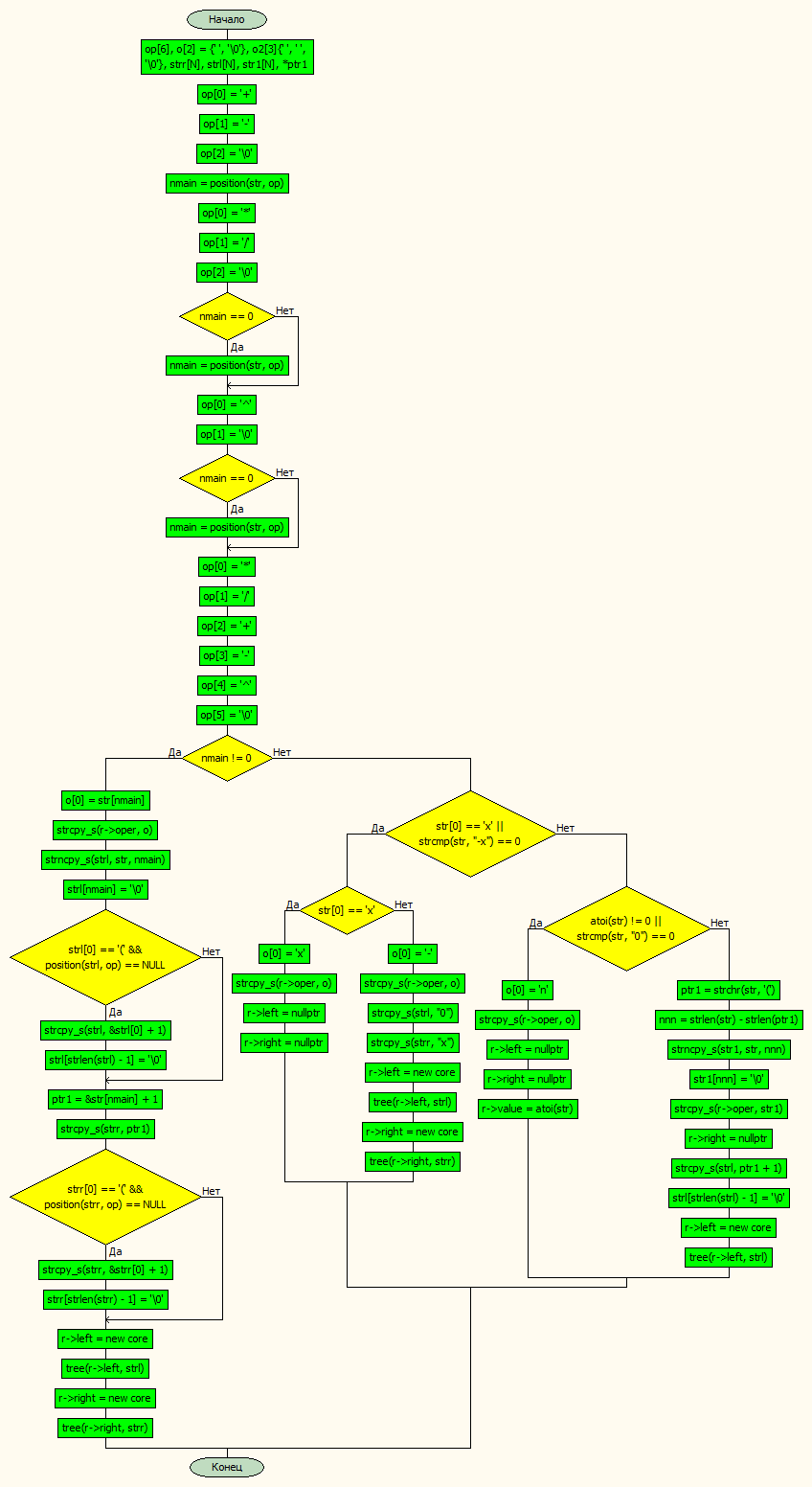
****

Рисунок 4 – определение последней операции

****

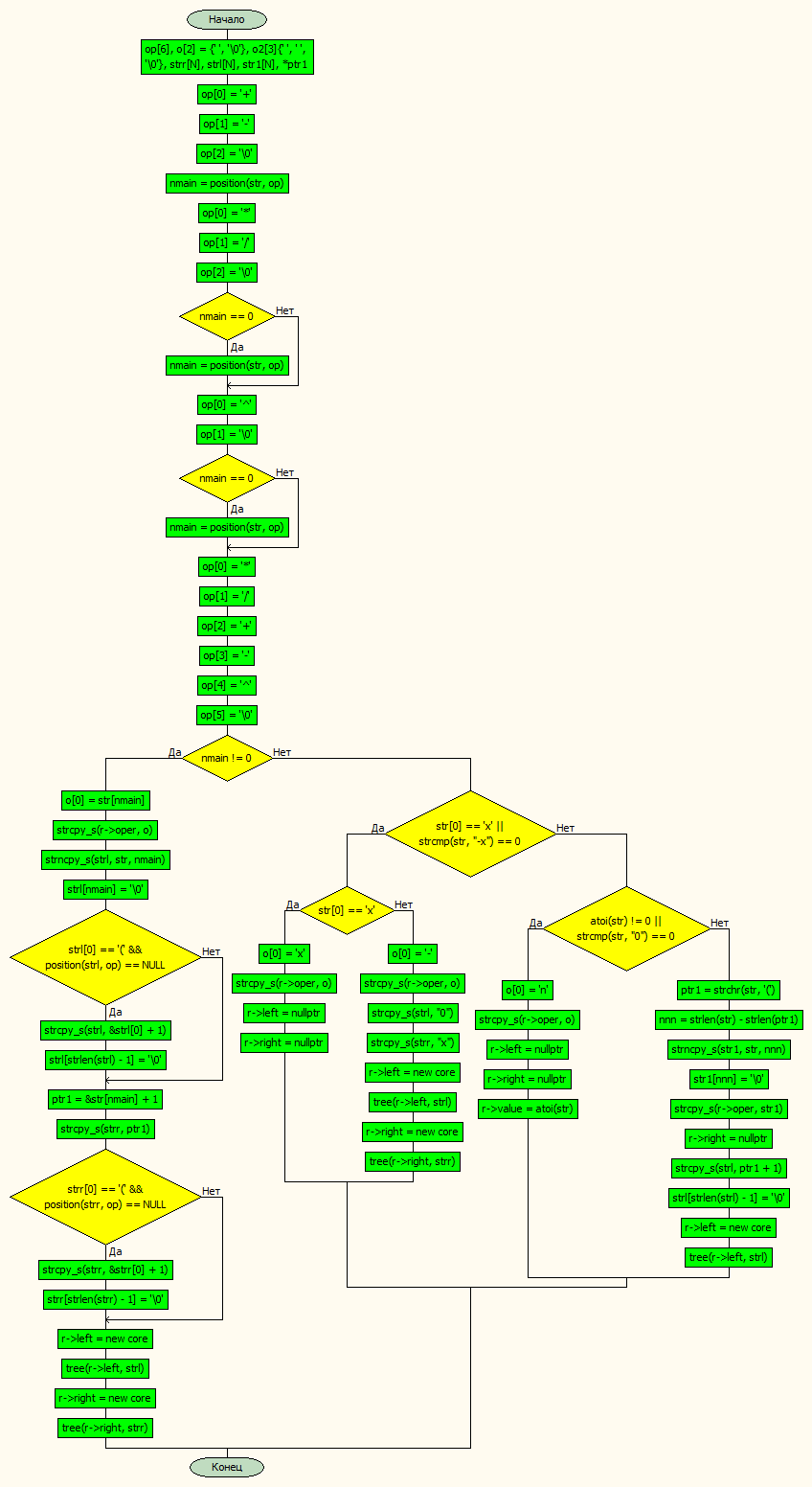
****

Рисунок 5 – составление дерева

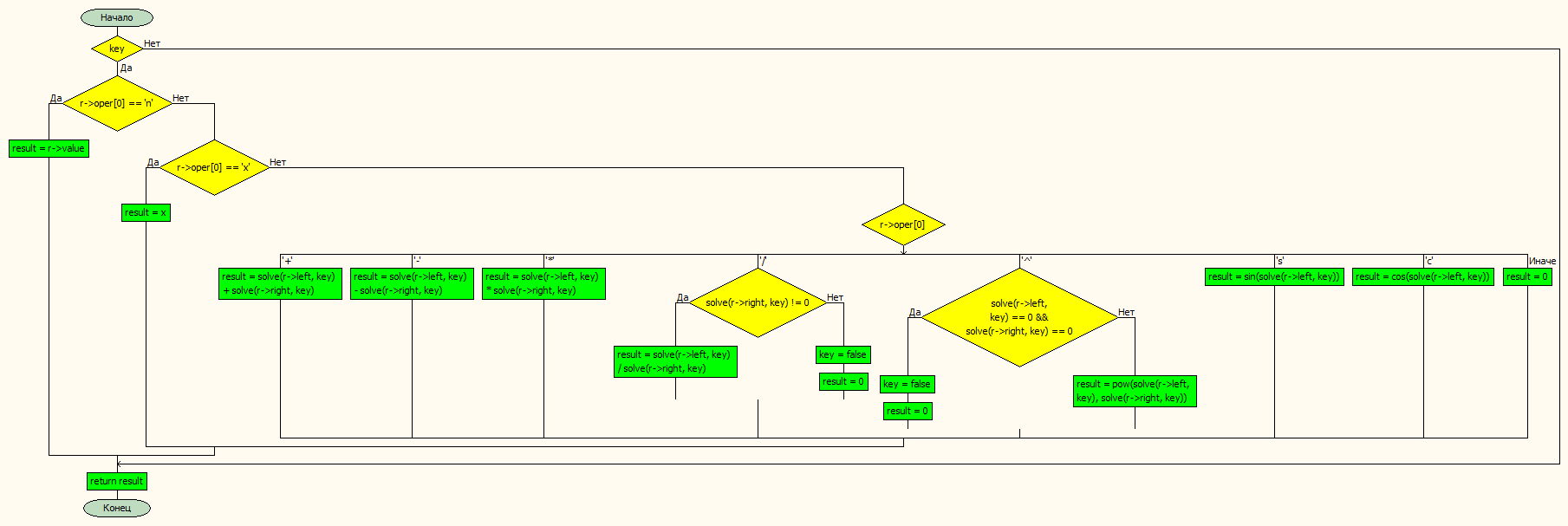
****

Рисунок 6 – вычисление значения функции



Рисунок 7 – диаграмма состояний интерфейса

**Вывод:** в соответствии с условием задачи была создана программа, которая предлагает пользователю на выбор функции и определяет её значение в зависимости от введённого аргумента.