**ФЕДЕРАЛЬНОЕ АГЕНСТВО ВОЗДУШНОГО ТРАНСПОРТА**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ГРАЖДАНСКОЙ АВИАЦИИ (МГТУ ГА)

Кафедра вычислительных машин, комплексов, сетей и систем.

Курсовая работа

защищена с оценкой

(подпись преподавателя, дата)

КУРСОВАЯ РАБОТА

по дисциплине ”Программирование на языке высокого уровня”

Вариант № 28

Тема: Разработать систему рисования кривых

Выполнил студент группы ЭВМ 2-2

Новичков Виталий Дмитриевич \_

(Ф.И.О.)

Руководитель:

Доц, к.ф.-м.н. Надейкина Людмила Анатольевна

(звание, степень, Ф.И.О.)

МОСКВА - 2012

<Вторая страница должна быть пустой (вместо неё будет вставлен листочек, который дала Надейкина)>

2. Возможности разработанной системы

Данная система предназначена для рисования графиков функций из введённой в строку формулы. График рисуется на экране цветную декартову систему координат XY, поверх неё рисует график функции, формула которой была введена строкой предварительно перед его прорисовкой.

Пользователь может указать размер графика (При этом, график автоматически масштабируется во весь экран), изменить цвета отображения графика функции. Также пользователю прямо из интерфейса программы доступна краткая помощь по вводу формулы для графика. Формулы рассчитываются как y=f(x).

3. Алгоритмы функций обработки данных

3.1 ГЛАВНАЯ ФУНКЦИЯ ПРОГРАММЫ  
Рисунок 3.1 Главная функция программы.



Рисунок 3.1 Продолжение





Рисунок 3.1 Продолжение



Рисунок 3.1 Продолжение



Рисунок 3.1 Продолжение



ФУНКЦИЯ GRAPH  
Рисунок 3.2 Функция graph  


Рисунок 3.2 Продолжение



Рисунок 3.3 Функция goto25



Рисунок 3.4 Функция goto27



Рисунок 3.5 Функция ems



Рисунок 3.6 Функция prc



Рисунок 3.7 Функция factor



Рисунок 3.8 Функция squear



Рисунок 3.9. Функция test



Рисунок 3.10 Функция mod



Рисунок 3.11 Функция expr



Рисунок 3.12 Функция term



Рисунок 3.13 Функция skip



Рисунок 3.14 Функция numpar



4. Программа системы

4.1 ФАЙЛЫ ПРОГРАММЫ

**MATH1.CPP** – Главный файл программы

HEAD.H - Заголовочные файлы

GLOBALS.H - Список глобальных переменных

PROTOT.H - Прототипы функций

**MAIN.C** - Главная функция

FNLIB.H - Библиотека функций

EMS.C - Функция  
expr.c - Функция  
FACTOR.C - Функция  
GOTO2X.C - Функция  
graph.c - Функция  
MOD.C - Функция  
NEXT.C - Функция  
N

UMPAR.C - Функция  
NVDS.C - Функция  
PRC.C - Функция  
SKIP.C - Функция  
SQUEAR.C - Функция  
STCOLOR.C - Функция  
TERM.C - Функция  
test.c - Функция  
TXTBRDR.C - Функция

4.2 ГЛОБАЛЬНЫЕ ПЕРЕМЕННЫЕ ПРОГРАММЫ

|  |  |  |  |
| --- | --- | --- | --- |
| Имя | Тип | Умолчание | Назначение |
| i | Int | 0 | Счётчик |
| jjj | Int | 0 | Счётчик |
| sizea | Int | -5 | Минимум масштаба |
| sizeb | Int | 5 | Максимум масштаба |
| key | Int | <нет> | Для меню |
| menuitem | Int | 0 | Для меню |
| gd | Int | DETECT | Драйвер VGA |
| gm | Int | <нет> | Драйвер VGA |
| err | Int | <нет> | Сообщения об ошибках VGA |
| color1 | Int | BLACK | Параметр цвета 1 |
| color2 | Int | DARKGRAY | Параметр цвета 2 |
| color3 | Int | GREEN | Параметр цвета 3 |
| color4 | Int | YELLOW | Параметр цвета 4 |
| color5 | Int | GREEN | Параметр цвета 5 |
| c | Char | <нет> | Буфер |
| jjk | Char\* | <нет> | Счётчик |
| s, | Char\* | <нет> | Буфер формул |
| tmp, | Char\* | <нет> | Буфер |
| sid[6], | Char | <нет> | Запоминание позиции распознавания формулы |
| sback, | Char\* | <нет> | Резервная копия буфера формул |
| sidback[6]; | Char | <нет> | Резервная копия памяти позиции |
| x\_dot | Float | <нет> | Истинность Попадания точки в пределы экрана |
| y\_dot | Float | <нет> | Истинность Попадания точки в пределы экрана |
| xpr | Float | <нет> | Истинность Попадания точки в пределы экрана |
| ypr | Float | <нет> | Истинность Попадания точки в пределы экрана |
| ycheck | Float | <нет> | Истинность Попадания точки в пределы экрана |
| ycheck2 | Float | <нет> | Истинность Попадания точки в пределы экрана |
| xcheck | Float | <нет> | Истинность Попадания точки в пределы экрана |
| xcheck2 | Float | <нет> | Истинность Попадания точки в пределы экрана |

4.3 СТРУКТУРА ПРОГРАММЫ



5. Руководство по использованию

**Главное меню:**

Запустив файл **math1.exe** Перед Вами появится меню:

1. Нарисовать график по формуле
2. Задать масштаб для графика
3. Выбрать цветовую палитру
4. Краткая помощь по вводу формул
5. Об авторе (С)
6. Выход

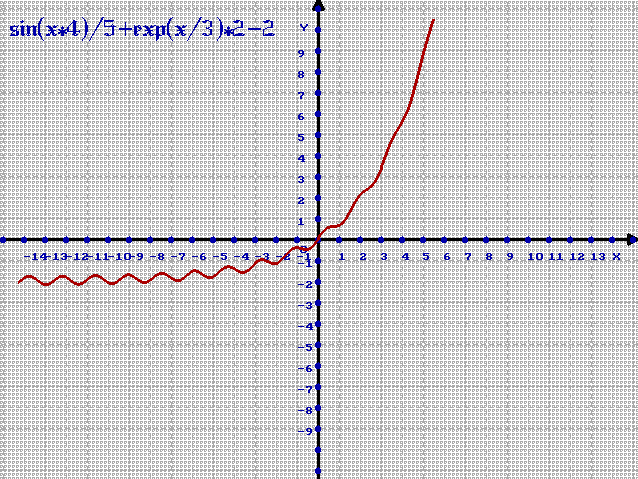
Выбор пункта меню производится нажатием клавиш-цифр 1-6 в соответствие с желаемым пунктом меню.

**Рисование графика по формуле:**

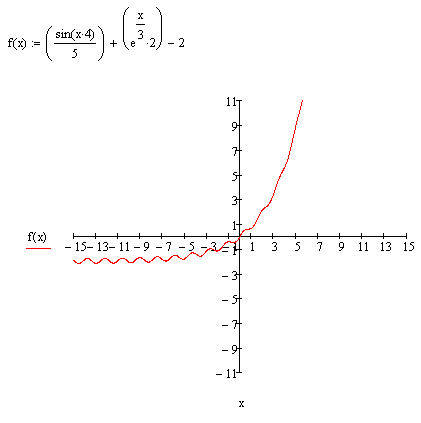
Выбрав первый пункт меню, Вы начинаете вводить формулу, по которой будет нарисован график. На запрос “f(x)=” Вы должны будете ввести по соответствующим правилам формулу или ввести нолик “0” для **возврата в главное меню**.

Если Вы введёте формулу, то будет прорисовываться график функции:  
  
**Пример:**

Формула f(x)= **sin(x\*4)/5+exp(x/3)\*2-2**



(Вывод формулы В math1.exe)



(сравнение с выводом той же формулы в Mathcad 14)

**Задание масштаба графика.**

Выбрав пункт меню 2, Вам будет выведен диалог, в котором Вы должны будете ввести числовое значение минимума и максимума по очереди.

**Выбрать цветовую палитру.**

Выбрав пункт меню 3, Вам будет высвечено пять элементов, чей цвет Вы можете назначить. Выбрав в соответствие с цифрой указанного элемента Вам будет выведено окно, в котором клавишами-стрелками или выбором соответствующего пункта клавише-цифре (0-9, A-F). Выбранный цвет отобразится в меню. Нажатие любой сторонней клавиши, Esc, влево отменяет Ваш выбор, цвет элемента останется без изменений. Нажатие клавиш Enter или вправо сохраняет выбранную опцию и выбранный цвет отобразится в списке элементов.

**Краткая помощь по формулам**

Вы можете в любой удобный момент просмотреть, а какие формулы в программе существуют и какие операторы разрешены в синтаксисе набора формул.

**Об Авторе.**

Вывод на экран Все сведения об авторе программы.

**Выход из программы.**

Для выхода из программы нажмите в главном меню клавишу «6».

**Синтаксис набора формул в приложение:**

**Условные обозначения:**

x - Переменная

Z - Любое допустимое выражение

**Используемые Функции**

sin(Z) - Синус выражения Z cos(Z) - Косинус выражения Z

tg(Z) - Тангенс выражения Z ctg(Z) - Котангенс выражения Z

exp(Z) - e^Z sqrt(Z) - Квадратный корень выражения Z

log(Z) - Логорифм Z от основания 10

**Используемые операторы**

+ - Сложение

- - Вычитание

\* - Умножение

/ - Деление

^ - Возведение в степень

() - Объединение выражения в одно значение

**Пример формулы:**  f(x)= sin((3/pi)\*x)/4+cos(3\*x)^3-exp(x/7)  
Равносильно формуле:

Приложение 1. Исходные коды программы.

**MATH1.CPP**

///////////////////////////////////////////////////////

// //

// Московский Государственный Технический //

// Университет Гражданской Авиации //

// //

// Курсовая работа студента группы: ЭВМ-2-2 //

// Новичкова Виталия Дмитриевича //

// //

// //

///////////////////////////////////////////////////////

//Главный файл

/////////Голова////////////////////////////////////////

#include "head.h"

/////////Переменные/и/прототипы////////////////////////

#include "globals.h" //Глобальные переменные

#include "protot.h" //Прототипы функций

/////////Библиотека/функций////////////////////////////

#include "fnlib.h" //Библиотека функций

/////////Главная функция///////////////////////////////

#include "main.c"

**HEAD.H**

//Заголовок

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <math.h>

#include <ctype.h>

#include <conio.h>

#include <alloc.h>

#include <graphics.h>

#include <DOS.h>

#include <iostream.h>

#define pi 3.14159265

**GLOBALS.H**

//Глобальные переменные

int

i=0,

jjj=0,

sizea=-5,

sizeb=5,

key,

menuitem=0,

gd=DETECT,

gm,

err,

color1 = BLACK,

color2 = DARKGRAY,

color3 = GREEN,

color4 = YELLOW,

color5 = GREEN;

char

c,

\* jjk,

\* s,

\* tmp,

sid[6],

\* sback,

sidback[6];

float

x\_dot,

y\_dot,

xpr,

ypr,

ycheck,

ycheck2,

xcheck,

xcheck2;

struct Operation

{

char\* sid[7];

int Op;

double tokval;

double x;

}Oper={"sin","cos","tg","ctg","sqrt","log","exp"};

PROTOT.H

//Прототипы функций

int mod(char \*s);

int next(char \*s,char \*sid);

int skip(int t, char \*s,char \*sid);

int test();

int graph(char \*s,char \*sid);

int stcolor(int);

double expr(char \*s,char \*sid);

double numpar(char \*s,char \*sid);

double factor(char \*s,char \*sid);

double term(char \*s,char \*sid);

void goto25(int);

void goto27(int);

void prc(char\*, int, int);

void sqwear(int, int, int, int, int);

void textborder(int, int, int, int, int);

void ems(char \*, int);

void nvd\_logo();

class graphic

{

public:

void on()

{

detectgraph(&gd,&gm);err=graphresult();

if(err){printf("\n%s",grapherrormsg(err));exit(1);}

initgraph(&gd,&gm,"graph");err=graphresult();

if(err){printf("n%s", grapherrormsg(err));exit(2);}

cleardevice();clrscr();

}

void off()

{

cleardevice();clrscr();closegraph();

}

};

graphic GDI;

**MAIN.C**

//Главная функция

void main()

{

nvd\_logo();

mainmenu:

clrscr();

printf("\n\n\n\n\n\n");

for(i=0;i<80;i++) printf("=");

printf(" ==Система рисования графиков==\n");

printf(" ====Главное меню====\n");

printf("\n");

printf(" 1. Нарисовать график по формуле\n");

printf(" 2. Задать масштаб для графика\n");

printf(" 3. Выбрать цветовую палитру\n");

printf(" 4. Краткая помощь по вводу формул\n");

printf(" 5. Об авторе (с)\n");

printf(" 6. Выход\n");

for(i=0;i<80;i++) printf("=");

printf(" Введите команду: ");

do

{

gotoxy(wherex()-1, wherey());printf(" ");gotoxy(wherex()-1, wherey());

menuitem=getche();

if (menuitem==0) menuitem=getche();

}

while((menuitem>'6')||(menuitem<'1'));

switch(menuitem)

{

case '1':

goto EnterFormula;

case '2':

goto ChSize;

case '3':

goto ChColor;

case '4':

goto info;

case '5':

goto aboutme;

case '6':

clrscr();

goto exit;

}

ChColor:

//sqr();

clrscr();

ColorMenu:

gotoxy(1,1);

printf("Выберите параметр:\n");

i=0;jjj=color1;if(jjj==0){jjj=0;i=7;}

prc("1. Фон\n",i,jjj);

gotoxy(1,wherey());

i=0;jjj=color2;if(jjj==0){jjj=0;i=7;}

prc("2. Сетка\n",i,jjj);

gotoxy(1,wherey());

i=0;jjj=color3;if(jjj==0){jjj=0;i=7;}

prc("3. Оси координат\n",i,jjj);

gotoxy(1,wherey());

i=0;jjj=color4;if(jjj==0){jjj=0;i=7;}

prc("4. График функции\n", i, jjj);

gotoxy(1,wherey());

i=0;jjj=color5;if(jjj==0){jjj=0;i=7;}

prc("5. Подписи\n", i, jjj);

gotoxy(1,wherey());

cprintf("\nВведите параметр (Esc - Возврат): ");

ChnMenu:

key=getch();

if(!((key>='1')&&(key<='5')||(key==27)))goto ChnMenu;

switch(key)

{

case '1':

color1 = stcolor(color1);

goto ColorMenu;

case '2':

color2 = stcolor(color2);

goto ColorMenu;

case '3':

color3 = stcolor(color3);

goto ColorMenu;

case '4':

color4 = stcolor(color4);

goto ColorMenu;

case '5':

color5 = stcolor(color5);

goto ColorMenu;

case 27:

;

}

goto mainmenu;

ChSize:

gotoxy(20, wherey()+2);

printf("Задайте масштабы для графика:\n");

gotoxy(20, wherey());

printf("Введите минимум (Текущий %i):", sizea);

cin >> sizea;

gotoxy(20, wherey());

printf("Введите Максимум (Текущий %i):", sizeb);

cin >> sizeb;

gotoxy(20, wherey());

printf("Масштаб назначен!");

getch();

goto mainmenu;

aboutme:

clrscr();

for(i=0;i<80;i++) printf("=");

printf(" Об Авторе\n");

for(i=0;i<80;i++) printf("=");

printf("\n\

\n\

Виталий Дмитриевич Новичков\n\

Студент группы ЭВМ-2-2\n\

\n\

(C) 2005-2011 Система WohlNET\n \

\n\

\n\

\n\

Проект выполнен по заказу:\n \

Заказчик: МГТУ ГА\n\n\

\

Представитель организации:\n\

Заместитель Декана ФПМиВТ Надейкина Л.А.\n\

\n \

Нажмите любую клавишу для возврата...\n\

\

");

if(!getch())getch();

goto mainmenu;

info:

clrscr();

for(i=0;i<80;i++) printf("=");

gotoxy(wherex()+40-strlen("Краткая помощь")/2, wherey());printf("Краткая помощь\n");

for(i=0;i<80;i++) printf("=");

printf("Условные обозначения:\n\

x - Переменная\n\

Z - Любое допустимое выражение\n\n\

\nИспользуемые Функции\n\

sin(Z) - Синус выражения Z\n\

cos(Z) - Косинус выражения Z\n\

tg(Z) - Тангенс выражения Z\n\

ctg(Z) - Котангенс выражения Z\n\

exp(Z) - e^Z\n\

sqrt(Z) - Квадратный корень выражения Z\n\

log(Z) - Логорифм Z от основания 10\n\

\n\

Используемые операторы\n\

+ - Сложение\n\

- - Вычитание\n\

\* - Умножение\n\

/ - Деление\n\

^ - Возведение в степень\n\

() - Объединение выражения в одно значение\n\

\n\

Нажмите любую клавишу для возврата в главное меню...");

if(!getch())getch();

goto mainmenu;

EnterFormula:

formula:

clrscr();

i=0;

strcpy(s,"");

strcpy(jjk, "");

if (strcmp(s, "")!=0) printf("!ОПАСНО! Срочно выйдите из этой программы!!");

printf("\nВведите формулу или введите \"0\" для возврата в меню\nf(x)=");

fflush(stdin);

cin >> s;

if (strcmp(s, "0")==0) goto mainmenu;

if(mod(s)==1) goto EnterFormula;

if (test()==1) goto mainmenu;

//--------------------------------------------------------------

//------------------- Подключение графики

clrscr();

GDI.on();

graph(s,sid); //Черчение графика

if(!getch()) getch(); //Ожидание нажания клавишы

GDI.off();

//------------------- Очистка памяти

strcpy(sid, sidback);

strcpy(s, "");

goto formula;

//----------------------------

exit:

clrscr();

printf("Программа завершена\n\n");

exit(0);

}

FNLIB.H

//Библиотека функций

#include "goto2x.c" //Спец-переходы курсора

#include "prc.c" //Вывод цветной строки

#include "squear.c" //Закраска выбранной области текстового экрана

#include "txtbrdr.c" //обвести выбранную область текстовой рамочкой

#include "stcolor.c" //Меню, возвращающая номер выбранного цвета

#include "ems.c" //Вывод сообщения об ошибке

#include "nvds.c" //Вступление разработчика

#include "mod.c"

#include "next.c"

#include "skip.c"

#include "numpar.c"

#include "factor.c"

#include "term.c"

#include "expr.c"

#include "test.c"

#include "graph.c" //Вывод графика функции

EMS.C

//Сообщение об ошибке

void ems(char \*s, int k)

{

printf("%s\nВы сделали ошибку (%i), введите формулу еще раз", s, k);

if(!getch()) getch();

}

**GOTO2X.C**

//Переход в указанную позицию по X

void goto25(int yyy=wherey())

{gotoxy(31, yyy);}

void goto27(int yyy=wherey())

{gotoxy(33, yyy);}

expr.c

//Сложение-Вычитание частей формулы

// expr ::= term | expr '+' term | expr '-' term

double expr(char \*s,char \*sid)

{

double res=term(s,sid);

for (;;)

{

if (Oper.Op=='+')

{

if (skip('+',s,sid)==1) return 999999999;

res+=term(s,sid);

}

else

if (Oper.Op=='-')

{

if (skip('-',s,sid)==1) return 999999999;

res-=term(s,sid);

}

else

return res;

}

}

FACTOR.C

//Возведение в степень

// factor ::= numpar | numpar '^' factor

double factor(char \*s,char \*sid)

{

double res=numpar(s,sid);

if (Oper.Op=='^')

{

skip('^',s,sid);

res=pow(res,factor(s,sid));

}

return res;

}

Mod.c

//Распознавание скобок

int mod(char \*s)

{

int a,j,n=0,i=0;

a=strlen(s);

for(i=0;s[i]!='\0';i++)

{

if(s[i]=='-')

{

if(s[i-1]=='(')

{

for (j=0;j!=a+1-i;j++) s[(a+1)-j]=s[a-j];

s[i]='0';

}

}

}

a=strlen(s);

for(i=0;i<a;i++)

{

if(s[i]=='(') n=n+1;

if(s[i]==')') n=n-1;

}

if(n!=0)

{

closegraph();

ems(s, 1);

return 1;

}

return 0;

}

prc.C

//Вывод цветной строки

void prc(char \*str, int c1=0, int c2=7)

{

//Назначение выбранных пользователем цветов

textbackground(c1); //выделения текста (фона)

textcolor(c2); //и самого текста

//вывод строки

cprintf(str);

//Возвращение параметров по умолчанию:

textcolor(7); //цвет текста светло-серый

textbackground(0); //цвет экрана чёрный

}

NEXT.C

//Расчет следующей точки на графике

int next(char \*s,char \*sid) //Функция Next

{

int c,count=0;

static int i=0;

char cbuf[50];

if(Oper.Op=='╞') i=0;

for (;;)

{

c=s[i++];

if(strchr("+-\*/^()\n",c)!=NULL&&i!=1)

{

if(c=='\*'&&s[i]=='\*')

{

i=i+1;

return Oper.Op='^';

}

return Oper.Op=c;

}

if(isspace(c)) continue;

if(isdigit(c)||c=='.'||c=='x')

{

for(;(c=='x'||(c>='0'&&c<='9')||c=='.');)

{

if(c=='x')

{

if(s[i-i]=='-') Oper.tokval=-1\*Oper.x;

else

Oper.tokval=Oper.x;

return Oper.Op='n';

}

cbuf[count++]=c;

cbuf[count]='\0';

c=s[i++];

}

i-=1;

Oper.tokval=atof(cbuf);

return Oper.Op='n';

}

if(isalpha(c))

{

if(c=='p'&&s[i++]=='i')

{

Oper.tokval=pi;

return Oper.Op='n';

}

do

{

cbuf[count++]=c;

cbuf[count]='\0';

c=s[i++];

}

while(isalpha(c));

strcpy(sid,cbuf);

i--;

return Oper.Op='a';//alpha function

}

}

}

NVDS.C

/\*

Собственная Титульная заставка автора

NVDS.C

CopyRight (C) 2005-2012 NVD's Privacy Servers

Виталий Новичков, студент МГТУ ГА, ФПМВТ, ЭВМ-2-2

\*/

void nvd\_logo()

{

register int y, z, PAGE, soundhz, counter; clrscr();

char string[][50]=

{

"NVD's Privacy Servers",

" и "," "," Домашнаяя "," Сеть ",

" Wohlhhabenden.NET ","\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ "," В разработке "," Новичкова ",

" Виталия "," Дмитриевича "," "," Студента ",

" МГТУ ГА ЭВМ-2-2 "," "," Представляют... "," ",

" "," "," Декабрь 2011 "," "

};

textcolor(0);

textbackground(0);

printf("\n\n\n\n");

counter=0;

\_setcursortype(\_NOCURSOR);

for (y=0, soundhz=0, z=14, PAGE=0; counter<3; PAGE++, y++,soundhz+=500, z--)

{

if (z==8) z=14; textbackground(z); printf(" ");

cprintf(" "); textbackground(7);

textcolor(y); cprintf(" "); cprintf(string[PAGE], y);

cprintf(" "); textbackground(z); cprintf(" ", y);

cprintf("\r\n"); sound(soundhz); delay(50);

if (y>=16){gotoxy(1, 6);counter++;y=0;soundhz=0;PAGE=0;}

}

nosound(); textbackground(0); textcolor(7); cprintf("\n"); sleep(2); \_setcursortype(\_NORMALCURSOR);

clrscr();

}

//Тест механизма расчёта функций

int test()

{

char s[]="(sin(pi)^2+cos(pi)^2)";

char sid[6];

next(s,sid);

if(sin(pi)\*sin(pi)+cos(pi)\*cos(pi)!=expr(s,sid))

{ clrscr(); printf("\nTest failed");

if(!getch()) getch();

return 1; }

Oper.Op='╞'; Oper.tokval=0; return 0;

}

TXTBRDR.C

//Обвести выбранную область в рамочку

void textborder(int bx, int by, int cx, int cy,int color=7)

{

gotoxy(bx,by);

textcolor(color);

cprintf("╔");

gotoxy(bx+1,by);

for(i=bx+1;i<cx;i++)

{

gotoxy(i,by);

cprintf("═");

}

gotoxy(cx,by);

cprintf("╗");

for(i=by+1;i<cy;i++)

{

gotoxy(cx,i);

cprintf("║");

}

gotoxy(cx,cy);

cprintf("╝");

for(i=cx-1;i>bx;i--)

{

gotoxy(i,cy);

cprintf("═");

}

gotoxy(bx,cy);

cprintf("╚");

for(i=cy-1;i>by;i--)

{

gotoxy(bx,i);

cprintf("║");

}

}

SQUEAR.C

//Закрасить выбранную область

void sqwear(int x1, int y1, int x2, int y2,int cc=0)

{ int i,j; int posx=wherex(), posy=wherey();

gotoxy(x1,y1); for(i=y1;i<=y2;i++)

{ for(j=x1; j<=x2; j++) {gotoxy(j,i);prc(" ",cc,8);} }

gotoxy(posx, posy);}

SKIP.C

//Пропустить текущую точку на графике

int skip(int t,char \*s,char \*sid)

{ if(Oper.Op!=t) { closegraph(); ems(s, 2); return 1;

} next(s,sid); return 0; }

TERM.C

//Распознавание умножения/деления

// term ::= factor | term '\*' factor | term '/' factor

double term(char \*s,char \*sid)

{

double temp;

double res=factor(s,sid);

for (;;)

{

if (Oper.Op=='\*')

{

if(skip('\*',s,sid)==1) return 999999999;

res\*=factor(s,sid);

}

else

if (Oper.Op=='/')

{

if(skip('/',s,sid)==1) return 999999999;

temp=factor(s,sid);

if(temp!=0) res/=temp;

else

{

temp=temp+0.001;

res/=temp;

}

}

else

return res;

}

}

STCOLOR.C

//Меню назначения цвета выбранного элемента

int stcolor(int current=0)

{

int pos=1,key=0;

pos=current;

//Стрелка вверх - 0/72

//Стрелка вниз - 0/80

//Стрелка вправо- 0/77

//Стрелка влево - 0/75

Menu:

\_setcursortype(\_NOCURSOR);

goto25(3);if(pos==0) prc("\*",7,0);else prc(" ",0,7);

goto27(3);prc("Черный 0",7,0);

goto25(4);if(pos==1) prc("\*",1,15);else prc(" ",0,7);

goto27(4);prc("Темно-Синий 1",0,1);

goto25(5);if(pos==2) prc("\*",2,10);else prc(" ",0,7);

goto27(5);prc("Зеленый 2",0,2);

goto25(6);if(pos==3) prc("\*",3,10);else prc(" ",0,7);

goto27(6);prc("Бирюзовый 3",0,3);

goto25(7);if(pos==4) prc("\*",4,10);else prc(" ",0,7);

goto27(7);prc("Красный 4",0,4);

goto25(8);if(pos==5) prc("\*",5,10);else prc(" ",0,7);

goto27(8);prc("Малиновый 5",0,5);

goto25(9);if(pos==6) prc("\*",6,10);else prc(" ",0,7);

goto27(9);prc("Коричневый 6",0,6);

goto25(10);if(pos==7) prc("\*",7,10);else prc(" ",0,7);

goto27(10);prc("Св.-Серый 7",0,7);

goto25(11);if(pos==8) prc("\*",0,8);else prc(" ",0,7);

goto27(11);prc("Тм.-Серый 8",0,8);

goto25(12);if(pos==9) prc("\*",1,9);else prc(" ",0,7);

goto27(12);prc("Св.-синий 9",0,9);

goto25(13);if(pos==10) prc("\*",2,10);else prc(" ",0,7);

goto27(13);prc("Св.-Зелный A",0,10);

goto25(14);if(pos==11) prc("\*",3,11);else prc(" ",0,7);

goto27(14);prc("Голубой B",0,11);

goto25(15);if(pos==12) prc("\*",4,12);else prc(" ",0,7);

goto27(15);prc("Алый C",0,12);

goto25(16);if(pos==13) prc("\*",5,13);else prc(" ",0,7);

goto27(16);prc("Розовый D",0,13);

goto25(17);if(pos==14) prc("\*",6,14);else prc(" ",0,7);

goto27(17);prc("Желтый E",0,14);

goto25(18);if(pos==15) prc("\*",7,15);else prc(" ",0,7);

goto27(18);prc("Белый F",0,15);

sqwear(32,3,32,18,0);

textborder(30,2,49,19,3);

gotoxy(2,25);

printf("Вверх/Вниз - выбор; Esc/Влево - Отмена; Enter/Вправо - Принять");

key=getch();

\_setcursortype(\_NORMALCURSOR);

switch(key)

{

case 0:

switch(getch())

{

//Стрелка вверх - 0/72

//Стрелка вниз - 0/80

//Стрелка вправо- 0/77

//Стрелка влево - 0/75

case 72:

if(pos>=0)pos--;

if(pos==-1)pos=15;

goto Menu;

case 80:

if(pos<=15)pos++;

if(pos==16)pos=0;

goto Menu;

case 75:

sqwear(30,2,49,19,0);

gotoxy(2,25);for(i=0;i<67;i++)printf(" ");

return current;

case 77:

sqwear(30,2,49,19,0);

gotoxy(2,25);for(i=0;i<67;i++)printf(" ");

return pos;

}

case 27:

sqwear(30,2,49,19,0);

gotoxy(2,25);for(i=0;i<67;i++)printf(" ");

return current;

case 13:

sqwear(30,2,49,19,0);

gotoxy(2,25);for(i=0;i<67;i++)printf(" ");

return pos;

case '0':case '1':case '2':case '3':

case '4':case '5':case '6':case '7':case '8':

case '9':case 'a':case 'A':case 'b':case 'B':

case 'c':case 'C':case 'd':case 'D':case 'e':

case 'E':case 'f':case 'F':

sqwear(30,2,49,19,0);

gotoxy(2,25);for(i=0;i<67;i++)printf(" ");

switch(key)

{

case '0':

return 0;

case '1':

return 1;

case '2':

return 2;

case '3':

return 3;

case '4':

return 4;

case '5':

return 5;

case '6':

return 6;

case '7':

return 7;

case '8':

return 8;

case '9':

return 9;

case 'A':case 'a':

return 10;

case 'B':case 'b':

return 11;

case 'C':case 'c':

return 12;

case 'D':case 'd':

return 13;

case 'E':case 'e':

return 14;

case 'F':case 'f':

return 15;

}

}

return current;

}

NUMPAR.C

//Распознавание математических функций

// numpar ::= number | '(' expr ')' | (func1||...||func2) '(' expr ')'

double numpar(char \*s,char \*sid)

{

int i=0;

if (Oper.Op=='n')

{

double res=Oper.tokval;

skip('n',s,sid);

return res;

}

if (Oper.Op=='a')

{

if(skip('a',s,sid)==1) return 999999999;

if(skip('(',s,sid)==1) return 999999999;

double res=expr(s,sid);

if(skip(')',s,sid)==1) return 999999999;

for(i=0;(strcmp(sid,Oper.sid[i])!=0);i++);

switch(i)

{

case 0: res=sin(res); break;

case 1: res=cos(res); break;

case 2: res=sin(res)/cos(res); break;

case 3: res=cos(res)/sin(res); break;

case 4: if(res>=0) res=sqrt(res);

else

{

res=res+getmaxx()/2;

res=sqrt(res);

}

break;

case 5: if(res>=0) res=log(res);

else

{

res=res+getmaxx()/2;

res=log(res);

}

break;

case 6: res=exp(res); break;

default: closegraph(); printf("\nBad function: %s\n",sid);res=0;

}

return res;

}

if(skip('(',s,sid)==1) return 999999999;

double res=expr(s,sid);

if(skip(')',s,sid)==1) return 999999999;

return res;

}

GRAPH.C

//Рисование декартовых координат и вычисление графика функции

int graph(char \*s,char \*sid)

{

float a,b,i;

int xmax,ymax,px,py,x0,y0;

a=sizea;

b=sizeb;

xmax=getmaxx()+1;

ymax=getmaxy()+1;

px=100;

py=100;

x0=xmax/2;

y0=ymax/2;

for(;;)

{

if(((xmax/2+b\*px)>xmax)||((xmax/2+a\*px)<0))

{

next(s,sid);

putpixel(xmax/2+Oper.x\*px,ymax/2-expr(s,sid)\*py, color5);

px=px-1;

py=py-1;

Oper.Op='╞';

Oper.tokval=0;

continue;

}

setfillstyle(SOLID\_FILL, color1);

bar(0,0,xmax,ymax);

setfillstyle(SOLID\_FILL, color1);

setcolor(color2);

setlinestyle(DASHED\_LINE,0,2);

for(i=x0;i<xmax;i+=px) line(i,0,i,ymax);

for(i=x0;i>0;i-=px) line(i,0,i,ymax);

for(i=y0;i>0;i-=py) line(0,i,xmax,i);

for(i=y0;i<ymax;i+=py) line(0,i,xmax,i);

setlinestyle(DOTTED\_LINE,0,2);

for(i=x0;i<xmax;i+=(px/10)) line(i,0,i,ymax);

for(i=x0;i>0;i-=(px/10)) line(i,0,i,ymax);

for(i=y0;i>0;i-=(py/10)) line(0,i,xmax,i);

for(i=y0;i<ymax;i+=(py/10)) line(0,i,xmax,i);

setcolor(color3);

setlinestyle(SOLID\_LINE,0,3);

line(0,y0, xmax,y0);

line(x0,0,x0,ymax);

line(xmax, y0, xmax-10,y0+5);

line(xmax, y0, xmax-10,y0-5);

line(xmax, y0, xmax-10,y0+3);

line(xmax, y0, xmax-10,y0-3);

line(xmax-10, y0+5, xmax-10, y0-5);

line(x0, 0, x0-5, 0+10);

line(x0, 0, x0+5, 0+10);

line(x0, 0, x0-3, 0+10);

line(x0, 0, x0+3, 0+10);

line(x0-5, 0+10, x0+5,0+10);

setcolor(color5);

setlinestyle(SOLID\_LINE,0,3);

settextstyle(0,0,1);

outtextxy(x0-18,y0+7,"0");

for(i=x0;i<xmax;i+=px)

circle(i,y0,2);

for(i=x0;i>0;i-=px)

circle(i,y0,2);

for(i=y0;i>0;i-=py)

circle(x0,i,2);

for(i=y0;i<ymax;i+=py)

circle(x0,i,2);

for(i=x0, jjj=0;i<xmax; i+=px,jjj++)

if((jjj!=0)&&i<xmax-40){sprintf(jjk, "%i",jjj); outtextxy(i,y0+14, jjk);}

for(i=x0, jjj=0;i>0; i-=px,jjj--)

if((jjj!=0)&&i>20){sprintf(jjk, "%i",jjj); outtextxy(i,y0+14, jjk);}

for(i=y0, jjj=0;i>0; i-=py,jjj++)

if((jjj!=0)&&i>40){sprintf(jjk, "%i",jjj); outtextxy(x0-20,i, jjk);}

for(i=y0, jjj=0;i<ymax-40; i+=py,jjj--)

if((jjj!=0)&&i<ymax-40){sprintf(jjk, "%i",jjj); outtextxy(x0-20,i, jjk);}

outtextxy(xmax-25,y0+14,"X");

outtextxy(x0-18,0+25,"Y");

settextstyle(4,0,3);

outtextxy(12,12,s);

setcolor(color4);

setlinestyle(SOLID\_LINE,0,3);

for(Oper.x=a; Oper.x<b; Oper.x+=(b-a)/xmax)

{ next(s,sid);

y\_dot = expr(s,sid);

x\_dot = Oper.x;

xcheck = ((xmax/2+x\_dot\*px>20)&&(xmax/2+x\_dot\*px<xmax-20));

ycheck = (ymax/2-y\_dot\*py>20)&&(ymax/2-y\_dot\*py<ymax-20)&&xcheck;

if(x\_dot == a)

{if(ycheck) putpixel(xmax/2+x\_dot\*px, ymax/2-y\_dot\*py,color4);}

Else

if(ycheck) {if(ycheck2) line(xmax/2+x\_dot\*px, ymax/2-y\_dot\*py, xpr, ypr);

}

xpr=xmax/2+x\_dot\*px;

ypr=ymax/2-y\_dot\*py;

ycheck2=ycheck;

delay(5);

Oper.Op='╞';

Oper.tokval=0;

}

break;

}

return 0;

}

Приложение 2. Экранные формы.

Рисунок П.2.1 Приветствие

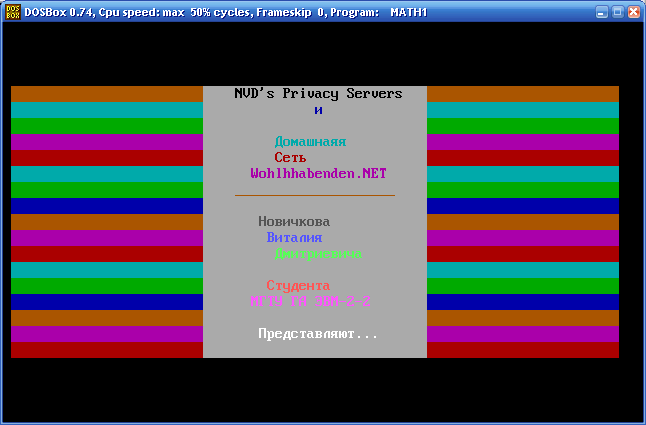


Рисунок П.2.2 Главное меню (негатив снимка)

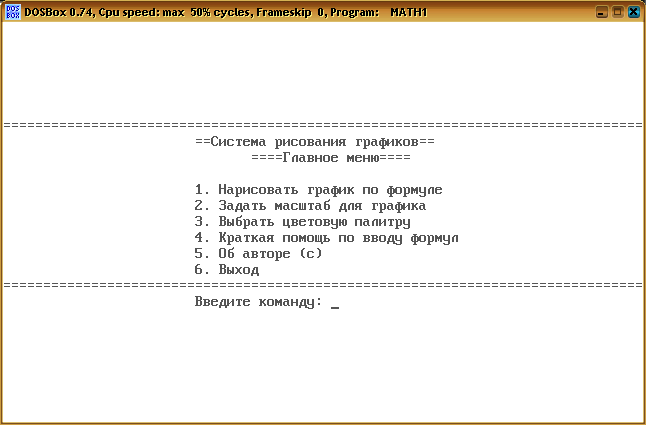


Рисунок П.2.3 Интерфейс выбора цвета (выбор элемента)

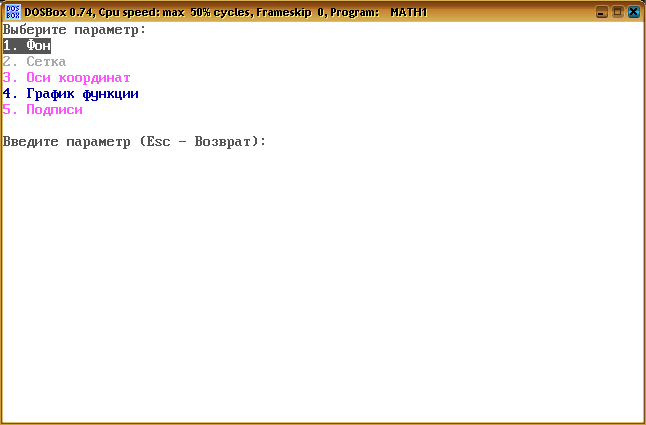


Рисунок П.2.3 Интерфейс выбора цвета (назначение цвета на элемент)

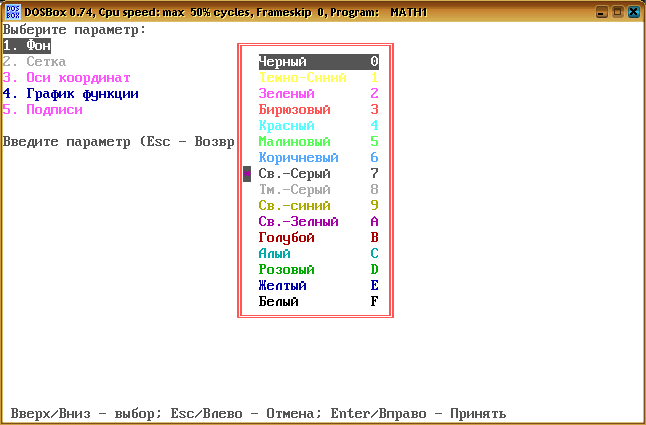


Рисунок П.2.4 Интерфейс ввода формулы

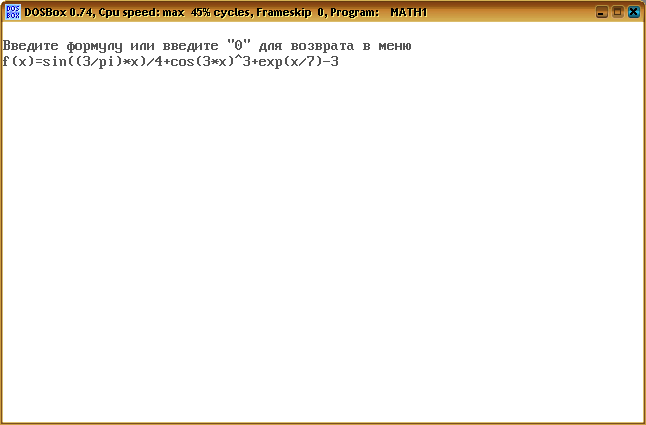


Рисунок П.2.5 Вывод графика (С сеткой (цвета сетки и фона разные))

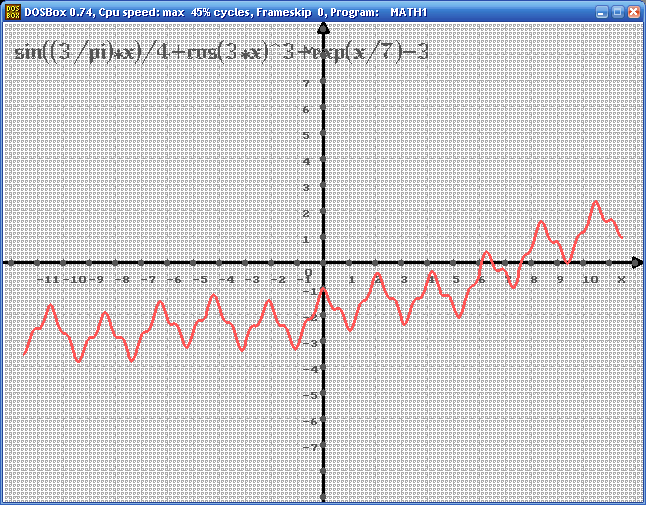


Рисунок П.2.6 Вывод графика (без сетки (цвета сетки и фона совпадают))

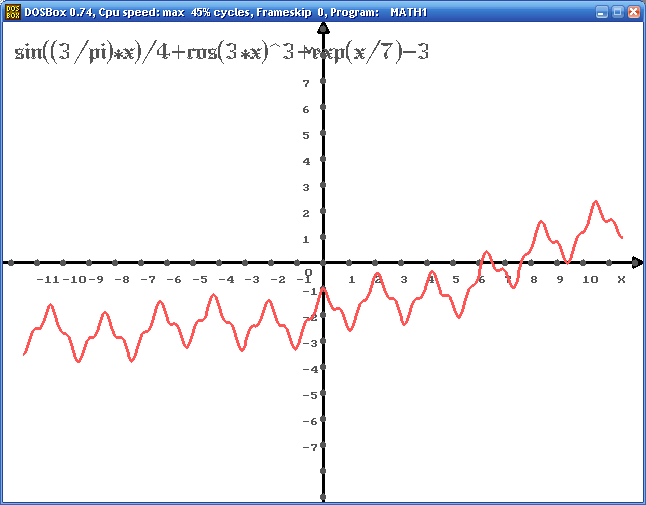


Рисунок П.2.7 Вывод краткой справки по вводу формул

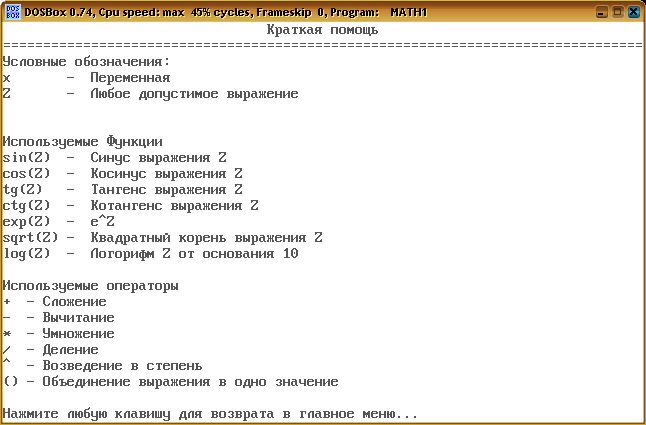
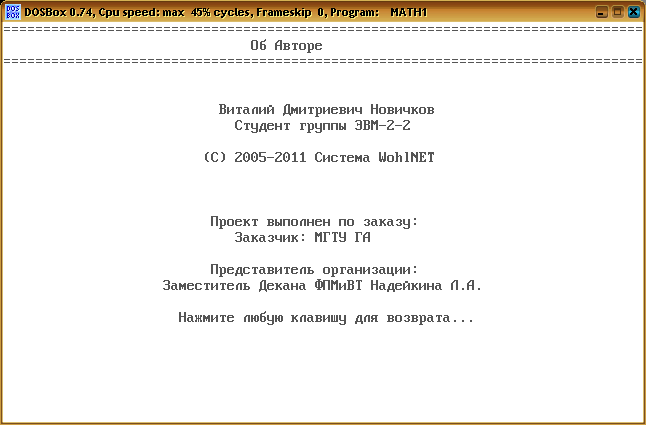


Рисунок П.2.8 Вывод информации об авторе



СОДЕРЖАНИЕ

1. Техническое задание на разработку системы 2
2. Возможности разработанной системы 3
3. Алгоритмы функций обработки данных 4
   1. Главная функция программы 4
   2. Функция **graph** 8
   3. Функция **goto25** 9
   4. Функция **goto27** 9
   5. Функция **ems** 10
   6. Функция **prc** 10
   7. Функция **factor** 10
   8. Функция **squear** 10
   9. Функция **test** 11
   10. Функция **mod** 11
   11. Функция **expr** 12
   12. Функция **term** 12
   13. Функция **skip** 12
   14. Функция **numpar** 13
4. Программа системы
   1. Файлы программы 14
   2. Глобальные переменные программы 15
   3. Структура программы 15
5. Руководство по использованию 16

Приложение 1. Исходные коды программы 17

Приложение 2. Экранные формы 36