Министерство образования и науки Российской Федерации Федеральное агентство по образованию Федеральное государственное бюджетное образовательное учреждение высшего образования «Вятский государственный университет»

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Лабораторная работа №3

по курсу «Программирование»

**Изучение базовых принципов организации процедур и функций**

Выполнил студент группы ИВТ-11\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Птахова А.М/

Проверил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Чистяков Г. А./

Киров 2020

**Цель работы:** Освоить синтаксис построения процедур и функций, изучить способы передачи данных в подпрограммы, получить навыки организации минимального пользовательского интерфейса.

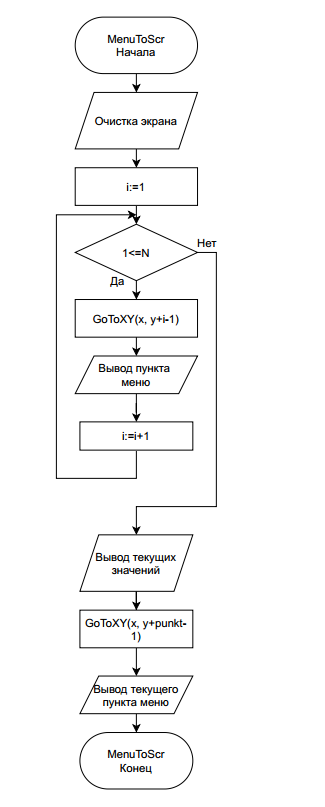
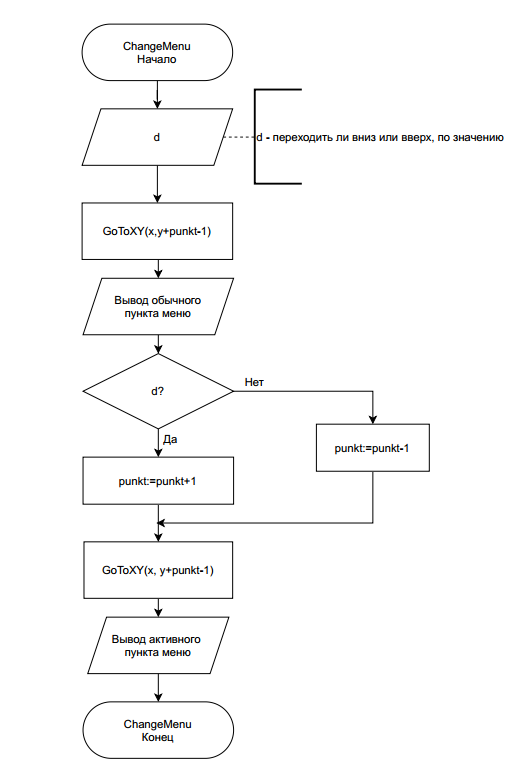
**Задание:**

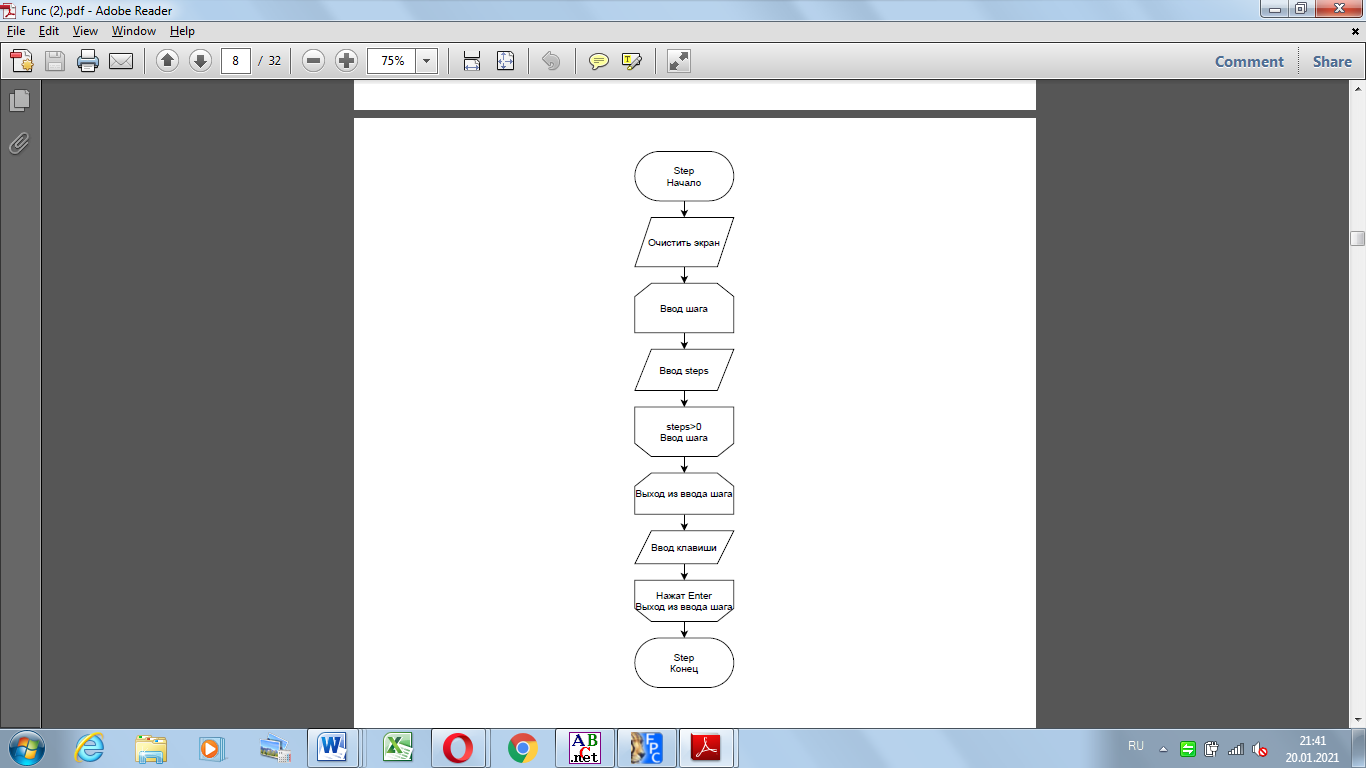
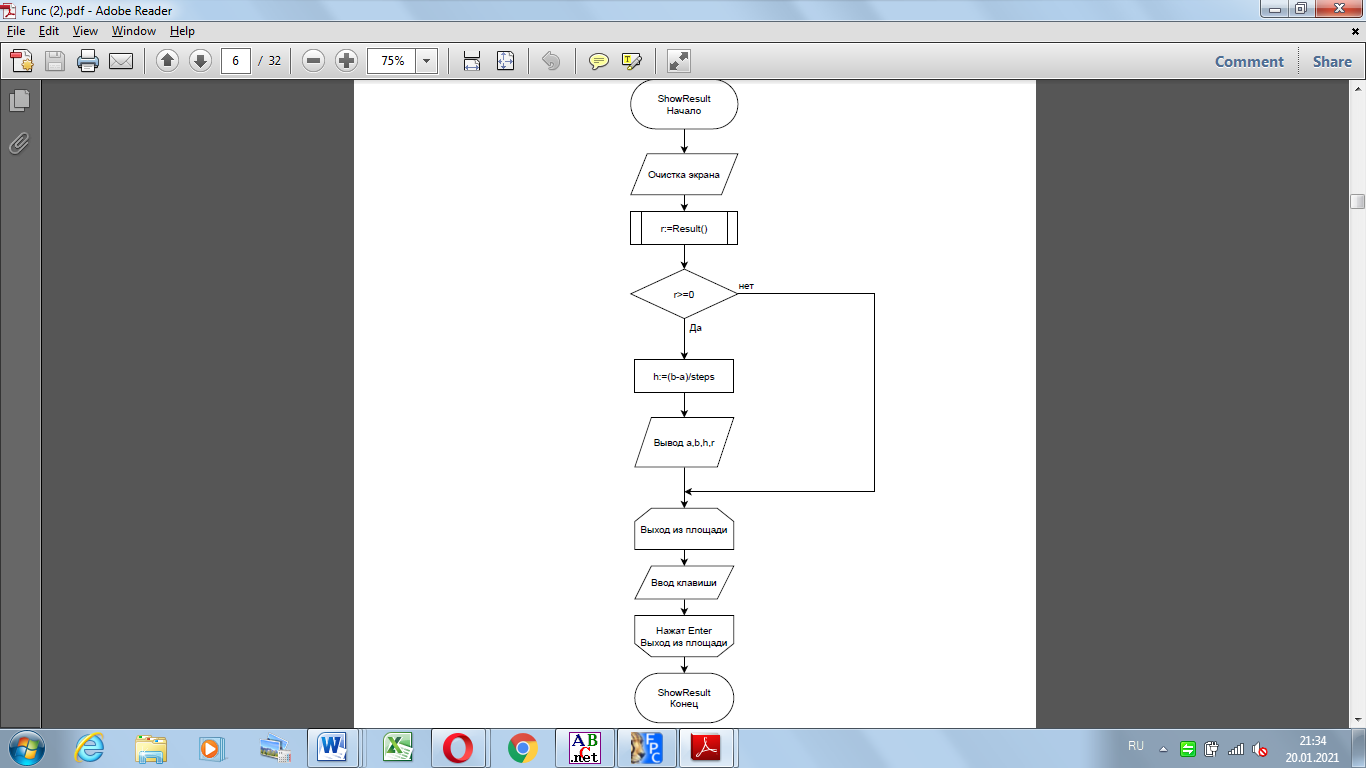
1. Реализовать программу вычисления площади фигуры, ограниченной кривой

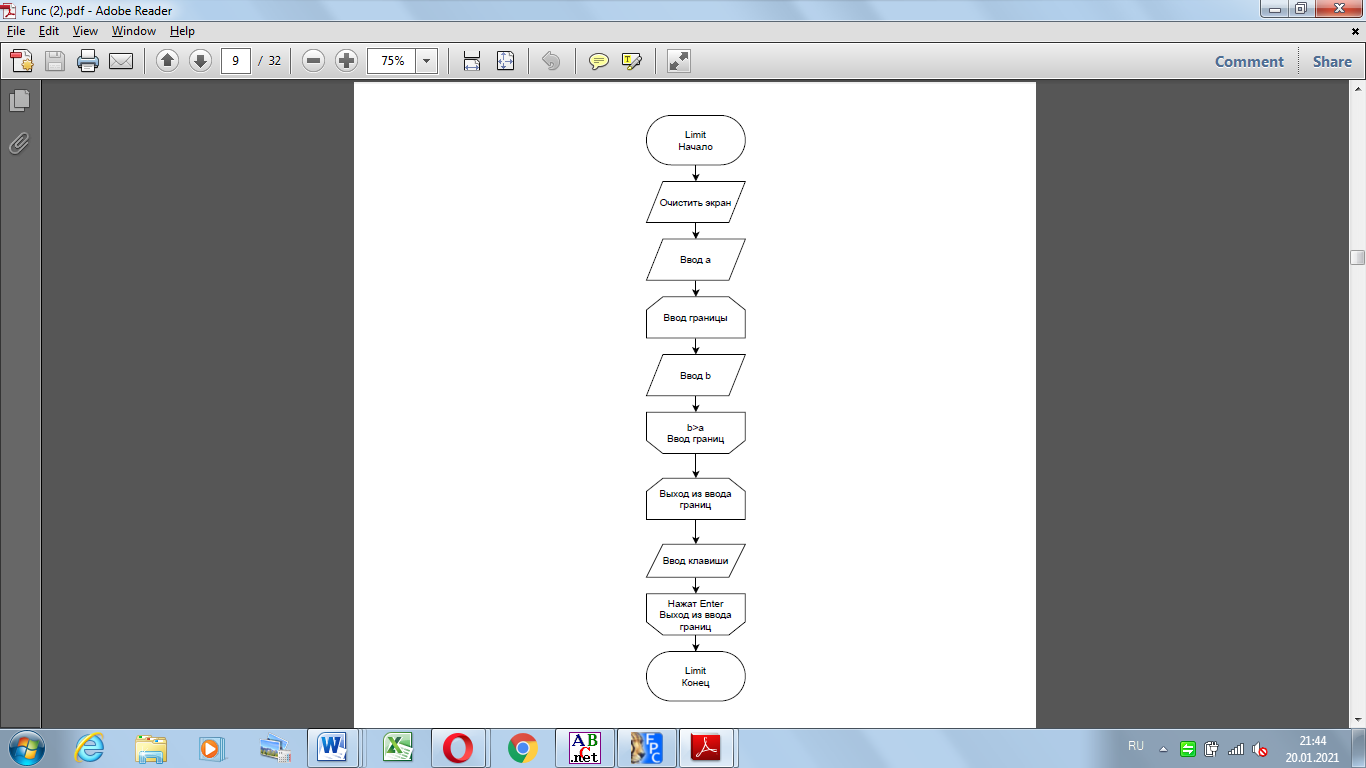
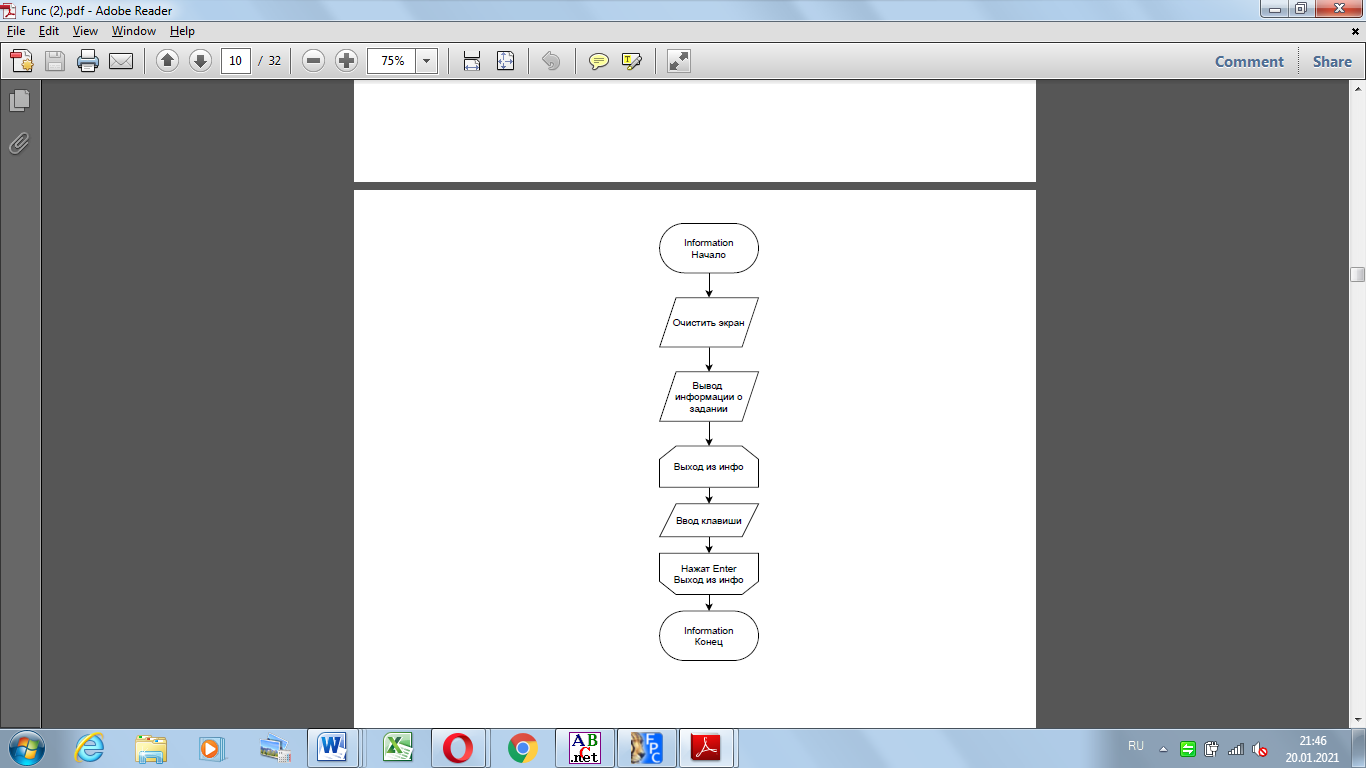
x ^ 3 + (-1)\*x ^ 2 + (-2) \* x + (1) и осью OX (в положительной части по оси OY).

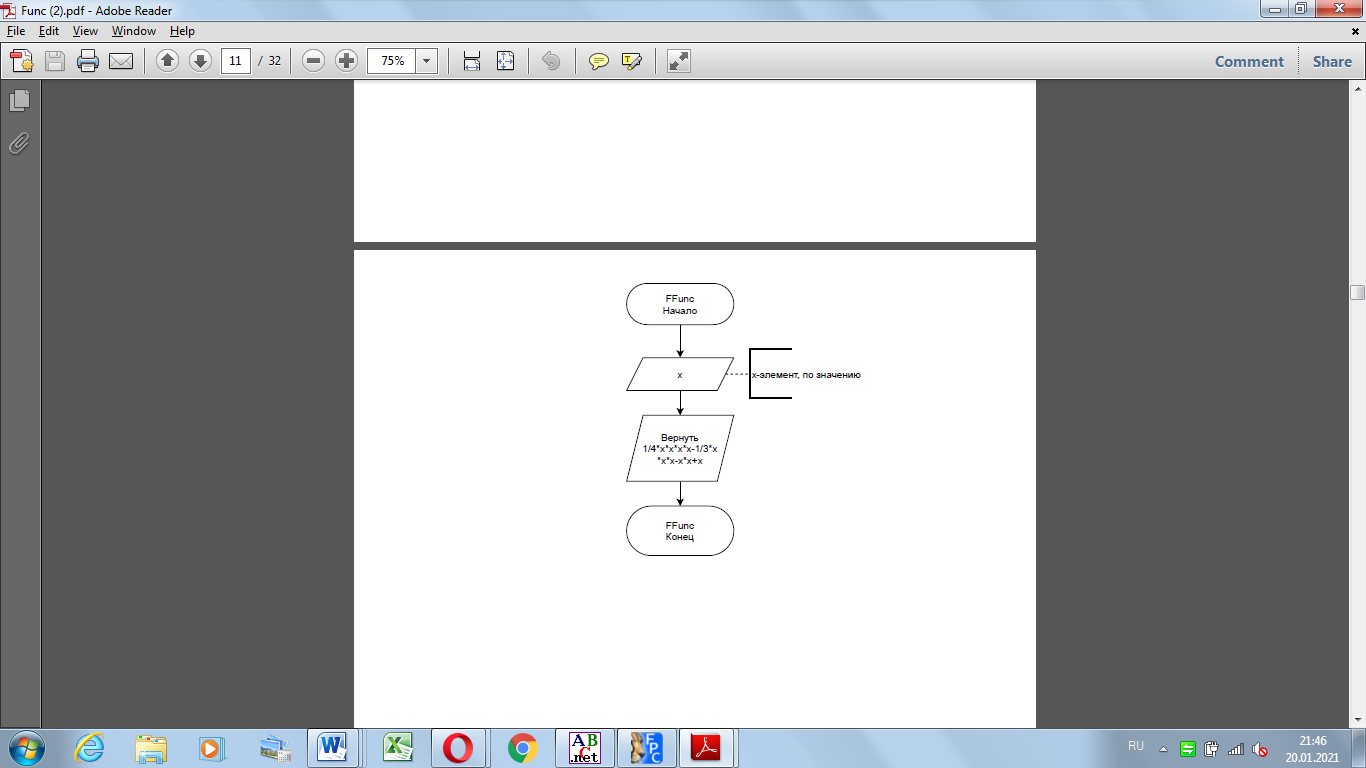
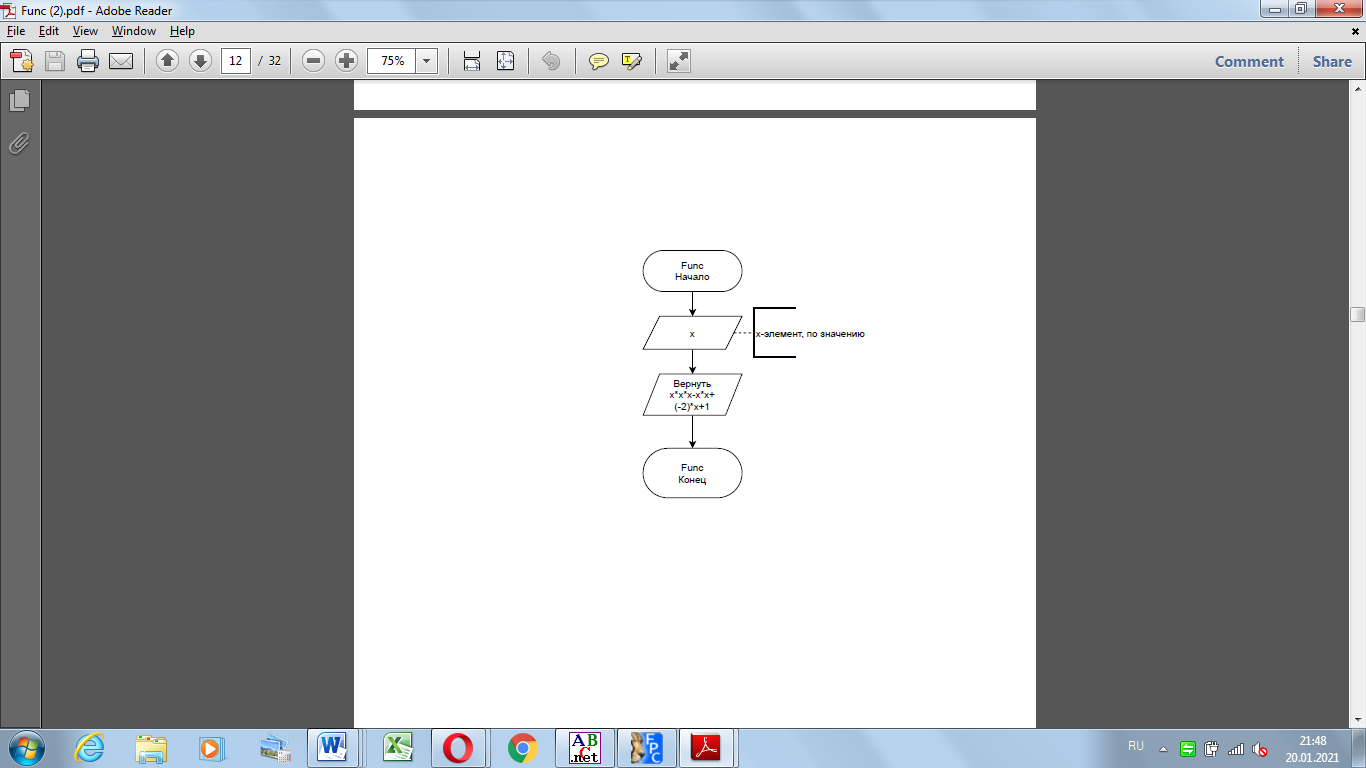
1. Вычисление определенного интеграла должно выполняться численно, с применением метода правых прямоугольников.
2. Пределы интегрирования вводятся пользователем.
3. Взаимодействие с пользователем должно осуществляться посредством case-меню.
4. Требуется реализовать возможность оценки погрешности полученного результата.
5. Необходимо использовать процедуры и функции там, где это целесообразно

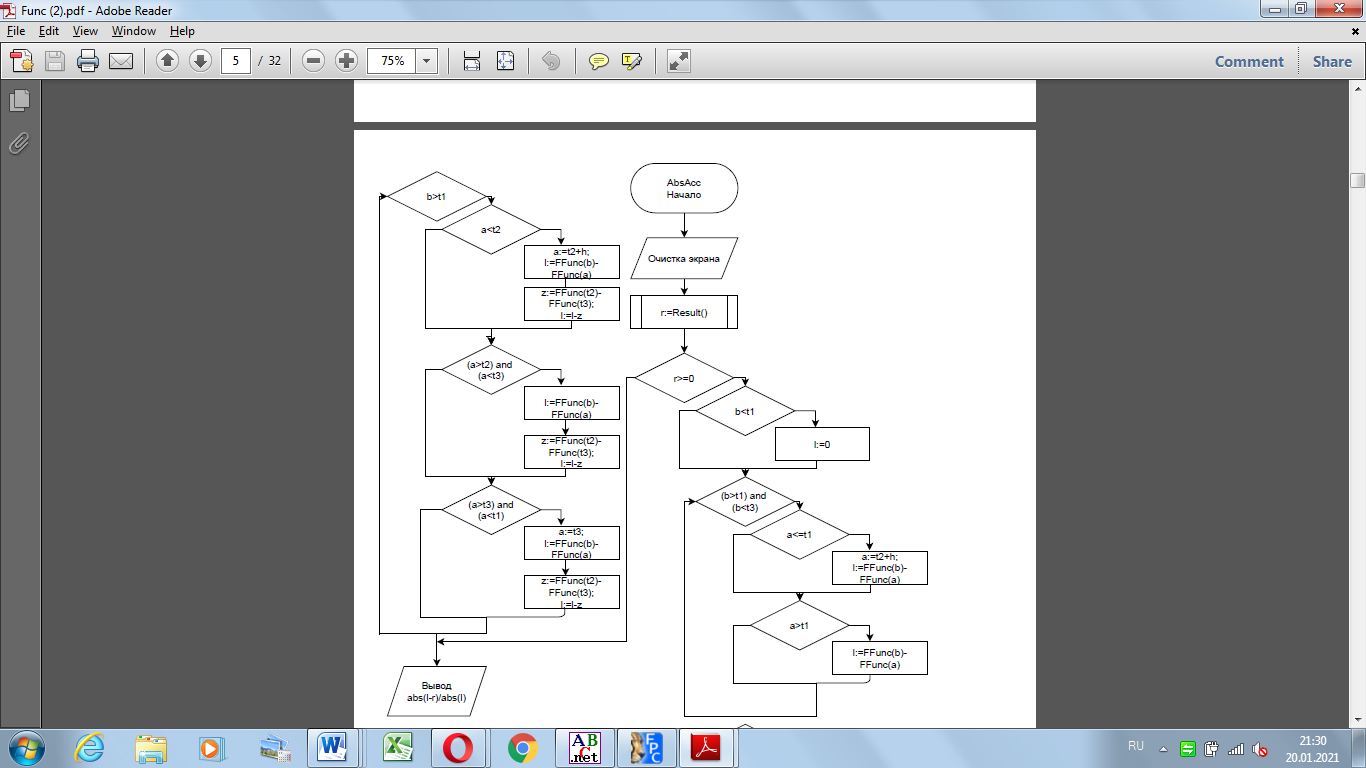
**Схема Алгоритмов**

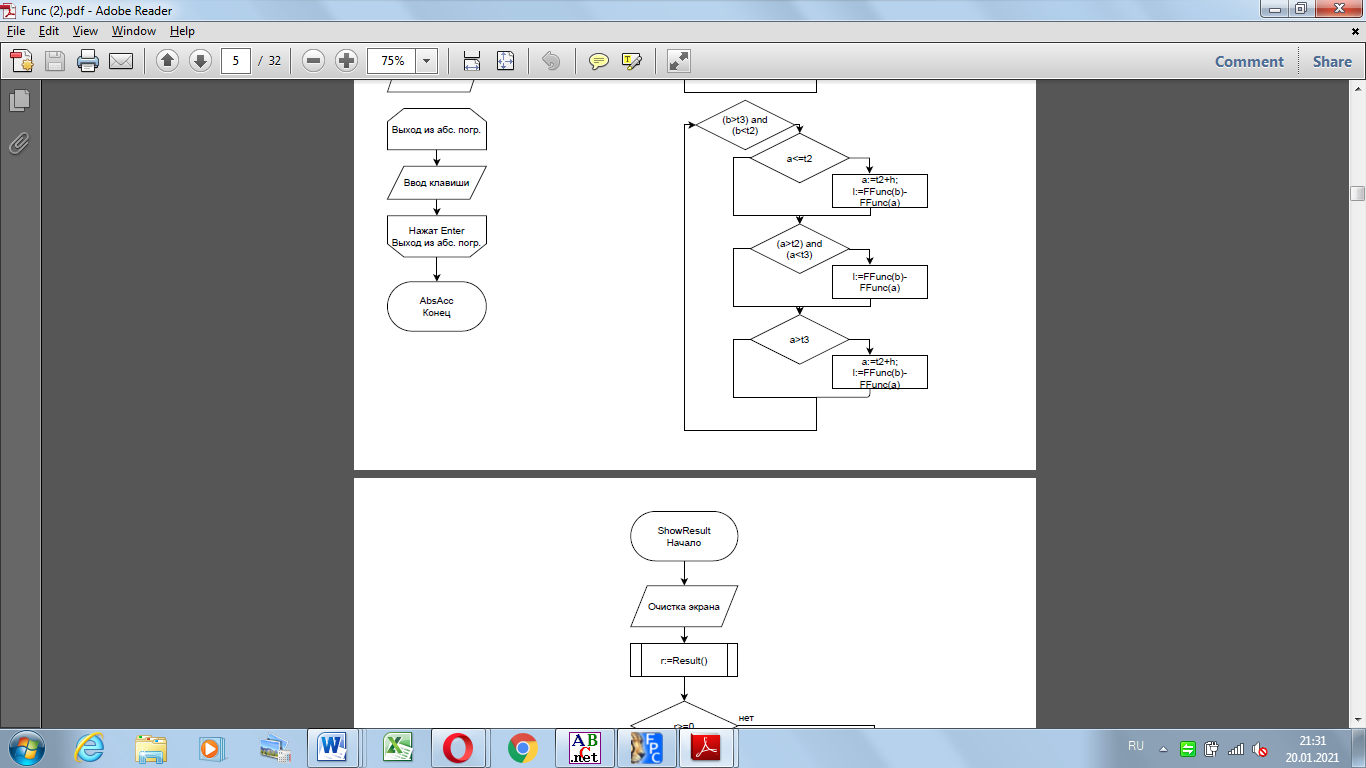


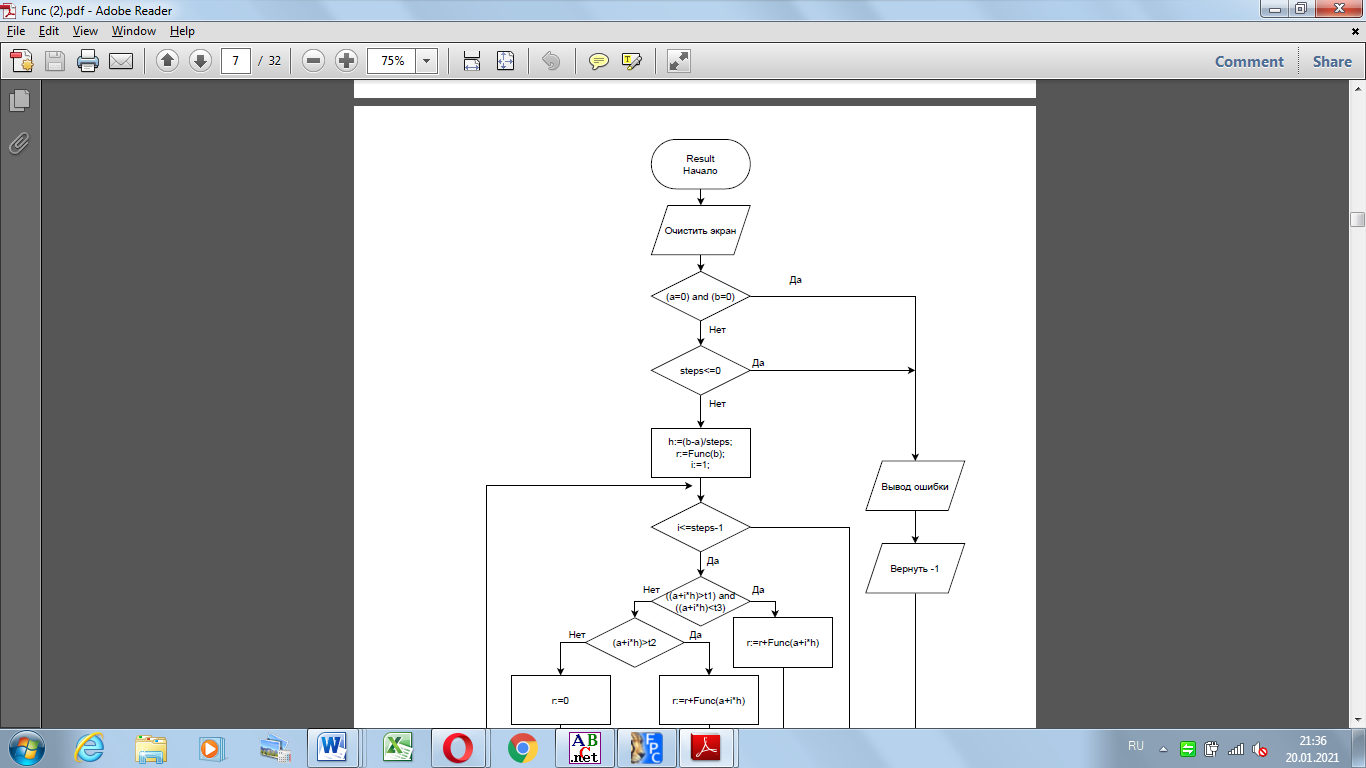


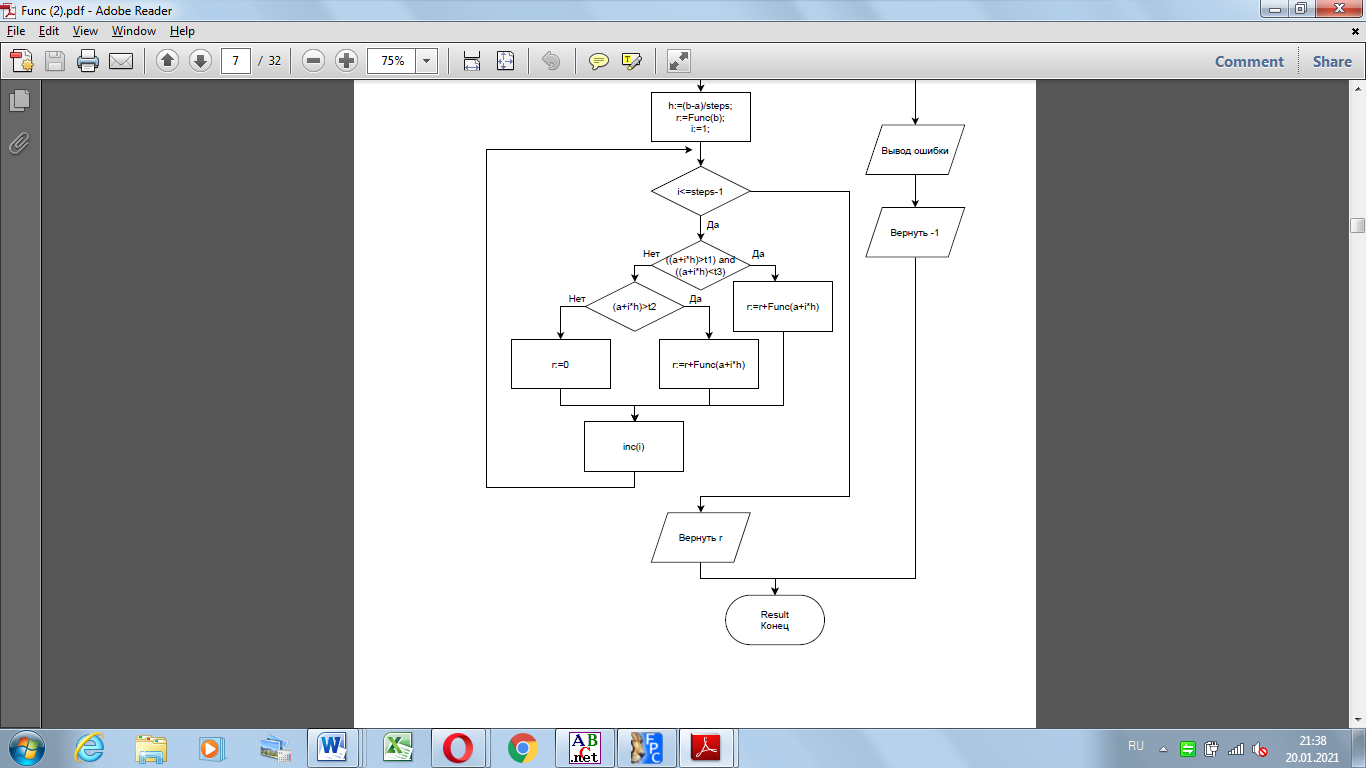


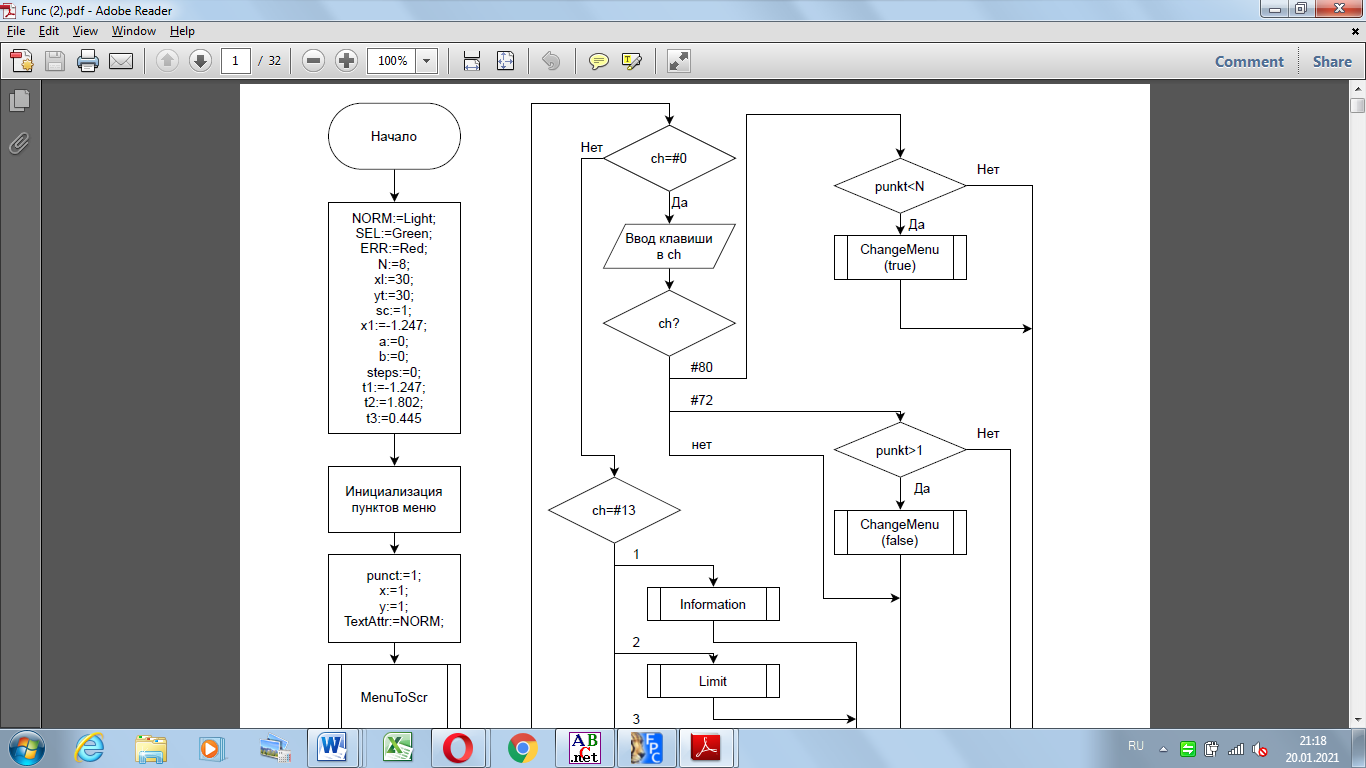
 

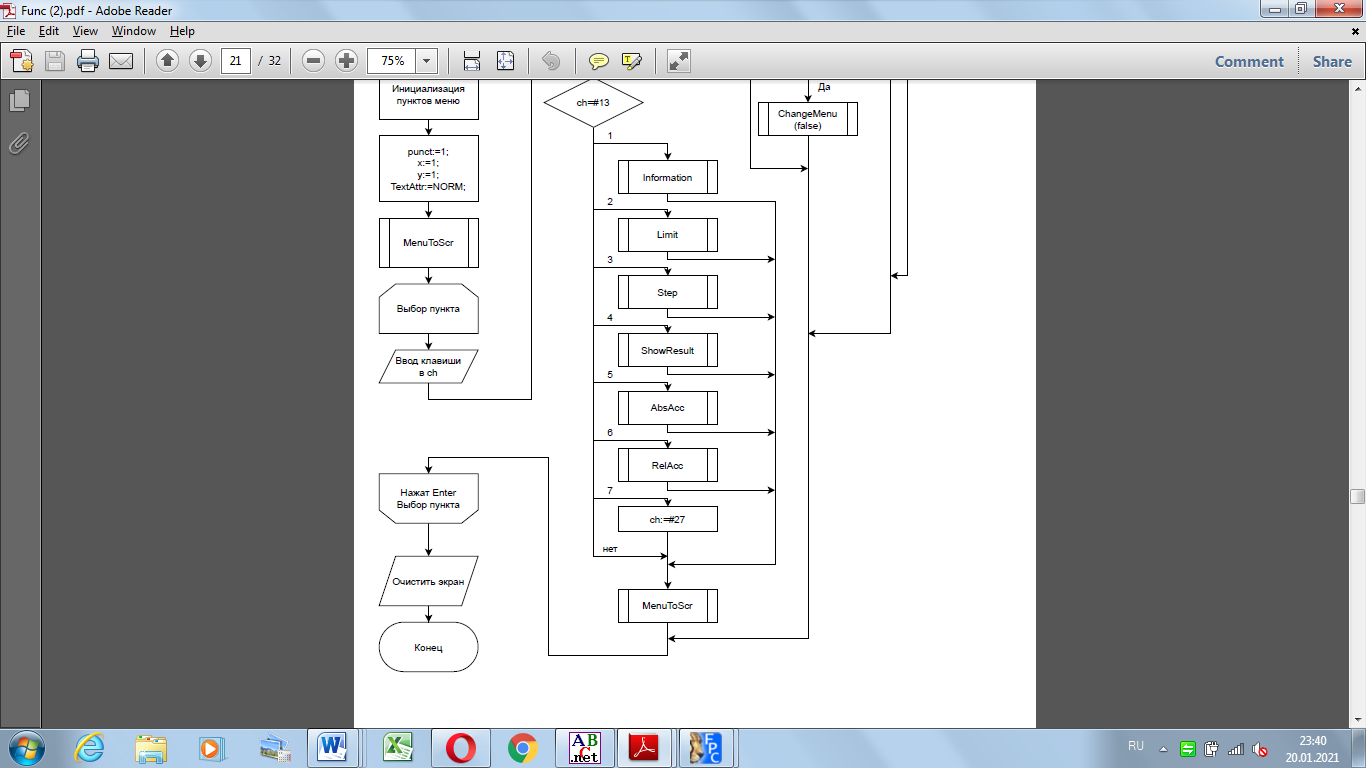












**Листинг программы**

program lab3;

uses Crt;

const

NORM=LightGray;

SEL=Green;

ERR=Red;

N=8;

x1=-1.247;

t1=-1.247;

t2=1.802;

t3=0.445;

eps=1e-6;

sc=1;

xl=30;

yt=30;

var

menu:array[1..N] of string;

punkt:integer;

ch:char;

Gd,Gm:integer;

my,mx:real;

ay,by:integer;

x0,y0:integer;

s:string;

x,y:integer;

a: real=0;

b: real=0;

c: real=0;

d: real=0;

i: integer;

steps: integer=0;

xr,yd:integer;

Procedure WriteError(str:string);

begin

TextAttr:=ERR;

writeln(str);

TextAttr:=NORM;

end;

Procedure WriteSuccess(str:string);

begin

TextAttr:=SEL;

writeln(str);

TextAttr:=NORM;

end;

Function Func(x: real):real;

begin

Func:=x\*x\*x-x\*x+(-2)\*x+1;

end;

Function FFunc(x: real):real;

begin

FFunc:=1/4\*x\*x\*x\*x-1/3\*x\*x\*x-x\*x+x;

end;

//show information

Procedure Information;

begin

ClrScr;

writeln('The task:');

writeln('1. Implement a program for calculating the area of a figure bounded by a curve');

writeln('x^3+(2)-x^2+(-2)\*x+(1) and an x-axis(in the positive y-axis).');

writeln('2. The calculation of the definite integral must be performed numerivally, using the');

writeln('right rectangle method.');

writeln('3. Integration limirs are entered by the user.');

writeln('4. Interaction with the user should be done through the case-menu.');

writeln('5. It is required to realize the possibility of evaluating the аccuracy of the obtained result.');

writeln('6. Procedures and functions should be used where appropriate.');

WriteSuccess('Enter <Enter> for continue');

repeat

ch:= readkey;

until ch=#13;

end;

//enter limit of integration

Procedure Limit;

begin

ClrScr;

write('Enter the start of integration: ');

Readln(a);

repeat

write('Enter the end of integration: ');

Readln(b);

if b <= a then

WriteError('The number must be greater than the beginning!');

until (b>a);

WriteSuccess('Done! Enter <Enter> for continue');

repeat

ch:= readkey;

until ch=#13;

end;

//enter step of integration

Procedure Step;

begin

ClrScr;

Write('Enter number of steps: ');

repeat

Readln(steps);

if steps <= 0 then begin

WriteError('The number steps must be greater than 0! Try again.');

end;

until (steps > 0);

WriteSuccess('Done! Enter <Enter> for continue');

repeat

ch:= readkey;

until ch=#13;

end;

Function Result:real;

var

r:real = 0;

h:real;

i:integer;

begin

ClrScr;

if (a=0) and (b=0) then begin

WriteError('You have not entered the limits of integration! Enter <Enter> for continue');

end

else

begin

if steps <= 0 then begin

WriteError('You did not set the number of steps! Enter <Enter> for continue');

end

else begin

h:=(b-a)/steps;

r:=Func(b);

for i:=1 to steps-1 do

if ((a+i\*h)>t1) and ((a+i\*h)<t3) then r:=r+Func(a+i\*h)

else if (a+i\*h)>t2 then r:=r+Func(a+i\*h)

else r:=0;

r:=r\*h;

Exit(r);

end;

end;

Exit(-1);

repeat

ch:=readkey;

until ch=#13;

end;

procedure ShowResult;

var r:real;

var h:real;

begin

r:=Result;

if r >= 0 then begin

h:=(b-a)/steps;

write('The area on the segment [',a:5:2,';',b:5:2,'] with a step ',h:5:2,' is equal to ');

writeln(r:5:2);

WriteSuccess('Enter <Enter> for continue.');

end;

repeat

ch:= readkey;

until ch=#13;

end;

procedure AbsAcc;

var r,l,z,h:real;

begin

ClrScr;

r:=Result;

h:=(b-a)/steps;

if r >= 0 then begin

if (b<t1) then l:=0;

if (b>t1) and (b<t3) then begin

if (a<=t1) then begin

a:=t2+h;

l:=FFunc(b)- FFunc(a);

end;

if (a>t1) then l:=FFunc(b)- FFunc(a);

end;

if (b>t3) and (b<t2) then begin

if (a<=t2) then begin

a:=t2+h;

l:=FFunc(b) - FFunc(a);

end;

if (a>t2) and (a<t3) then l:=FFunc(b)-FFunc(a);

if (a>t3) then begin

a:=t3+h;

l:=FFunc(b)-FFunc(a);

end;

end;

if (b>t1) then begin

if (a<t2) then begin

a:=t2+h;

l:=FFunc(b) - FFunc(a);

z:=FFunc(t2) -FFunc(t3);

l:=l-z;

end;

if (a>t2) and (a<t3) then begin

l:=FFunc(b)-FFunc(a);

z:=FFunc(t2)-FFunc(t3);

l:=l-z;

end;

if (a>t3) and (a<t1) then begin

a:=t3;

l:=FFunc(b)-FFunc(a);

z:=FFunc(t2)-FFunc(t3);

l:=l-z;

end;

end;

writeln(abs(l-r):10:5);

WriteSuccess('Enter <Enter> for continue.');

end;

repeat

ch:= readkey;

until ch=#13;

end;

procedure RelAcc;

var r,l,h,z:real;

begin

ClrScr;

r:=Result;

h:=(b-a)/steps;

if r >= 0 then begin

if (b<t1) then l:=0;

if (b>t1) and (b<t3) then begin

if (a<=t1) then begin

a:=t2+h;

l:=FFunc(b)- FFunc(a);

end;

if (a>t1) then l:=FFunc(b)- FFunc(a);

end;

if (b>t3) and (b<t2) then begin

if (a<=t2) then begin

a:=t2+h;

l:=FFunc(b) - FFunc(a);

end;

if (a>t2) and (a<t3) then l:=FFunc(b)-FFunc(a);

if (a>t3) then begin

a:=t3+h;

l:=FFunc(b)-FFunc(a);

end;

end;

if (b>t1) then begin

if (a<t2) then begin

a:=t2+h;

l:=FFunc(b) - FFunc(a);

z:=FFunc(t2) -FFunc(t3);

l:=l-z;

end;

if (a>t2) and (a<t3) then begin

l:=FFunc(b)-FFunc(a);

z:=FFunc(t2)-FFunc(t3);

l:=l-z;

end;

if (a>t3) and (a<t1) then begin

a:=t3;

l:=FFunc(b)-FFunc(a);

z:=FFunc(t2)-FFunc(t3);

l:=l-z;

end;

end;

writeln(abs(l-r)/abs(l):10:5);

WriteSuccess('Enter <Enter> for continue.');

end;

repeat

ch:= readkey;

until ch=#13;

end;

Procedure MenuToScr;

var i:integer;

begin

ClrScr;

for i:=1 to N do

begin

GoToXY(x,y+i-1);

write(menu[i]);

end;

writeln();

writeln();

writeln('[',a:5:2,';',b:5:2,'], number of steps: ',steps);

TextAttr:=SEL;

GoToXY(x,y+punkt-1);

write(menu[punkt]);

TextAttr:=NORM;

end;

Procedure ChangeMenu(d:boolean);

begin

GoToXY(x,y+punkt-1);

write(menu[punkt]);

if d then punkt:=punkt+1

else punkt:=punkt-1;

TextAttr:=SEl;

GoToXY(x,y+punkt-1);

write(menu[punkt]);

TextAttr:=NORM;

end;

|  |
| --- |
| begin |
|  |

|  |
| --- |
| menu[1]:='Information on the program'; |
|  |

|  |
| --- |
| menu[2]:='Enter limit of integration'; |
|  |

|  |
| --- |
| menu[3]:='Enter the number of steps'; |
|  |

|  |
| --- |
| menu[4]:='Result'; |
|  |

|  |
| --- |
| menu[5]:='Absolute accuracy'; |
|  |

|  |
| --- |
| menu[6]:='Relative accuracy'; |
|  |

|  |
| --- |
| menu[7]:='Exit'; |
|  |

|  |
| --- |
| punkt:=1; |
|  |

|  |
| --- |
| x:=1; |
|  |

|  |
| --- |
| y:=1; |
|  |

|  |
| --- |
| TextAttr:=NORM; |
|  |

|  |
| --- |
| MenuToScr; |
|  |

|  |
| --- |
| repeat |
|  |

|  |
| --- |
| ch:=ReadKey; |
|  |

|  |
| --- |
| if ch=#0 then begin |
|  |

|  |
| --- |
| ch:=ReadKey; |
|  |

|  |
| --- |
| case ch of |
|  |

|  |
| --- |
| #80: |
|  |

|  |
| --- |
| if punkt<N then |
|  |

|  |
| --- |
| begin |
|  |

|  |
| --- |
| ChangeMenu(true); |
|  |

|  |
| --- |
| end; |
|  |

|  |
| --- |
| #72: |
|  |

|  |
| --- |
| if punkt>1 then |
|  |

|  |
| --- |
| begin |
|  |

|  |
| --- |
| ChangeMenu(false); |
|  |

|  |
| --- |
| end; |
|  |

|  |
| --- |
| end; |
|  |

|  |
| --- |
| end |
|  |

|  |
| --- |
| else if ch=#13 then begin |
|  |

|  |
| --- |
| case punkt of |
|  |

|  |
| --- |
| 1:Information; |
|  |

|  |
| --- |
| 2:Limit; |
|  |

|  |
| --- |
| 3:Step; |
|  |

|  |
| --- |
| 4:ShowResult; |
|  |

|  |
| --- |
| 5:AbsAcc; |
|  |

|  |
| --- |
| 6:RelAcc; |
|  |

|  |
| --- |
| 7:ch:=#27; |
|  |

|  |
| --- |
| end; |
|  |

|  |
| --- |
| MenuToScr; |
|  |

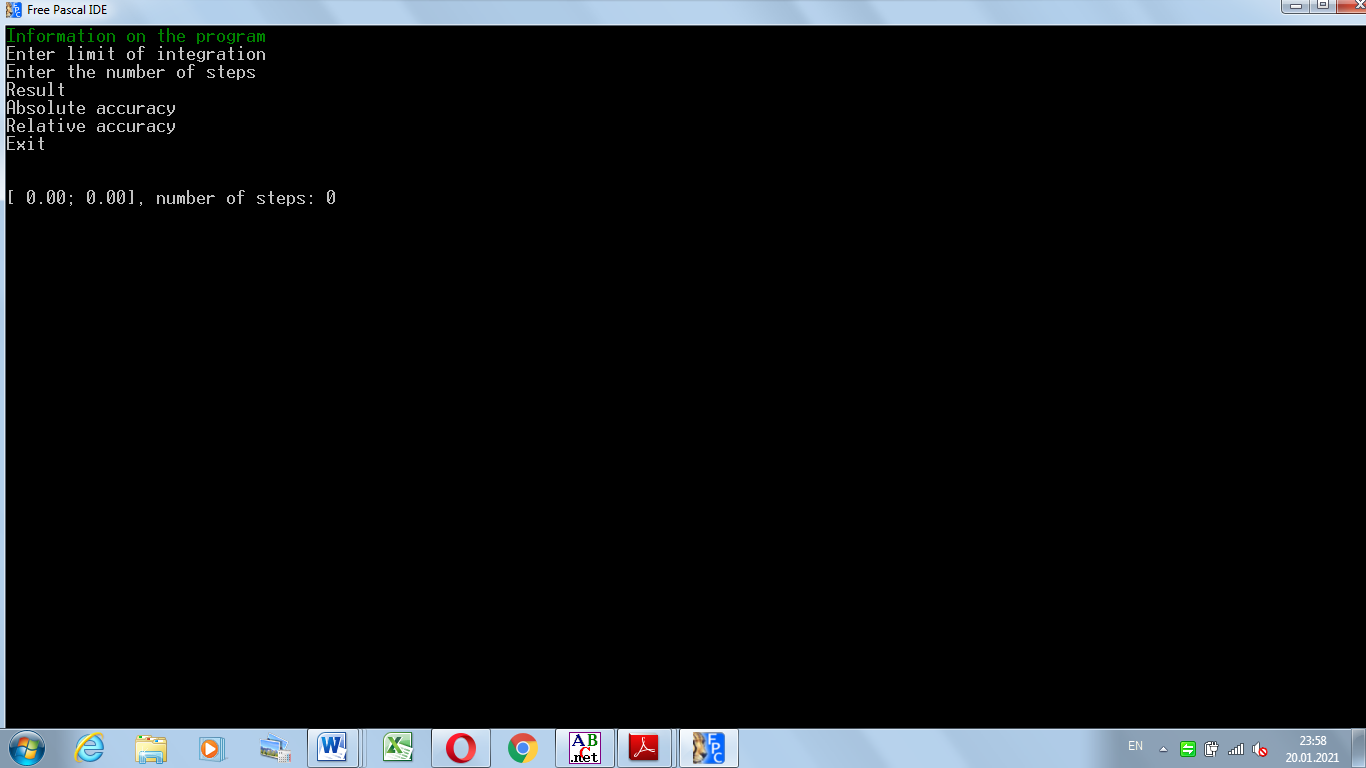
|  |
| --- |
| end; |
|  |

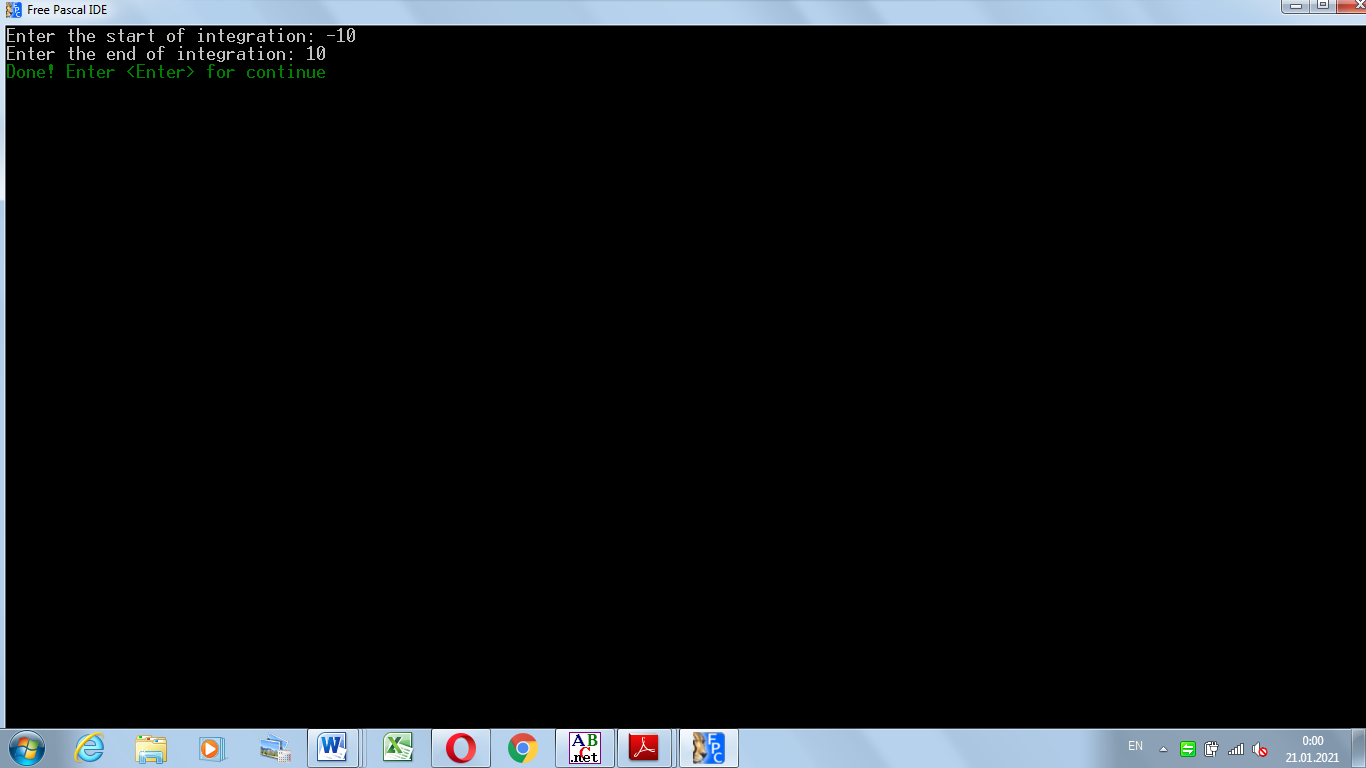
|  |
| --- |
| until ch=#27; |
|  |

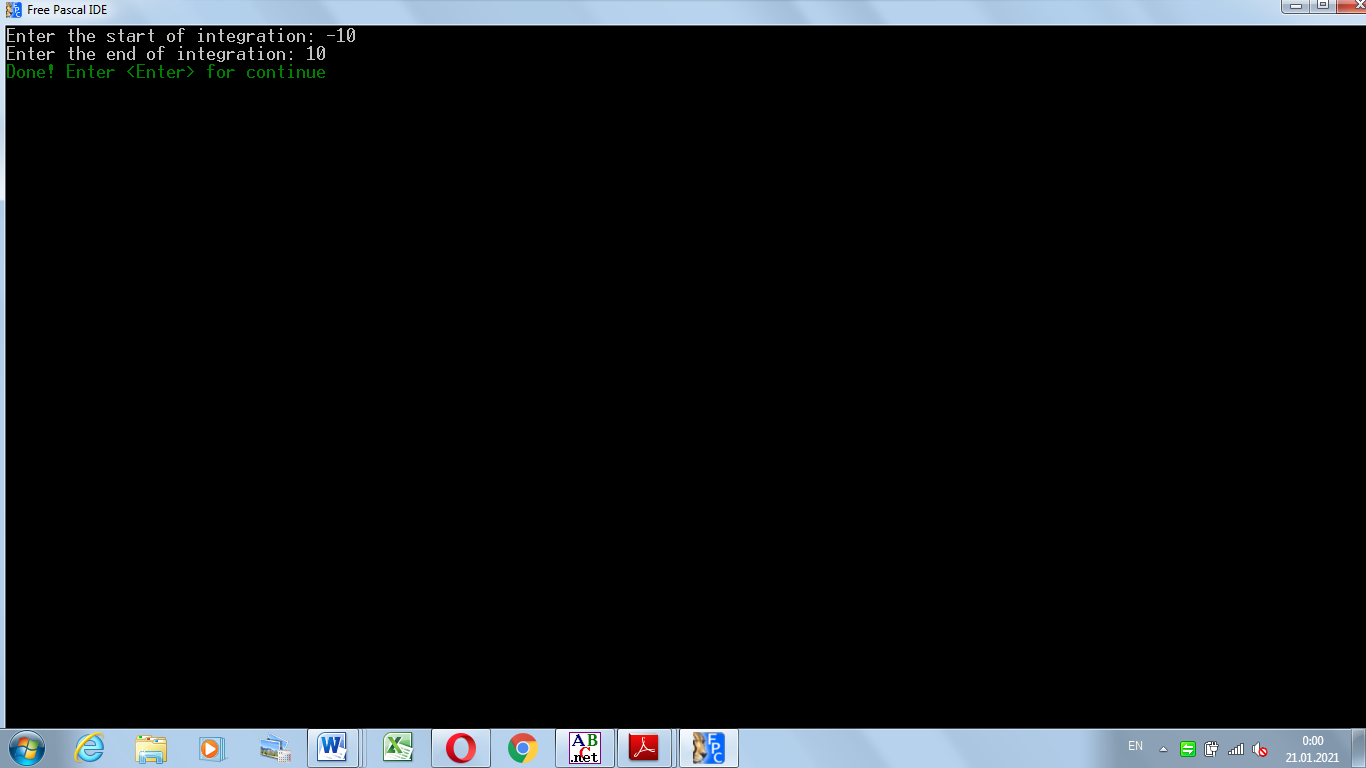
|  |
| --- |
| ClrScr; |
|  |

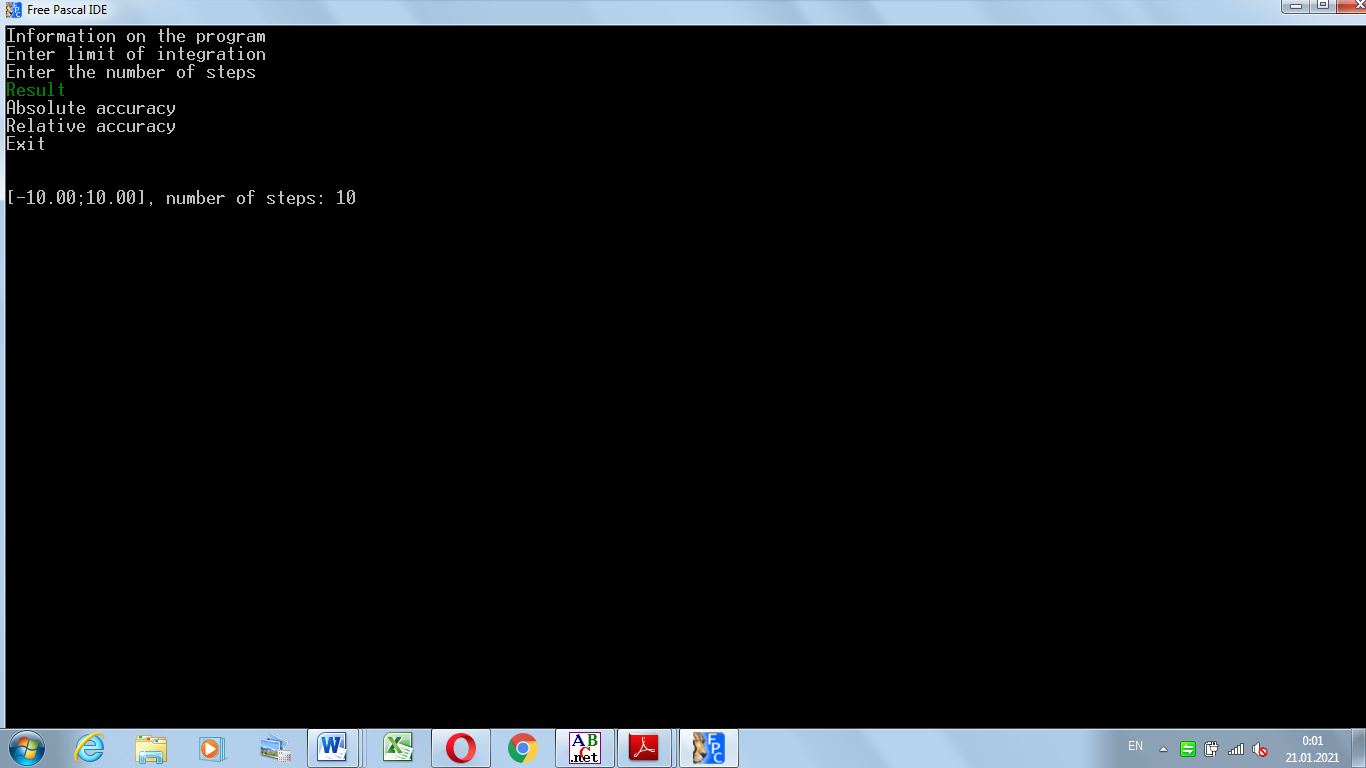
end.

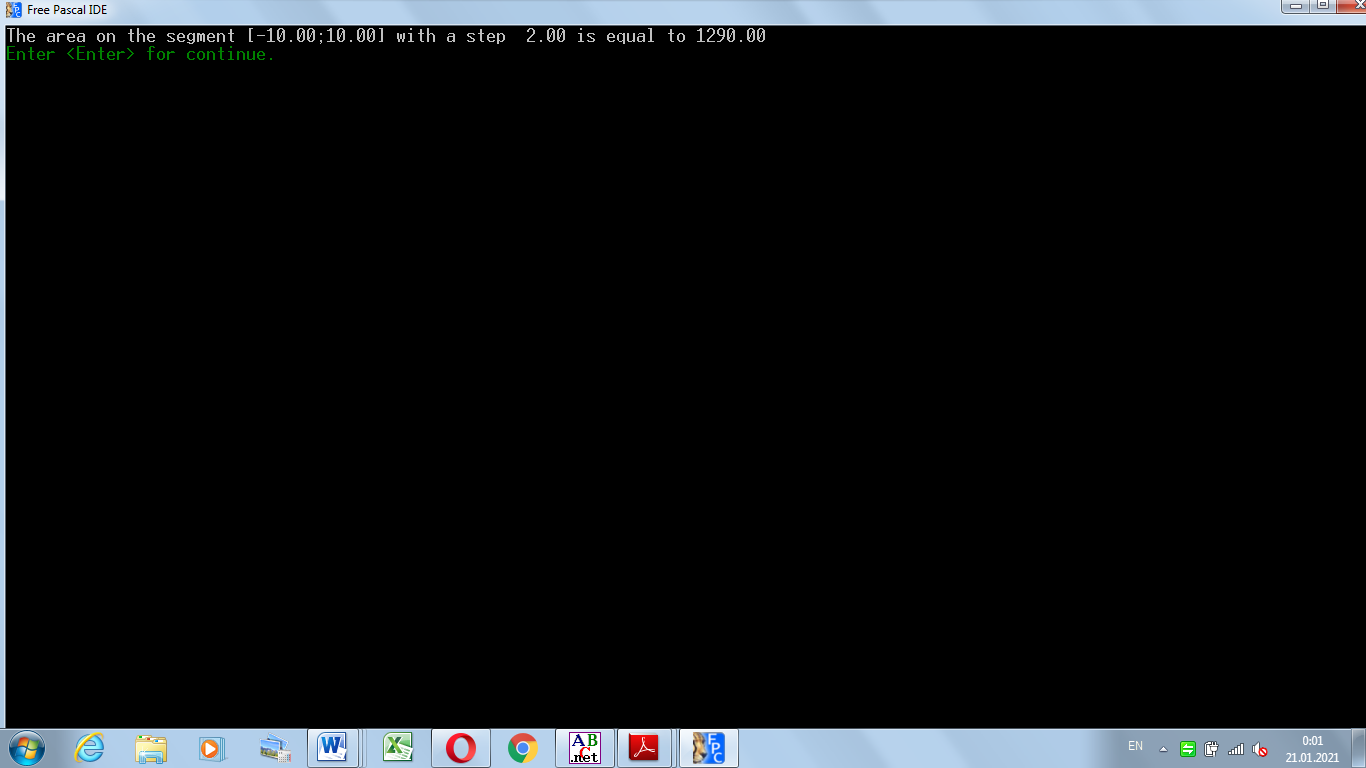
**Экранная форма**











**Вывод:**  В ходе данной лабораторной работы были освоены навыки построения процедур и функций, познакомились понятиями передачи аргументов по значению. Познакомились с понятием видимости переменных: глобальные и локальные переменные. Научились строить минимальный пользовательский интерфейс в стиле case-меню. Узнали как оформлять в схеме алгоритма процедуры и функции.