МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

**«Вятский государственный университет»**

**(ФГБОУ ВО «ВятГУ»)**

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Вычисление значения функции

Отчет

Лабораторная работа №4 по дисциплине

«Программирование»

Выполнил студент группы ИВТ-11 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Кудяшев Я.Ю./

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

**Цель работы**

Освоить принципы работы в графическом режиме, получить базовые навыки взаимодействия с графическими примитивами.

**Задание**

1. Дополнить программу, реализованную в ходе предыдущей лабораторной работы, режимом визуализации.
2. Предусмотреть возможность вывода кривой, ограничивающей фигуру, на координатную плоскость.
3. Реализовать следующие возможности и элементы: масштабирование графика, подписи на осях, вывод информации о задании.
4. Реализовать не менее двух возможностей из представленных: независимое масштабирование по осям, штриховка вычисляемой площади, визуализация численного расчёта интеграла.

**Программа**

**program** lab4;

**uses** wincrt, graph, crt;

**const** norm = lightgray;

kk = -1.73;

**var** menu:**array** [1..5] **of** string;

punkt,x,y:integer;

ch:char;

i:integer;

n:char;

h,xb,t,s,kek,shrek:extended;

v:char;

a,b,o,c,temp:real;

**function** f(x:real): real;

**begin**

f:= 2\*(x\*x\*x)+(-1)\*(x\*x)+(-3)\*x+(5);

**end**;

**function** f1(x:real):real;

**begin**

f1 := x\*(3\*x\*x\*x-2\*x\*x-9\*x+30)/6;

**end**;

**procedure** area(i:integer);

**begin**

**repeat**

clrscr;

writeLn('Enter a lower integration limit between -30 and 30');

readln(a);

**until** (a >= -30) **and** (a <= 30);

**repeat**

clrscr;

writeLn('Enter the upper limit of integration between -30 and 30');

readln(b);

**until** (b>=-30) **and** (b<=30);

**if** a>b **then**

**begin**

temp:=a;

a:=b ;

b:=temp;

**end**;

**if** a<kk **then** a:=kk;

**if** b<kk **then** b:=kk;

**if** a=b **then**

**begin**

writeLn('Square is 0');

readLn;

**end**

**else**

**begin**

**repeat**

clrscr;

writeLn('Enter the number of rectangles enter the number of rectangles between 1 and 10000');

readln(c);

writeLn('Press "Enter" to continue');

**until** (c>=1) **and** (c<=10000);

**begin**

h:=(b-a)/(c);

s:=0;

xb:=a;

i:=0;

**repeat**

**begin**

t:=xb+i\*h;

s:=s+f(t)\*h;

i:= i+1;

**end**;

**until** i>c;

readLn;

**end**;

**end**;

**end**;

**procedure** inaccuracy;

**begin**

clrscr;

writeln('The area of the figure = ',abs(s):12:10);

writeLn;

o:= f1(b)-f1(a);

kek:= abs(o-s);

writeLn('Absolute inaccuracy = ', kek:0:5);

shrek:=abs(kek/s\*100);

writeLn('Comparative inaccuracy = ',shrek:0:5,'%');

writeLn('Press "Enter" to continue');

readLn;

**end**;

**procedure** information;

**begin**

clrscr;

writeLn('Function: 2\*x^3+(-1)\*x^2+(-3)\*x+5');

writeLn;

writeLn('The method of computation of square: left triangle method');

writeLn;

writeLn('The intersection point of the graph with the x-axis is -1.73');

writeLn;

writeLn('Press "Enter" to continue');

readLn;

**end**;

**procedure** graphik;

**var**

p,gd,gm,x,i,j,z:integer;

l,oo:real;

c:longint;

s:string;

ch2:char;

**begin**

z:=3;

c:=5000;

gd:=Detect;

gm:=GetMaxMode;

initgraph(gd,gm,'');

**repeat**

setcolor(2);

line(400+500,0,400+500,800);

line(0+500,400,800+500,400);

OuttextXY(getmaxx-1200,10,'2\*x^3+(-1)\*x^2+(-3)\*x+5');

OuttextXY(getmaxx-1200,25,'Scaling');

OuttextXY(getmaxx-1200,40,'Press "+" for +');

OuttextXY(getmaxx-1200,55,'Press "-" for -');

OuttextXY(getmaxx-1200,70,'Press D for +OX');

OuttextXY(getmaxx-1200,85,'Press A for -OX');

OuttextXY(getmaxx-1200,100,'Press W for +OY');

OuttextXY(getmaxx-1200,115,'Press S for -OY');

OuttextXY(getmaxx-1200,135,'The metod of left rectangles');

OuttextXY(getmaxx-1200,155,'Exit "ESC"');

x:=0;

**for** i:=0 **to** 16 **do**

**begin**

line(x+500,380,x+500,420);

str(i\*z-8\*z,s);

OutTextXY(x+500,380,s);

line(880,x,920,x);

str(8\*c-i\*c,s);

OutTextXY(900,x+10,s);

x:=x+50;

**end**;

p:=0;

oo:=c/50;

**while** p<=800 **do**

**begin**

l:=-(f(((p-400)\*z/50)))/oo+400;

PutPixel(p+500,round(l),9);

setcolor(3);

**if** (p>((a\*50)/z+400)) **and** (p<(b\*50)/z+400) **and** (round(l)<=400) **then**

line(p+500,round(l),p+500,400);

p:=p+1;

**end**;

ch2:=wincrt.readkey;

**case** ch2 **of**

#119:

**if** c<10000 **then**

**begin**

c:=c+1000;

cleardevice;

**end**;

#115:

**if** c>1000 **then**

**begin**

c:=c-1000;

cleardevice;

**end**;

#97:

**if** z>1 **then**

**begin**

z:=z-1;

cleardevice;

**end**;

#100:

**if** z<4 **then**

**begin**

z:=z+1;

cleardevice;

**end**;

#45:

**if** (z>1) **and** (c>1000) **then**

**begin**

z:=z-1; c:=c-1000;

cleardevice;

**end**;

#61:

**if** (z<4) **and** (c<10000) **then**

**begin**

z:=z+1; c:=c+1000;

cleardevice;

**end**;

**end**;

**until** (ch2=#27);

closegraph;

**end**;

**procedure** menutoscr;

**var** i:integer;

**begin**

clrscr;

**for** i:=1 **to** 5 **do**

**begin**

gotoxy(x,y+i-1);

write(menu[i]);

**end**;

textattr := red;;

gotoxy(x,y+punkt-1);

write(menu[punkt]);

textattr:=norm;

**end**;

**begin**

menu[1] := 'Data input';

menu[2] := 'Result of computation';

menu[3] := 'Information';

menu[4] := 'Graph of function';

menu[5] := 'Exit';

punkt := 1;

x := 5;

y := 5;

textattr := norm;

menutoscr;

**repeat**

ch := readkey;

**if** ch=#0 **then**

**begin**

ch := readkey;

**case** ch **of**

#80:

**if** punkt<5 **then**

**begin**

gotoxy(x,y+punkt-1);

write(menu[punkt]);

punkt:= punkt+1;

textattr := red;

gotoxy(x,y+punkt-1);

write(menu[punkt]);

textattr:=norm;

**end**;

#72:

**if** punkt>1 **then**

**begin**

gotoxy(x,y+punkt-1);

write(menu[punkt]);

punkt := punkt - 1;

textattr := red;

gotoxy(x,y+punkt-1);

write(menu[punkt]);

textattr:= norm;

**end**;

**end**;

**end**

**else**

**if** ch=#13 **then**

**begin**

**case** punkt **of**

1:area(i);

2:inaccuracy;

3:information;

4:graphik;

5:ch:=#27;

**end**;

menutoscr;

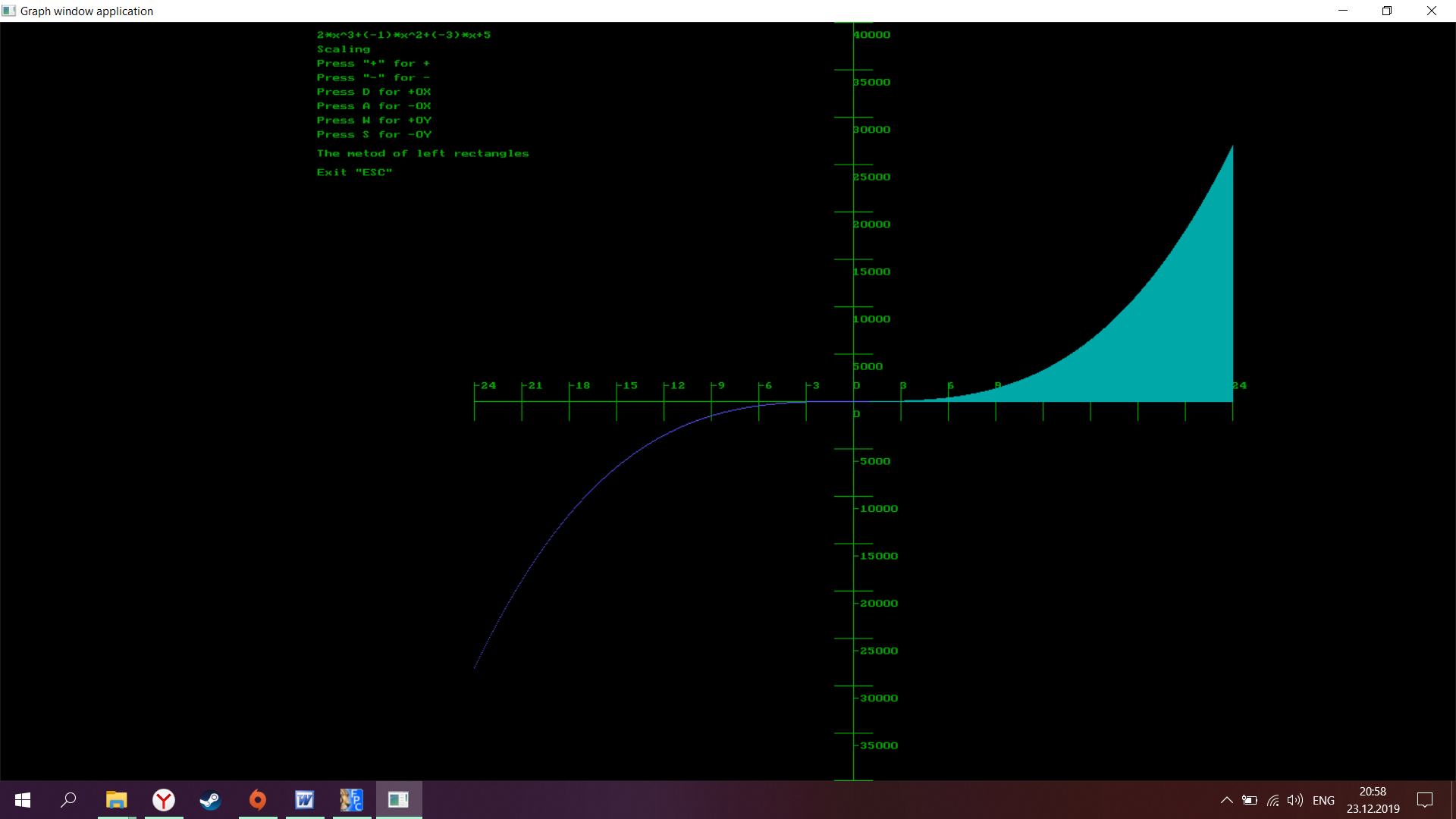
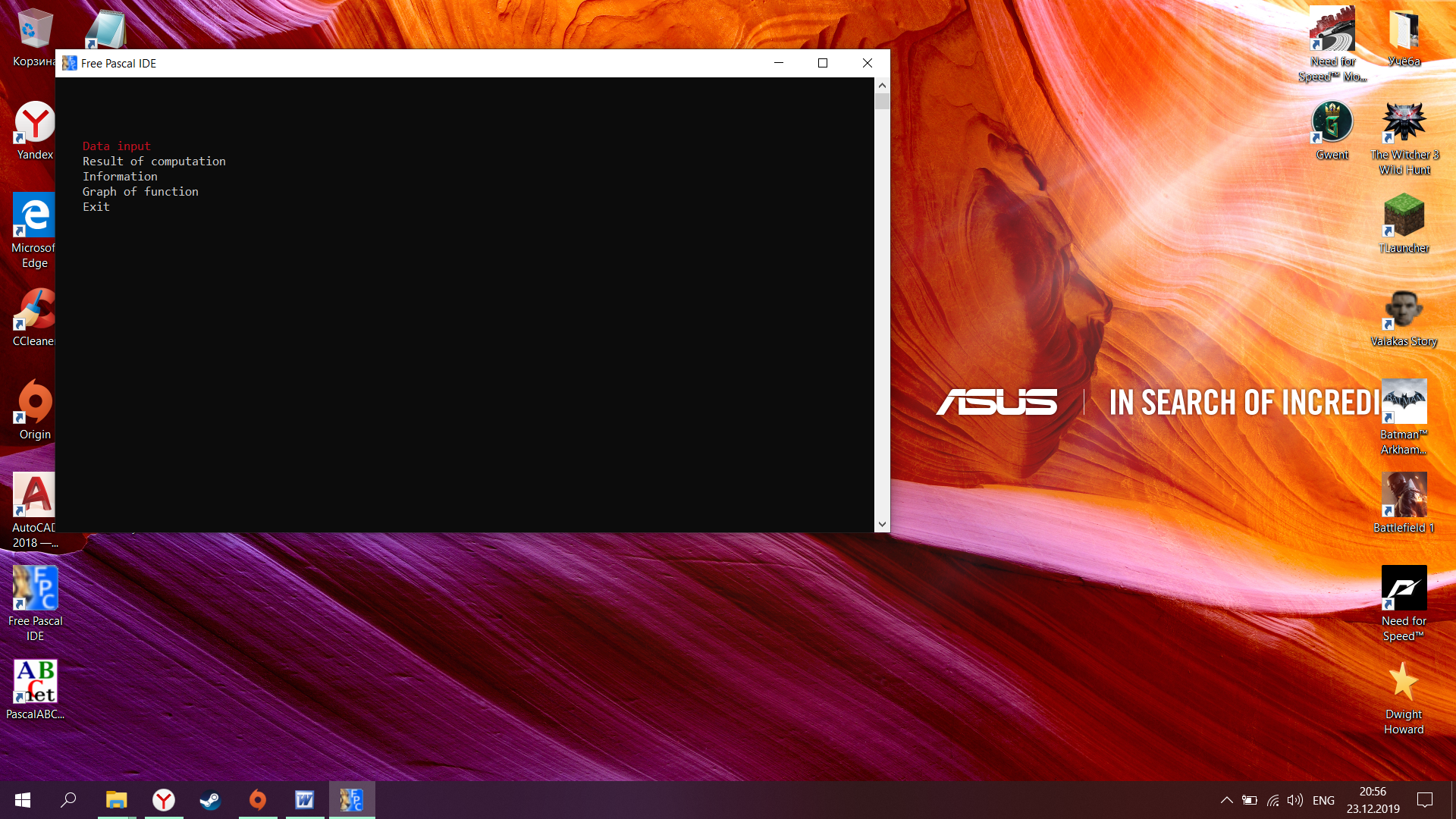
**end**;

**until** ch=#27;

**end**.

**Схема**

**Скриншоты программы**



**Вывод**

Язык программирования **Pascal** – это базовая высокоуровневая программа. Она используется как основа для программирования в большинстве современных школ. Такой востребованностью среди учебных заведений данный язык достиг благодаря своей простоте. Любая программа состоит из двух последовательных частей: описание последовательных действий, которое необходимо выполнить, и описание данных, которыми оперируют эти действия.

Для реализации визуализации графика функции был необходим модуль **graph**. Модуль **graph** представляет собой библиотеку подпрограмм **Pascal**, обеспечивающих полное управление графическими режимами. Библиотека содержит более 50 графических процедур и функций, как базовых (рисование точек, линий, окружностей), так и расширяющих возможностей базовых (многоугольники, закрашивание фигур, вывод текста и др.).

В ходе выполнения данной лабораторной работы **case-меню** из предыдущей лабораторной работы претерпело небольшие изменения – появился новый пункт «**Graph of function»**, в котором и была реализована возможность визуализации графика функции.

Во время реализации программы не обошлось и без подключения модуля **wincrt**. Главной особенностью данного модуля является то, что не нужно писать «специфический» для Windows исходный код. Модуль **wincrt** реализует аналогичный терминалу текстовый экран в окне. С помощью модуля **wincrt** можно легко создавать программы, использующие стандартные процедуры Read, ReadLn, Write, WriteLn для операций вода и вывода.

При помощи модуля **crt** была реализована возможность смены цвета экрана и его очистка. Данный модуль даёт пользователю доступ к набору различных цветов, каждый из которых имеет свой номер.

Подводя итоги, можно сказать, что в ходе выполнения лабораторной работы №4 были освоены принципы работы с модулем **graph**. Программа из лабораторной работы №3 была усовершенствована – появилась возможность наглядного просмотра графика задаваемой функции. Также была реализована возможность свободного масштабирования графика по осям OX и OY и заливка вычисляемой площади.