**Геометрическое описание метода хорд**

Будем искать нуль функции A description.... Выберем две начальные точки A description...(A description...;A description...) и A description...(A description...;A description...) и проведем через них прямую. Она пересечет ось абсцисс в точке (A description...;0). Теперь найдем значение функции с абсциссой A description.... Временно будем считать A description... корнем на отрезке [A description...;A description...]. Пусть точка A description... имеет абсцисcу A description... и лежит на графике. Теперь вместо точек A description... и A description... мы возьмём точку A description... и точку A description.... Теперь с этими двумя точками проделаем ту же операцию и так далее, то есть будем получать две точки A description... и A description... и повторять операцию с ними. Отрезок, соединяющий последние две точки, пересекает ось абсцисс в точке, значение абсциссы которой можно приближённо считать корнем. Эти действия нужно повторять до тех пор, пока не получим значение корня с нужным приближением.

**Метод хорд**

Иногда методом секущих называют метод с итерационной формулой

A description....

Этот метод можно считать разновидностью метода простой итерации и он имеет меньшую скорость сходимости. Далее для определённости этот метод будем называть методом хорд, а метод, описанный в предыдущем разделе, методом секущих.

**Задание:** Вычеслить и напечатать значение … нелинейного уравнения по 3-м методам с точностью e=0.001

19 вариант

unit Unit1;

{$mode objfpc}{$H+}

interface

uses

Classes, SysUtils, FileUtil, TAGraph, TASeries, Forms, Controls, Graphics,

Dialogs, StdCtrls, crt;

type

{ TForm1 }

TForm1 = class(TForm)

Button1: TButton;

Button2: TButton;

Chart1: TChart;

Chart1LineSeries1: TLineSeries;

Label1: TLabel;

Label2: TLabel;

Memo1: TMemo;

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

private

{ private declarations }

public

{ public declarations }

end;

var

Form1: TForm1;

c,x,y,z:real;

const e = 0.001;

implementation

{$R \*.lfm}

{ TForm1 }

function f(x:real):real;

begin

f:=sqr(x)-exp(x)-2;

end;

procedure TForm1.Button1Click(Sender: TObject);

var s:string;

code1,code2,code3:integer;

g:real;

begin

s:=InputBox('Введите x','Ввод x','');

val(s,x,code1);

if code1<>0 then

begin

showmessage('неверный ввод x');

exit;

end;

s:=InputBox('Введите y','Ввод y','');

val(s,y,code2);

if code2<>0 then

begin

showmessage('неверный ввод y');

exit;

end;

g:=x;

while g<=y do

begin

Chart1LineSeries1.AddXY(g,f(g),'');

g:=g+0.01;

end;

if f(x)\*f(y)>0 then memo1.lines.Add('Нет корней')

else

begin

repeat

c:=x-(y-x)\*f(x)/(f(y)-f(x));

if f(x)\*f(z)>0 then x:= z

else y := z;

until abs(f(z))<e;

x:=c;

memo1.lines.Add(floattostr(c));

end;

end;

procedure TForm1.Button2Click(Sender: TObject);

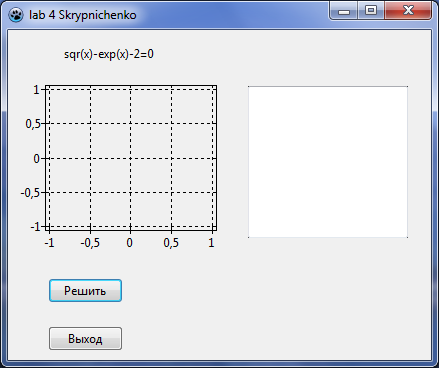
begin

close;

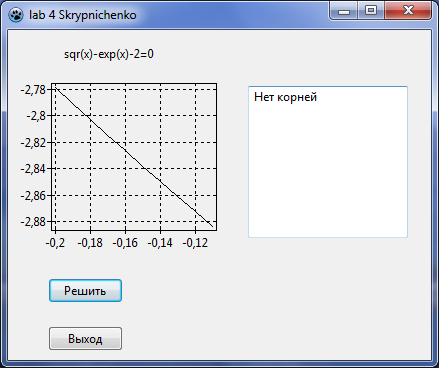
end;

end.

**Внешний вид программы:**



**Вычисление:**



**X: -0.2**

**Y: -0.1**