МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Вятский государственный университет»

Факультет автоматики и вычислительной техники Кафедра электронных вычислительных машин

РЕАЛИЗАЦИЯ АЛГОРИТМОВ РАСТРОВОЙ ГРАФИКИ ДЛЯ ЗАПОЛНЕНИЯ СПЛОШНЫХ ОБЛАСТЕЙ

Отчет

Лабораторной работе №4 по дисциплине

«Компьютерная графика»

Выполнил студент группы ИВТб-21\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Жеребцов К. А. /

Проверил преподаватель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Коржавина А.С./

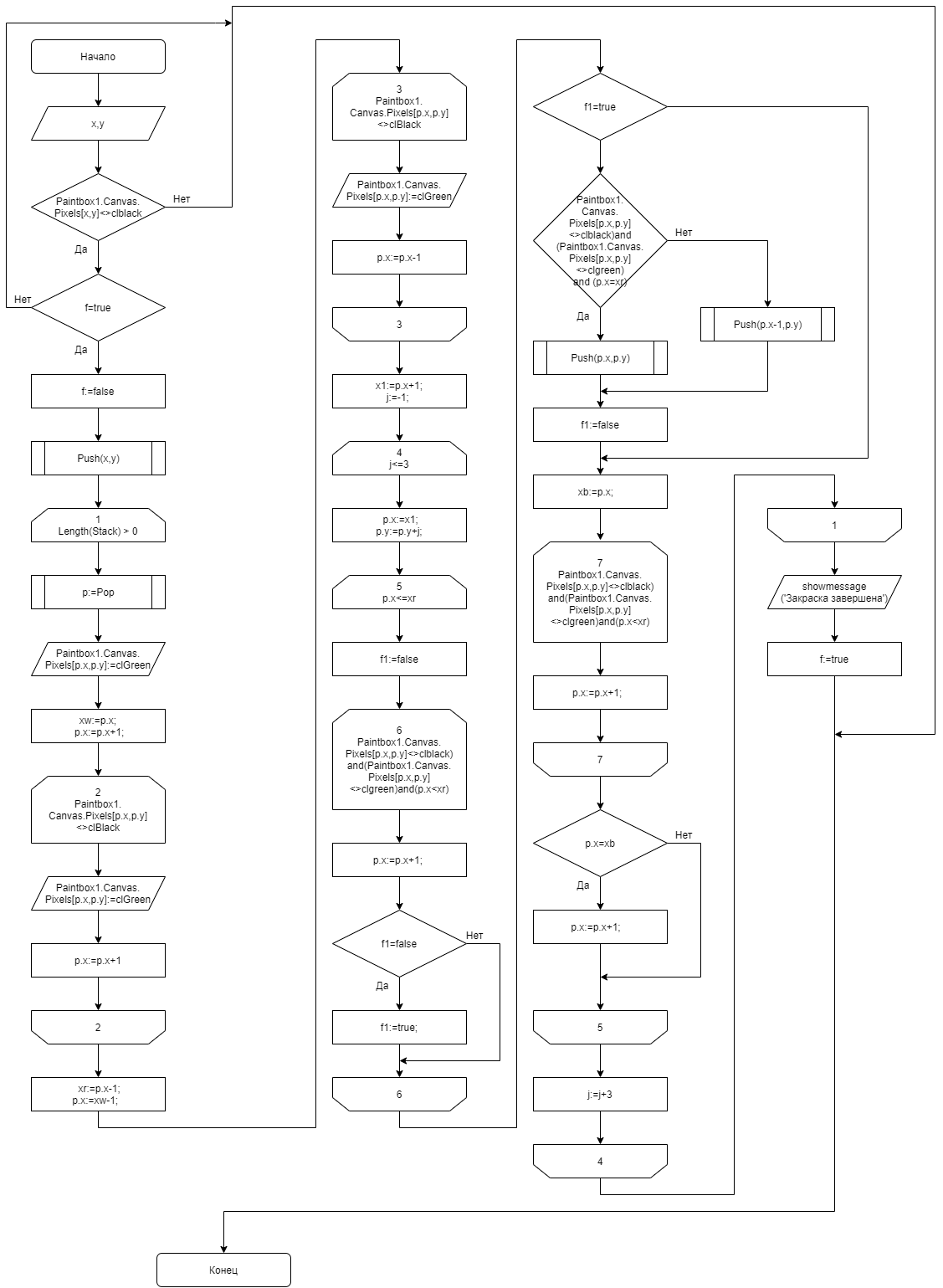
2021 г.

**Цель:** закрепить лекционный материал по изучению базовых алгоритмов компьютерной графики алгоритмов закраски.

**Задание на лабораторную работу:**

1. Написать на языке PASCAL программу, реализующую алгоритм заливки многоугольника любой формы.
2. Проверить правильность работы программы, нарисовав с помощью функций модуля GRAPH невыпуклый многоугольник (с дырами внутри), закрасьте его заданным цветом, указав координаты принадлежащей многоугольнику точки.
3. Написать и отладить программу, реализующую алгоритм построчного заполнения выпуклого многоугольника, заданного координатами вершин и цветом границы (можно воспользоваться алгоритмом заполнения треугольника, прочитанным на лекции).

**Схемы алгоритмов:**



**Исходный код**

unit Unit1;

{$mode objfpc}{$H+}

interface

uses

Classes, SysUtils, Forms, Controls, Graphics, Dialogs, ExtCtrls, StdCtrls;

type

PNode = ^Node; { указатель на узел }

Node = record

data: Tpoint; { полезная }

next: PNode; { указатель на след. элемент }

end;

{ TForm1 }

TForm1 = class(TForm)

Button1: TButton;

PaintBox1: TPaintBox;

Timer1: TTimer;

procedure Button1Click(Sender: TObject);

procedure FormCreate(Sender: TObject);

procedure PaintBox1MouseDown(Sender: TObject; Button: TMouseButton;

Shift: TShiftState; X, Y: Integer);

procedure Push(x0, y0: Integer);

function Pop: TPoint;

procedure Timer1Timer(Sender: TObject);

private

public

end;

const

ColorCount = 18;

krasas : array[1 .. ColorCount] of integer =

(clMaroon, clGreen, clOlive, clGray, clSilver,

clRed, clYellow, clBlue, clAqua, clLtGray, clDkGray, clFuchsia, clLime, clNavy,

clPurple, clTeal, clMenuHighlight, clMoneyGreen);

var

Form1: TForm1;

Stack: array of TPoint;

p: Tpoint;

xw,xr,x1,xb: integer;

j: integer;

f1: boolean;

s: PNode;

len: Integer;

res: TPoint;

f: boolean=false;

implementation

{$R \*.lfm}

{ TForm1 }

procedure Tform1.Push(x0, y0: Integer);

begin

len := Length(Stack);

SetLength(Stack, len+1);

Stack[len].X := x0;

Stack[len].Y := y0;

end;

function Tform1.Pop: TPoint;

begin

len := Length(Stack);

res := Stack[len-1];

SetLength(Stack, len-1);

Pop := res;

end;

procedure TForm1.Timer1Timer(Sender: TObject);

begin

with form1.PaintBox1.Canvas do

begin

Clear;

brush.Color:=clWhite;

fillrect(0,0,450,450);

rectangle(0,0,450,450);

ellipse(230,150,300,200);

Moveto(50,50);

lineto(400,400);

lineto(213,340);

lineto(332,150);

lineto(220,130);

lineto(150,40);

lineto(50,50);

end;

timer1.Enabled:=false;

end;

procedure TForm1.PaintBox1MouseDown(Sender: TObject; Button: TMouseButton;

Shift: TShiftState; X, Y: Integer);

var

col: Tcolor;

begin

randomize;

col:=krasas[random(colorcount)+1];

if Paintbox1.Canvas.Pixels[x,y]<>clblack then

begin

if f=true then begin

f:=false;

Push(x,y);

while Length(Stack) > 0 do

begin

p:=Pop; //извлечь из стека

Paintbox1.Canvas.Pixels[p.x,p.y]:=col;

xw:=p.x;

p.x:=p.x+1;

while (Paintbox1.Canvas.Pixels[p.x,p.y]<>clBlack) and (p.X <= 500) do

begin

Paintbox1.Canvas.Pixels[p.x,p.y]:=col;

p.x:=p.x+1;

end;

xr:=p.x-1;

p.x:=xw-1;

while (Paintbox1.Canvas.Pixels[p.x,p.y]<>clBlack) and (p.x >= 0) do

begin

Paintbox1.Canvas.Pixels[p.x,p.y]:=col;

p.x:=p.x-1;

end;

x1:=p.x+1;

j:=-1;

while (j<=3) do

begin

p.x:=x1;

p.y:=p.y+j;

while (p.x<=xr) do

begin

f1:=false;

while((Paintbox1.Canvas.Pixels[p.x,p.y]<>clblack)and(Paintbox1.Canvas.Pixels[p.x,p.y]<>col)and(p.x<xr)) do

begin

p.x:=p.x+1;

if not f1 then

f1:=true;

end;

if f1=true then begin

if((Paintbox1.Canvas.Pixels[p.x,p.y]<>clblack)and(Paintbox1.Canvas.Pixels[p.x,p.y]<>col)and (p.x=xr)) then

Push(p.x,p.y) //поместить в стек(x,y)

else

Push(p.x-1,p.y);//поместить в стек (x-1,y)

f1:=false;

end;

xb:=p.x;

while((Paintbox1.Canvas.Pixels[p.x,p.y]=clblack)or(Paintbox1.Canvas.Pixels[p.x,p.y]=col)and(p.x<xr)) do

p.x:=p.x+1;

if p.x=xb then

p.x:=p.x+1;

end;

j:=j+3;

end;

end;

f:= true;

end;

end;

end;

procedure TForm1.Button1Click(Sender: TObject);

begin

randomize;

with form1.PaintBox1.Canvas do

begin

Clear;

brush.Color:=clWhite;

fillrect(0,0,500,500);

rectangle(0,0,500,500);

ellipse(230,150,300,200);

Moveto(50,50);

lineto(400,400);

lineto(213,340);

lineto(332,150);

lineto(220,130);

lineto(150,40);

lineto(50,50);

end;

f:=true;

end;

procedure TForm1.FormCreate(Sender: TObject);

begin

randomize;

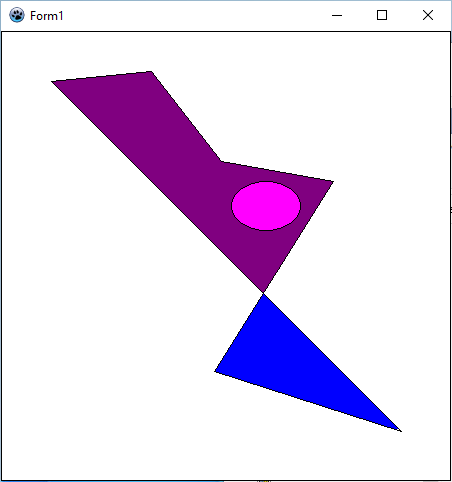
f:=true;

timer1.Enabled:=true;

end;

end.

**Экранные формы**



**Вывод:**

Была написана на языке PASCAL программа, реализующая алгоритм заливки многоугольника любой формы, а также проверена правильность работы программы.