МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Вятский государственный университет»**

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Реализация вывода сплайнов Безье

Отчет по лабораторной работе №5

по дисциплине «Компьютерная графика»

Выполнил студент группы ИВТ-21 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Седов М.Д./

Проверил преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Клюкин В.Л./

Киров 2018

1. Постановка задачи

Написать на языке PASCAL программу, реализующую геометрический алгоритм построения кривой Безъе. Кривая должна строиться пошагово (с задержкой), отображая вспомогательные многоугольники, используемые для получения каждой точки. Закрепить лекционный материал по изучению материала темы "Кривые Безье". Реализовать геометрический алгоритм вывода кривых Безье.

1. Схема алгоритма

На рисунке 1 представлен алгоритм программы построения кривой Безье



Рисунок 1 – алгоритм программы построения кривой Безье

1. Листинг

Листинг программы показан в приложении А.

1. Экранные формы

Экранные формы представлены в приложении Б.

1. Вывод

В ходе выполнения данной лабораторной работы был закреплен материал по изучению темы «Кривые Безье». Реализован геометрический алгоритм вывода сплайнов Безье.

Кроме того данная лабораторная работа помогла закрепить навыки работы с модулем GraphABC. Модуль GraphABC представляет собой простую графическую библиотеку и предназначен для создания несобытийных графических и анимационных программ в процедурном и частично в объектном стиле.

Приложение А

(обязательное)

Листинг программы реализующей геометрический алгоритм построения кривой Безъе

**uses GraphABC;**

**var xn, yn, i, j, x, y, m : integer;**

**t, step : real;**

**P, R : array [1..1000] of Point;**

**procedure DrawRect(x1, y1, x2, y2 : real);**

**var e, x, y, dx, dy, s1, s2 : real;**

**R, G, B : byte;**

**l : boolean;**

**begin**

**x := x1;**

**y := y1;**

**dx := abs(x2- x1);**

**dy := abs(y2 - y1);**

**s1 := sign(x2 - x1);**

**s2 := sign(y2 - y1);**

**if (dy > dx) then begin l := true; swap(dx, dy); end else l := false;**

**e := 2 \* dy - dx;**

**for var i := 1 to round(dx) do begin**

**SetPixel(round(x), round(y), clblack);**

**while (e >= 0) do begin**

**if (l = true) then x := x + s1 else y := y + s2;**

**e := e - 2 \* dx;**

**end;**

**if (l = true) then y := y + s2 else x := x + s1;**

**e := e + 2 \* dy;**

**end;**

**SetPixel(round(x), round(y), clblack);**

**end;**

**begin**

**readln(m);**

**for i := 1 to m do begin**

**readln(xn, yn);**

**P[i].X := xn; P[i].Y := yn;**

**end;**

**x := P[1].X; y := P[1].Y; t := 0; step := 0.01;**

**repeat**

**for i := 1 to m do R[i] := P[i];**

**for j := m downto 2 do**

**for i := 1 to j - 1 do begin**

**R[i].X := R[i].X + round(t \* (R[i + 1].X - R[i].X));**

**R[i].Y := R[i].Y + round(t \* (R[i + 1].Y - R[i].Y));**

**end;**

**DrawRect(x, y, R[1].X, R[1].Y);**

**t := t + step; x := R[1].X; y := R[1].Y;**

**until t > 1;**

**end.**

Приложение Б

Экранные формы



Рисунок 2 – Экранная форма программы реализующей геометрический алгоритм построения кривой Безъе