

Моделирование в среде PASCALABC.

Построение графиков функций, заданных в полярной системе координат

Выполнил работу:

ученик 7А класса МАОУ Лицей 36

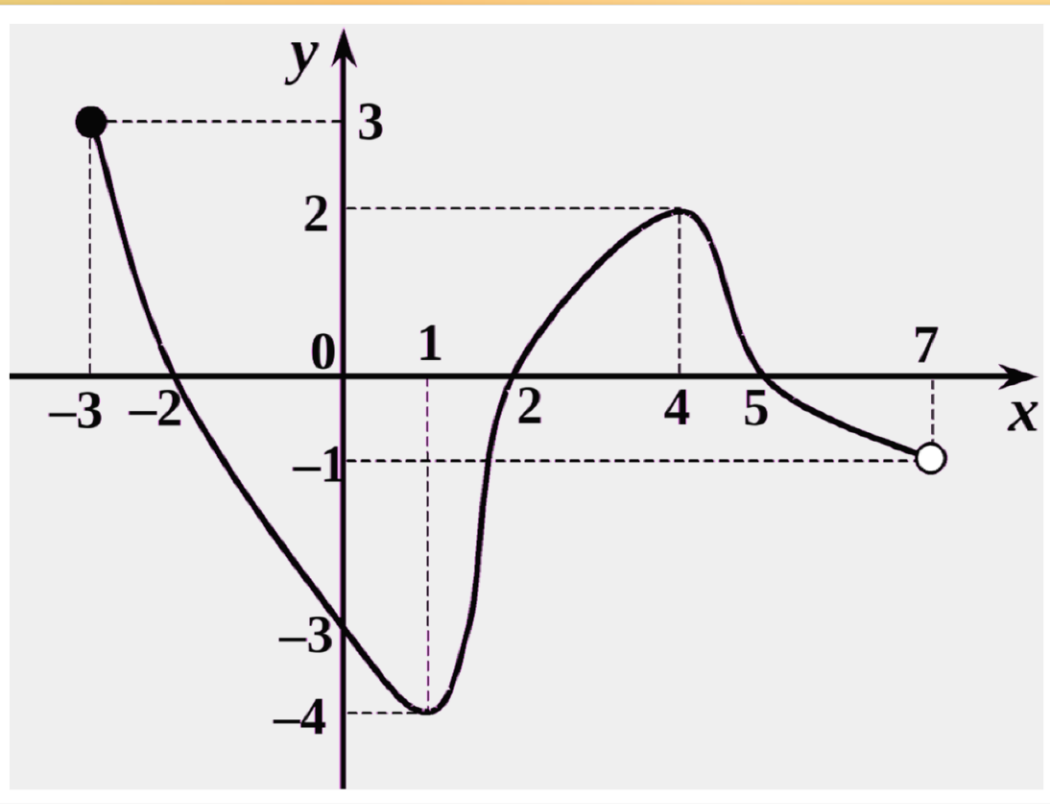
Моисеев Даниил

Руководитель:

Сесорова Лариса Николаевна

Введение

Функция — правило, отражающее взаимно-однозначное соответствие между переменными.



$$y = F(x)$$

$$F(1) = -4$$

$$F(4) = 2$$

$$F(-3) = 3$$

Цель данной работы:

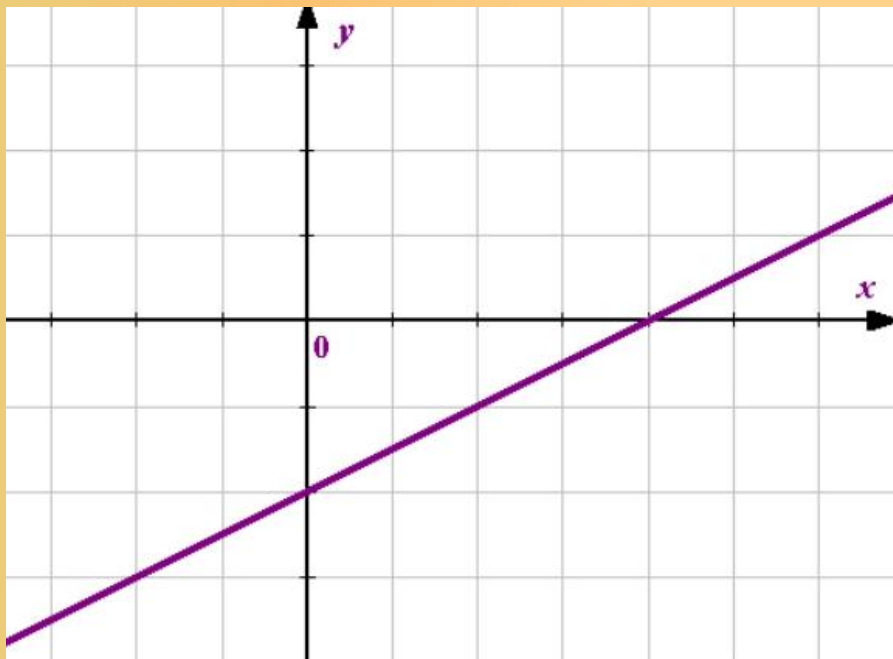
В данной работе мы рассмотрим основные (простые) виды функций, построим графики функций в полярной системе координат. Создадим свой модуль, с процедурами построения графиков функций, обращение к которому будет идти из основной программы.

Виды графиков функций:

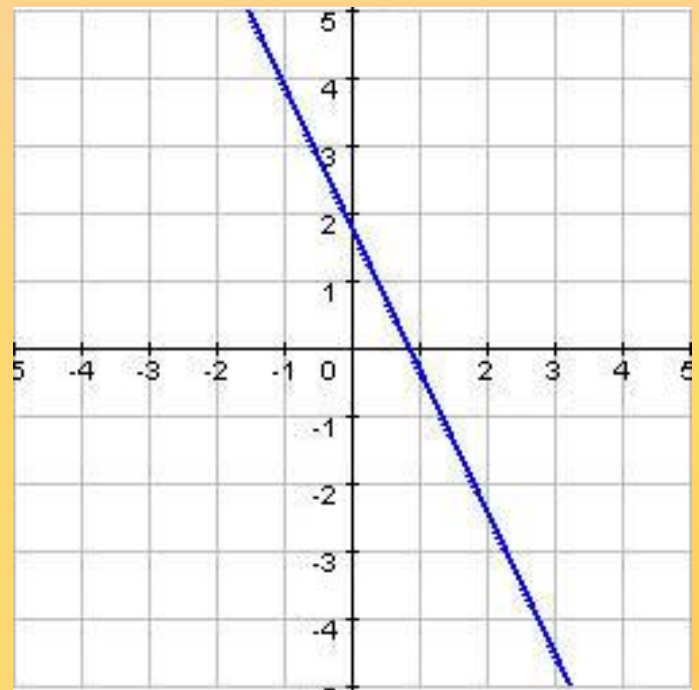
(алгебра 7 класс)

Линейная:

$$y = ax + b$$



$$y = \frac{1}{2} * x - 2$$



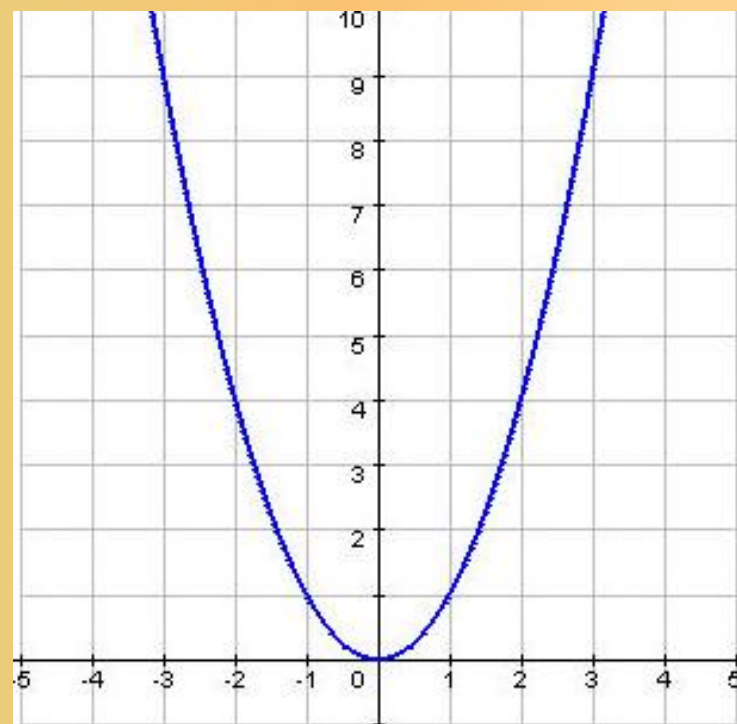
$$y = -2x + 2$$

Виды графиков функций:

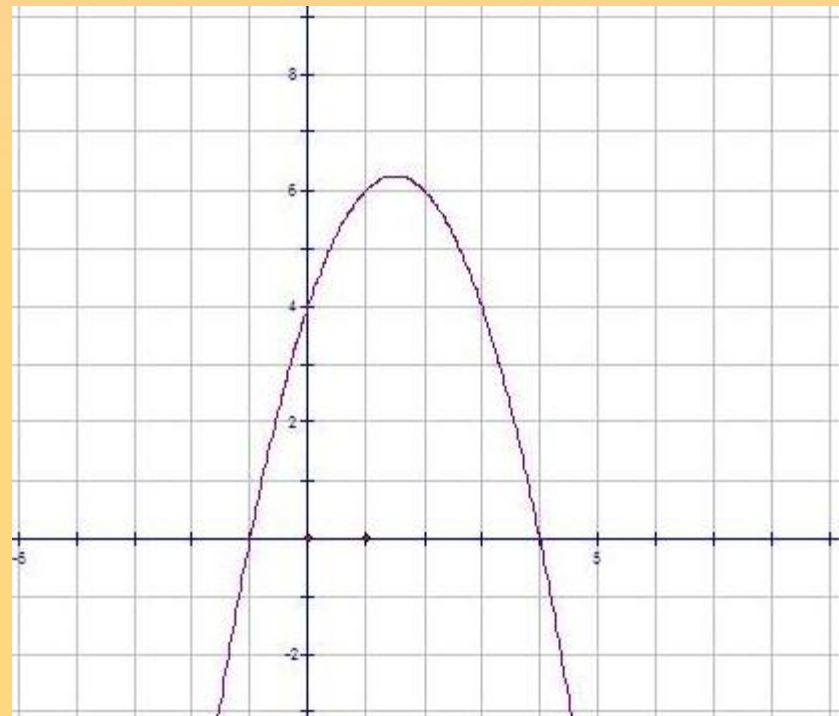
(алгебра 7 класс)

Квадратичная:
парабола

$$y = ax^2 + bx + c$$



$$y = x^2$$



$$y = -x^2 + 3x + 4$$

Структура работы

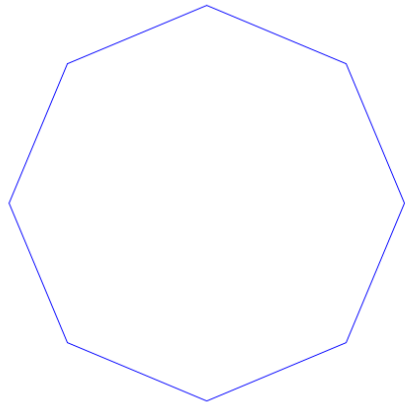
Эти графики строим с помощью циклического алгоритма (в цикле изменяя аргумент функции X , вычисляем $Y(x)$ и строим график по точкам.

```
...readln (a,b,c,m,n);  
x:=M;  
While x<=N do  
  begin  
    y:=a*x*x+b*x+c;  
    Putpixel (round(xc+m*x),round(yc-m*y));  
    x:=x+0.1;  
  end;  
...
```

Графики функций в полярной системе координат:

Многогранник

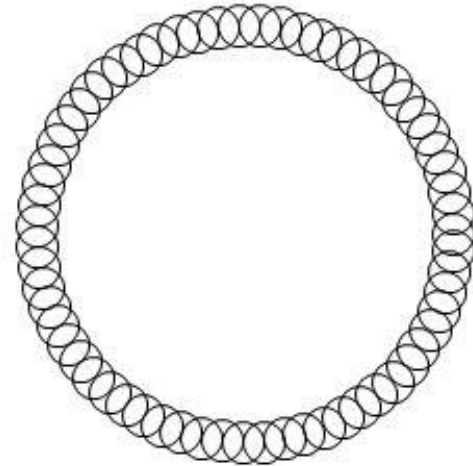
Введите центр:
Введите сколько углов:
Введите радиус:



```
x := x0 + round(r * cos(i * phi));  
y := y0 - round(r * sin(i * phi));
```

```
phi := 2 * pi / n;
```

Круги по кругу

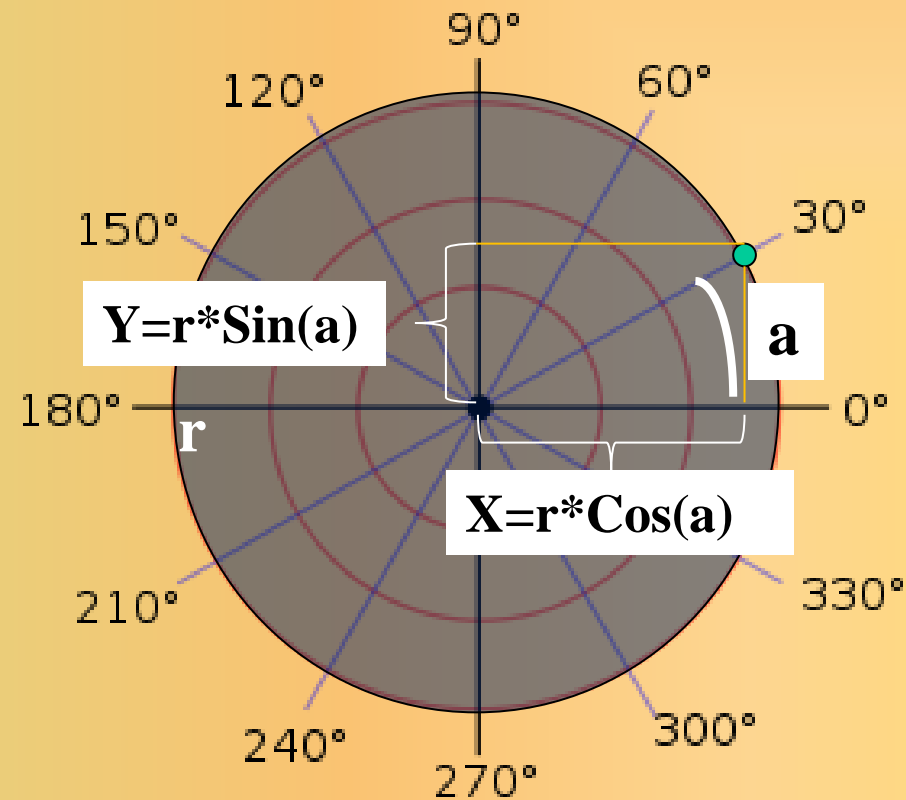


```
x:=xc+round(r*cos(a));  
y:=yc-round(r*sin(a));
```

```
a € [0; 2*Pi]
```

```
a+=0.1;
```

График функции в полярной системе координат



Построение графика функции, заданной в полярной системе координат: в цикле изменяем $a \Rightarrow$ угол движения точки, вычисляем $X(a)$, $Y(a)$, строим по точкам.

Построение графиков в полярной системе координат

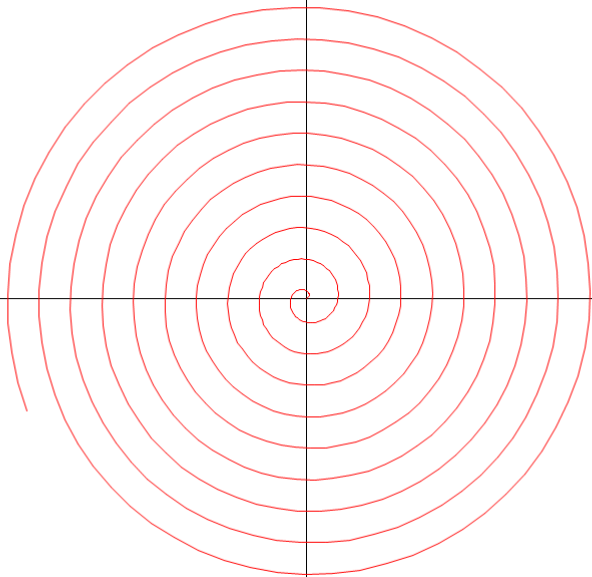
Меня заинтересовала данная тема и я самостоятельно рассмотрел построение графиков функций в полярной системе координат:

- Астроида
- Спираль Архимеда
- Многолепестковая роза
- Улитка Паскаля
- Кардиоида

Графики функций в полярной системе координат:

Спираль Архимеда

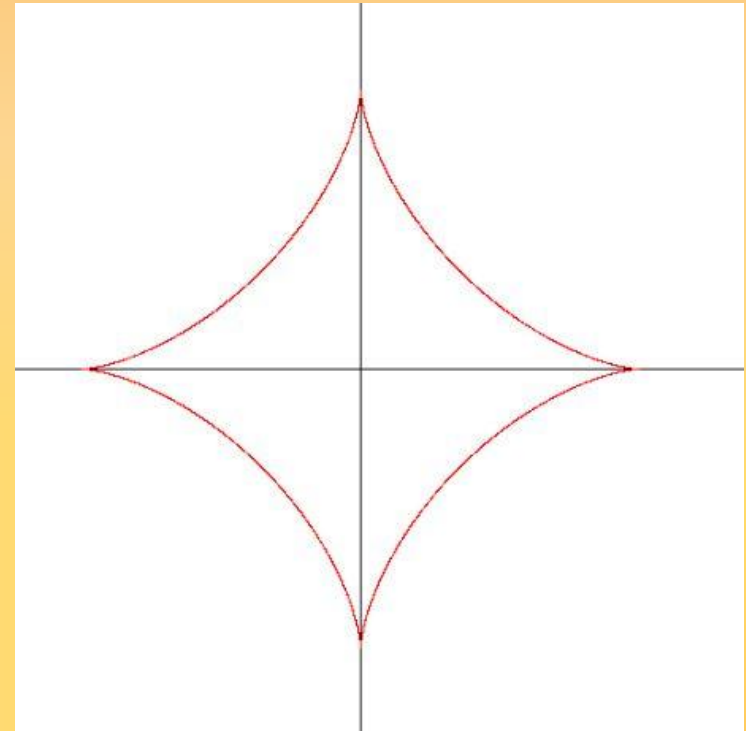
Введите центр:
Введите масштаб:
Введите внешний радиус:
Начальный угол спирали в градусах t:



```
x:=xc+round(m*r1*cos(t));  
y:=yc-round(m*r1*sin(t));|
```

$r1:=t/2; t \in [r1; r] \text{ с } H=0.1$

Астроида

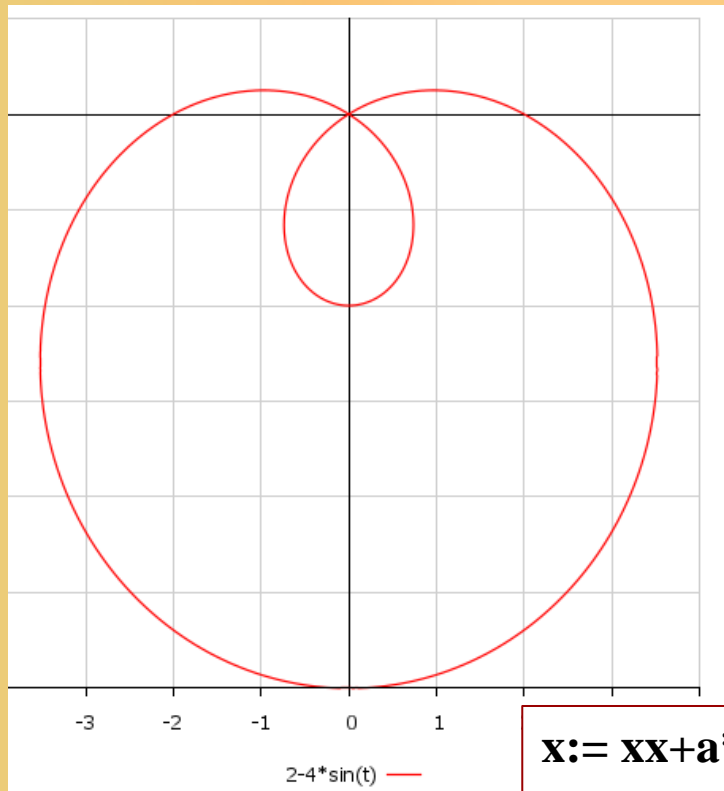


```
x:=x0+round(b*cos(t)*cos(t)*cos(t));  
y:=y0+round(b*sin(t)*sin(t)*sin(t));
```

$t \in [0; 2*Pi]$

Графики функций в полярной системе координат:

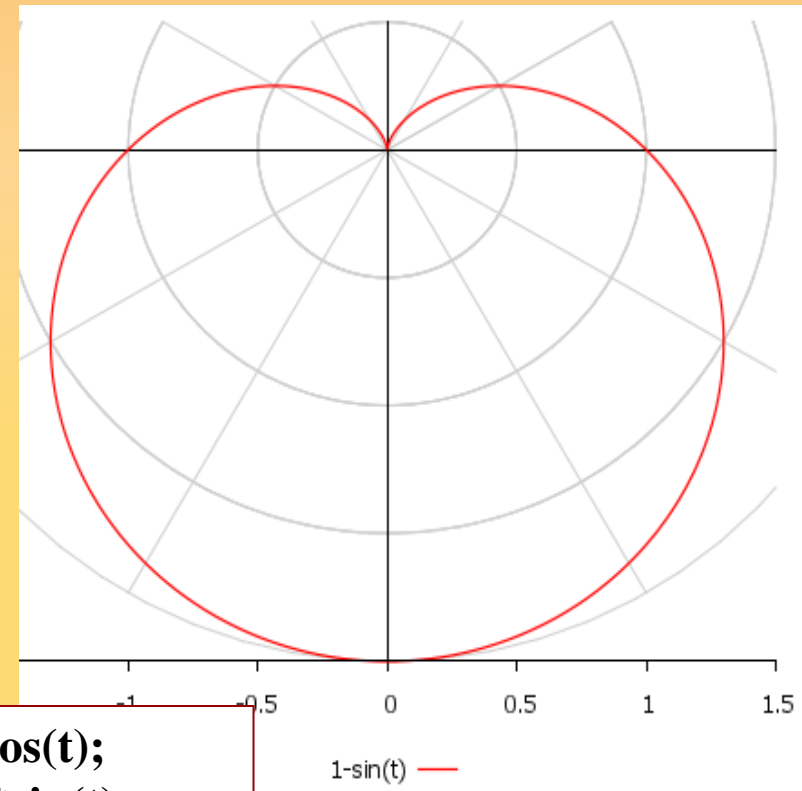
Улитка Паскаля



$$\begin{aligned}x &:= xx+a*\text{sqr}(\cos(t))+b*\cos(t); \\y &:= yy+a*\cos(t)*\sin(t)+b*\sin(t); \end{aligned}$$

$$r(t) = 2 - 4\sin(t), t \in [0; 2\pi]$$

Кардиоида

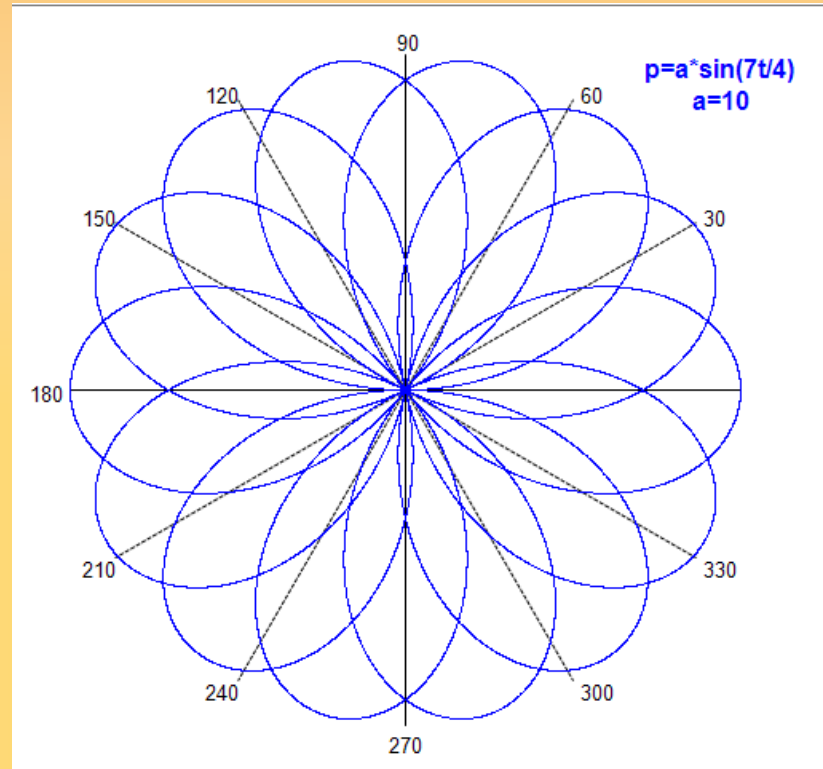


$$r(t) = 1 - \sin(t), t \in [0; 2\pi]$$

Программа многолепестковой розы и (без дополнительного модуля)

•Восьмилепестковая роза (полный график).pas*

```
//p=a sin (4u/3) (восьми-лепестковая роза);
uses graphABC;
var x0,y0,d,r,i,x1,y1:integer;
    t,p,a:real;
    s:string;
begin
x0:=windowwidth div 2; //центр
y0:=windowheight div 2;
a:=10;
//ГРАФИК
t:=0;
setpenwidth(2);
setpencolor(clBlue);
while t<8*pi do
begin
p:=a*sin(7*t/4);
setpixel(x0+round(d*p*cos(t)),y0-round(d*p*sin(t)),clBlue);
t:=t+0.001;
end;
end.
```



Программа Полярной розы

Модуль Graphs

Модуль Graphs – это модуль, который я сделал для того, чтобы рисовать сложные фигуры. На данный момент в нем есть: Астроида, Парабола, Кубическая парабола, Розы по спирали, Роза, Многогранник, Спираль Архимеда, Улитка Паскаля, Круги по кругу. Этот список будет пополняться.

Модуль состоит из вспомогательных процедур, и сам запускаться не может, так как в нём нет конкретных значений (коэффициентов). Фактические значения задаются в отдельной главной программе, в которой и происходит обращение к модулю.

[Модуль Graphs*](#)

```
1 uses GraphABC;  
   procedure spiral(xc,yc:integer;t,m,r:real);  
   var  
     x,y:integer;r1:real;  
   begin  
     r1:=0;//текущий радиус  
     setpencolor(clRed);  
     moveto(xc,yc);  
  
     while r1<=r do  
       begin  
         r1:=t/2;  
         x:=xc+round(m*r1*cos(t));  
         y:=yc-round(m*r1*sin(t));  
         lineto(x,y);  
         t:=t+0.1;  
         sleep(1);  
       end;  
     end;
```

Название процедуры

Локальные переменные
процедуры

Фрагмент модуля
Graph

Тело процедуры

```
uses graphabc, graphs;
var
  a, xx, yy, n, c, NN: integer; x0, y0, b, xc, yc, m, i, r: integer; t, x, y: real;
begin
  writeln('введите число от 1 до 11');
  writeln('1 - Астроида');
  writeln('2 - Спираль Архимеда');
  writeln('3 - Улитка Паскаля');
  writeln('4 - Кубическая парабола');
  writeln('5 - Многогранник');
  writeln('6 - Парабола');
  writeln('7 - Роза по спирали');
  writeln('8 - Парабола (с др. зн.)');
  writeln('9 - Круги по кругу');
  writeln('10 - Роза');
  writeln('11 - Много роз');

repeat
  readln(a);
  Pen.Color:=clRandom;
  case a of
  1:begin
    writeln('Введите центр: ');readln(x0,y0);
    writeln('Введите масштаб(100-200): ');readln(b);
    line(0, y0, 1000, y0);
    line(x0, 0, x0, 800);
    actroid(x0,y0,b);
  end;
  2:begin
    writeln('Введите центр: ');readln(xc,yc);
    writeln('Введите масштаб(1-10): ');readln(m);
```

**Основная
программа**

**Глобальные,
фактически
переменные
программы**

**Обращение к
процедуре
ASTROID
из модуля
Graphs**

Заключение

В данной работе я усовершенствовал PascalAbc, добавив в него свой модуль, и доволен этим достижением, потому что не каждый может это сделать.

Также я смоделировал различные графики функций в полярной системе координат, получив весьма необычные: например - в следствии изменения значений в функции розы и добавлении спирали, у нас получилось розы по спирали.

После данной проделанной работы, я решил отправить свой модуль разработчикам. **А вдруг они его поставят в клиентскую (главную) версию PascalABC?**

ЦОР

<http://grafikus.ru/examples/polar-functions>

<http://ru.stackoverflow.com>

<http://pascalabc.net/>

