

Pascal - grafika

1

Moduł graph

- Pascal zawiera standardowy moduł do tworzenia obiektów graficznych – linii, punktów, figur geometrycznych itp.
- Chcąc go użyć należy w programie (w nagłówku) wstawić deklarację:

```
uses graph;
```

2

Uruchomienie trybu graficznego

- `InitGraph (Sterownik: Integer, Tryb: Integer, Ścieżka: String);`
 - **sterownik** – identyfikator karty sterownika graficznego;
 - **tryb** – tryb graficzny (rozdzielczość, ilość kolorów);
 - **ścieżka** – katalog, w którym znajduje się sterownik *.bgi (zazwyczaj `c:\tp\bgi`)

3

Domyślny tryb graficzny

- Wykrywanie domyślnego trybu graficznego i karty graficznej

```
Sterownik:=Detect;
```

- lub

```
Sterownik:=0;
```

4

Zamknięcie trybu graficznego

- CloseGraph;
zamyka tryb graficzny (powrót do trybu tekstowego)

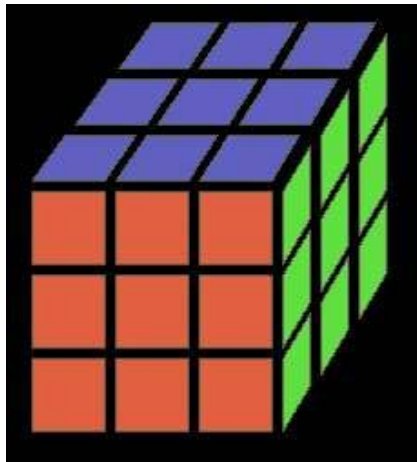
5

Szablon programu graficznego

```
program graffika;  
uses crt, graph;  
var i,j,k,x,y:integer;  
    sterownik, tryb:integer;  
  
begin  
    sterownik:=detect;  
        {uruchomienie trybu graficznego}  
    initgraph (sterownik, tryb, 'c:\tp\bgi');  
  
    ...{tu wpisujemy tekst programu}  
    closegraph; {zamknięcie trybu graficznego}  
    readln;  
end.
```

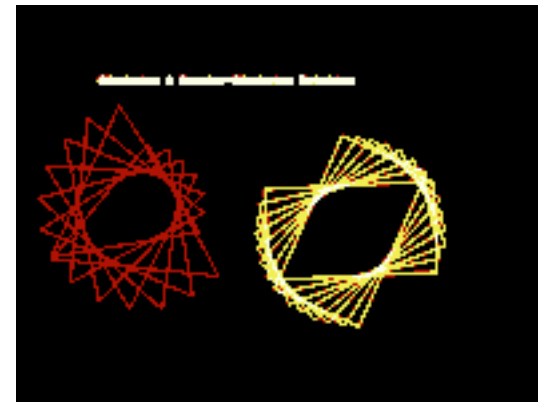
6

Przykłady – kostka rubika



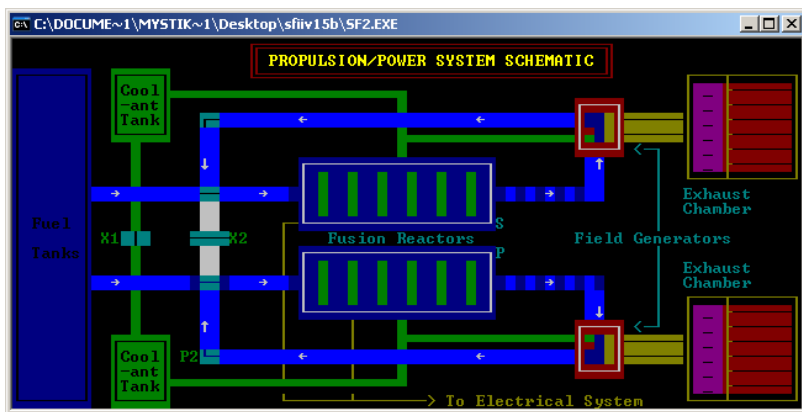
7

Przykłady – obroty trójkątów



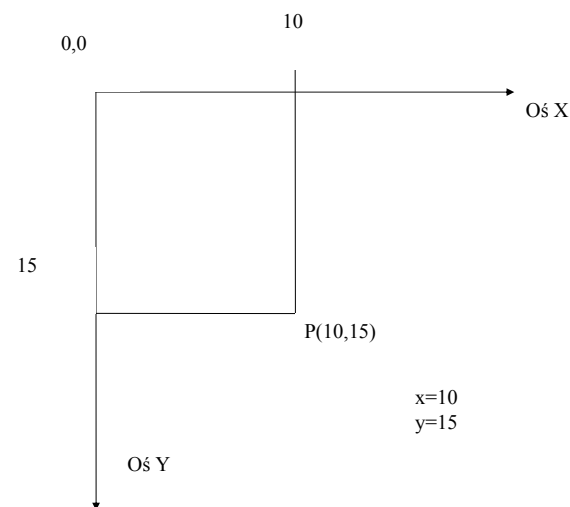
8

Przykłady – schemat układu



9

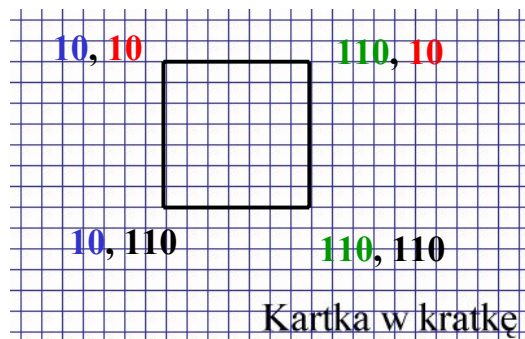
Układ ekranu w Pascalu



10

Zalecany tok postępowania

1. Zaznacz na kartce papieru jaką chcesz narysować figurę.



2. Policzyc współrzędne punktów.
3. Wpisać polecenie pascalowe:

```
line (10,10,110,110)
```

11

Wielkość ekranu w pascalu

- Operacje pobrania rozmiarów ekranu do zmiennych zewnętrznych

```
x:=GetMaxX;
```

```
y:=GetMaxY;
```

- Po zamknięciu trybu graficznego

```
writeln (x);
```

```
writeln (y);
```

12

Najprostszy program graficzny

```
program graffika;
uses crt, graph;

var i,j,k,x,y:integer;
    sterownik, tryb:integer;

begin
  sterownik:=detect;
  initgraph (sterownik, tryb, 'c:\tp\bgi');
  x:=GetmaxX;
  y:=GetmaxY;
  closegraph;
  writeln ('wspolrzedna x ekranu wynosi ',x);
  writeln ('wspolrzedna y ekranu wynosi ',y);
  readln;
end.
```

13

Rysowanie figur geometrycznych

14

Linia

- `line (x1, y1, x2, y2);`
 - **x1, y1** to współrzędne X,Y początku linii,
 - **x2, y2** to współrzędne X,Y końca linii

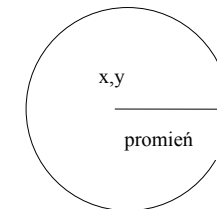
x1,y1 x1,y2

- Składnia
`Line (x1, y1, x2, y2: Integer);`
- Przykład
`Line (50,75,240,240);`

15

Koło

- `circle (x,y, promień);`
 - **x, y** to współrzędne X,Y środka,
 - **promień** to długość promienia koła,

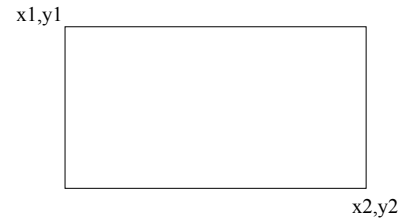


- Składnia
`Circle (X,Y: Integer; Radius: Word);`
- Przykład
`Circle (300,300,100);`

16

Prostokąt

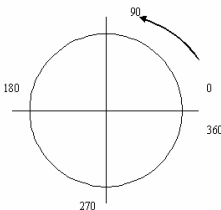
- `rectangle (x1, y1, x2, y2: Integer);`
 - **x1, y1** to współrzędne X,Y lewego górnego rogu prostokąta,
 - **x2, y2** to współrzędne X,Y prawego dolnego rogu prostokąta



- Składnia
`Rectangle (x1, y1, x2, y2: Integer);`
- Przykład
`Rectangle (100,100,250,320);`

17

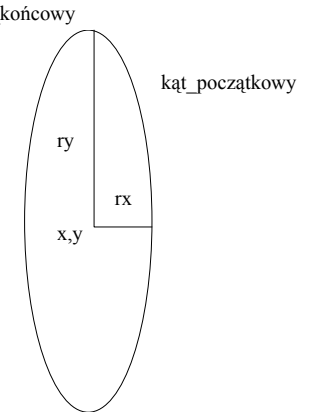
Kierunek rysowania elipsy



19

Elipsa

- `ellipse (x,y, ką_t_pocz,ką_t_kon, rx,ry);`
 - **x, y** to współrzędne X,Y środka,
 - **ką_t_pocz** to ką_t_początkowy (zalecane 0). Ką_t odpowiada godzinie 12.
 - **ką_t_kon** to ką_t_końcowy (zalecane 360). Ką_t jest liczony od godziny 12 zgodnie z ruchem wskazówek zegara.
 - **rx** to długość promienia na osi X,
 - **ry** to długość promienia na osi Y



- Składnia
`Ellipse(X, Y: Integer; StAngle, EndAngle: Word; XRadius, YRadius: Word);`
- Przykład
`Ellipse(70,400,0,360,60,30);`

18

Przykład programu rysującego figury geometryczne

```
program figury;
uses crt, graph;

var sterownik, tryb:integer;

begin
  sterownik:=detect;
  initgraph (sterownik, tryb, 'c:\tp\bgi');
  {rysujemy koło}
  circle (200,200,100);
  {rysujemy prostokąt}
  rectangle(300,300, 450,450);
  {rysujemy elipsę}
  ellipse (400, 100, 0,360,60,30);
  {rysujemy linię}
  line (50,50,100,400);
  readln;
end.
```

20

Kolor linii i figur

- Kolor realizujemy poleceniem
`setcolor(nazwa lub numer koloru)`

- Przykład

```
- setcolor(red);  
- setcolor(4);  
  
- for i:=0 to 15 do  
    setcolor(i);
```

21

Lista kolorów

Numer	Nazwa	Kolor	Numer	Nazwa	Kolor
0	black	Czarny	8	darkgray	Ciemnoszary
1	blue	Niebieski	9	lightblue	Jasnoniebieski
2	green	Zielony	10	lightgreen	Jasnozielony
3	cyan	Turkusowy	11	lightcyan	Jasnoturkusowy
4	red	Czerwony	12	lightred	Jasnoczerwony
5	magenta	Karmazynowy	13	lightmagenta	Jasnokarmazy- nowy
6	brown	Brązowy	14	yellow	Żółty
7	lightgray	Jasnoszary	15	white	biały

22

Przykład programu kolorowe figury

```
program kolorowe_figury;  
uses crt, graph;  
  
var sterownik, tryb:integer;  
  
begin  
    sterownik:=detect;  
    initgraph(sterownik, tryb, 'c:\tp\bgi');  
    {Ustalamy kolor}  
    setcolor(red);  
    {rysujemy kolo}  
    circle(200,200,100);  
    setcolor(blue);  
    circle(200,200,120);  
    setcolor(green);  
    circle(200,200,140);  
    setcolor(yellow);  
    circle(200,200,160);  
    readln;  
end.
```

23

Ćwiczenie

1. Narysuj kółka olimpijskie
2. Narysuj szachownicę żółto - czerwoną z kwadratów.

24

Animacje

- Moduł **graph** pozwala na narysowanie prostych animacji.
- Realizowane jest to w następującej kolejności:
 - `Circle(100,100,i);`
narysowanie pewnego kształtu
 - `Delay (100);`
Wstrzymanie pracy na pewien czas
 - `Cleardevice;`
Wyczyszczenie ekranu

25

Przykład animacji

```
for i:=1 to 100 do
  (pętla realizująca zmianę wielkości promienia – parametr i)
begin
  circle(100,100,i);
  (narysowanie koła zmieniającego swój promień)
  delay(100);
  (czas opóźnienia w milisekundach)
  cleardevice;
  (wymazanie ekranu)
end;
```

26

Przykład programu

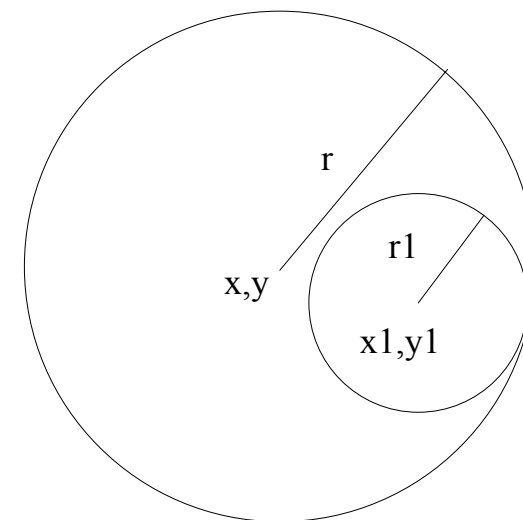
```
program animacja_kolek;
uses crt, graph;

var i,j,k,x,y:integer;
    sterownik, tryb:integer;

begin
  sterownik:=detect;
  initgraph (sterownik, tryb, 'c:\tp\bgi');
  i:=1; x:=100; y:=100;(Wstępny rozmiar promienia i współrzędnych środka koła)
  repeat
    circle(100,100,i);      (narysowanie koła)
    delay(100);             (czas opóźnienia w milisekundach)
    cleardevice;           (wymazanie ekranu)
    i:=i+1;                 (zmiana wielkości promienia koła)
    i:=i mod 100;          (po osiągnięciu 100 promień zmienia się na 0)
  until (keypressed);
  readln;
end.
```

27

Ruchome koła



28

```

program epicykle;
uses crt,graph;
var x,y,x1,y1,x2,y2,r,r1,r2,i,j:integer;
    sterownik, tryb:integer;

begin
sterownik:=detect;
InitGraph(sterownik, tryb,'c:\tp\bgi');
x:=250;y:=250;
r:=75; r1:=20;
r2:=5; i:=1;

repeat
begin
circle(x,y,r);
x1:=x+trunc((2*r+r1)*cos(i*pi/180));
y1:=y+trunc((2*r+r1)*sin(i*pi/180));
circle(x1,y1,r1);

x2:=x1+trunc((2*r1+r2)*cos(7*i*pi/180));
y2:=y1+trunc((2*r1+r2)*sin(7*i*pi/180));
circle(x2,y2,r2);

delay(10);
cleardevice;

i:=i+1;
i:=i mod 360;
end
until(keypressed);
end.

```