

FUNKCJE

Funkcja w matematyce wylicza jakąś wartość dla podanego argumentu. Na przykład funkcja sinus dla kąta równego 90° wylicza wartość 1.

W programowaniu funkcja pełni identyczną rolę jak i procedura (fragment programu zamknięty w jedną całość). Od procedury odróżnia ją możliwość wyliczania (jak w matematyce) jakiejś wartości. W pascalu istnieje wiele gotowych funkcji (np. cosinus sinus itp.), ale też można definiować swoje własne. Funkcje deklarujemy w podobnym miejscu programu jak procedury.

```
FUNCTION kwadrat(liczba:byte):longint;  
begin  
    kwadrat:=liczba*liczba;  
end;
```

W programie głównym funkcję nie możemy wywołać tak bezpośrednio, jak procedurę. Funkcja musi być skojarzona ze zmienną lub jej wynik podany bezpośrednio jako argument dla innej procedury lub funkcji.

```
k:=kwadrat(10);           albo           writeln(kwadrat(10));
```

Wewnątrz funkcji można zadeklarować zmiennne i stałe lokalne, które funkcjonują jedynie podczas działania funkcji i giną po jej zakończeniu (nie zajmują pamięci). Nie kolidują te zmiennne ze zmiennymi o takich samych nazwach zadeklarowanych w programie głównym (przesłanianie zmiennych). We wnętrzu funkcji musi wystąpić instrukcja przypisania do nazwy funkcji wyliczonej wartości (kwadrat:=...), tą wartość będzie zwracała funkcja, gdy wykonamy ją w programie głównym.

FUNKCJE W GRAFICE

Bardzo łatwo jest narysować linie, które są określone końcowymi punktami. Gorzej, jeśli mamy podany punkt początkowy, długość boku i należy linię wykreślić pod odpowiednim kątem, jak np. w logo. Ponieważ w Pascalu brak jest odpowiednich instrukcji, dlatego je sobie zdefiniujemy.

Punkt końcowy linii (X1,Y1) wyliczymy z następujących wzorów:

$$X1 = X+dX = X+bok*\cos(kat*pi/180)$$

$$Y1 = Y+dY = Y+bok*\sin(kat*pi/180)$$

(kąąt podajemy w stopniach, dlatego należy zamienić go na radiany - stała PI jest ustalona przez system). DX i DY należy zamienić na odpowiednie funkcje o nazwach deX i deY.

Gotowy wzór w pascalu będzie miał postać:

$$dX:=round(bok*\cos(kat*pi/180)) \text{ i } dY:=round(bok*\sin(kat*pi/180))$$

zaś gotowe funkcje będą miały postać:

```
function deX(bok,kat:integer):integer;  
begin deX:=round(bok*cos(kat*pi/180)) end;
```

```
function deY(bok,kat:integer):integer;  
begin deY:=round(bok*sin(kat*pi/180)) end;
```

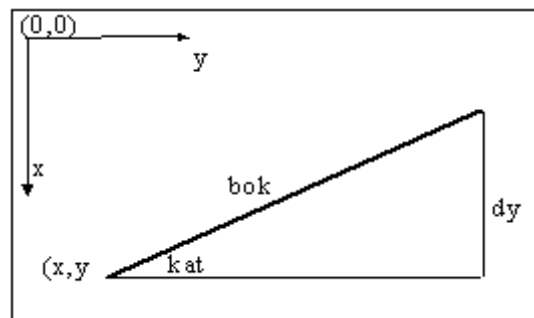
ROUND

W obu funkcjach użyliśmy dodatkowej funkcji ROUND, która zamienia liczbę rzeczywistą, powstałą w wyniku mnożenia i dzielenia na liczbę całkowitą, bo tylko w oparciu o liczby całkowite można rysować na ekranie graficznym.

Rysowanie odcinka za pomocą funkcji deX i deY

Podajemy punkt początkowy, długość i kąt oraz kolor

```
procedure linia(x,y,bok,kat,kolor:integer);  
begin  
    setcolor(kolor);  
    moveto(x,y);  
    lineto(x+deX(bok,kat),y-deY(bok,kat));  
end;
```



Rysowanie kwadratu za pomocą instrukcji LINIA

Kwadrat o boku 100, lewy górny róg w punkcie 200,200, kolor 12

```
linia(200,200,100,0,12);  
linia(300,200,100,90,12);  
linia(300,100,100,180,12);  
linia(200,100,100,270,12);
```

Animacja

kręcący się wskaźnik

```
for kat:=360 downto 1 do  
begin  
  linia(300,200,200,kat,13);  
  delay(20);  
  linia(300,200,200,kat,0);  
end;
```

sekundnik

```
kat:=360;  
repeat  
  linia(320,200,200,kat,11);  
  delay(1000);  
  linia(320,200,200,kat,0);  
  kat:=kat-6; {co 6 stopni skok}  
  if kat <=0 then kat:=360;  
until keypressed;
```

kręcący się i odbijający się radarek

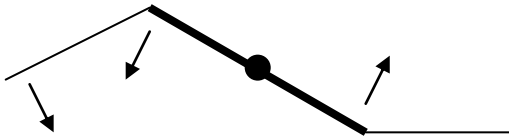
```
repeat  
  linia(x,y,bok,kat,13);  
  delay(10);  
  linia(x,y,bok,kat,0);  
  kat:=kat+3;  
  x:=x+dx;  
  y:=y+dy;  
  if kat >= 360 then kat:=0;  
  if (x > (640-bok)) or (x < 0) then dx:=-dx;  
  if (y > (480-bok)) or (y < 0) then dy:=-dy;  
until keypressed;
```

Inne instrukcje trybu graficznego

CLEARDEVICE	wyczyszczenie ekranu graficznego
GETMAXX,	funkcje zwracają wartości maksymalnych współrzędnych poziomej
GETMAXY	i pionowej ekranu graficznego - tryb VGA to 639,479
SETBKCOLOR(kolor)	ustalenie koloru tła
SETCOLOR(kolor)	ustalenie koloru rysowanych linii i napisów
OUTTEXTX(x,y,tekst)	wykreślenie napisu o treści tekst (string) od punktu x,y
CIRCLE(x,y,r)	okrąg o środku w punkcie x,y i promieniu r
PUTPIXEL(x,y,kolor)	narysowanie piksela w punkcie x,y i kolorze
RECTANGLE(x1,y1,x2,y2)	narysowanie prostokąta o podanych wierzchołkach

ĆWICZENIA

1. Obracająca się linia zostawia za sobą kolorowe punkty (PUTPIXEL) - jedno lub wielokolorowe (RANDOM)
2. Linia z zadania 1 dodatkowo przemieszcza się po ekranie (animacja)
3. Linia z zadania 2 odbija się od brzegów ekranu (warunki logiczne)
4. Po ekranie porusza się kilka obracających się linii (tablica). Losowo dobieramy kolor, długość, prędkość i kierunek obrotu.
5. Wokół środka obraca się odcinek, a na jego końcach obracają się dodatkowe odcinki.



6. Kwadrat obraca się wokół swojego środka
7. Kwadrat obraca się wokół jednego ze swoich rogów
8. Kwadrat z zadania 6 i 7 porusza się i odbija od brzegów