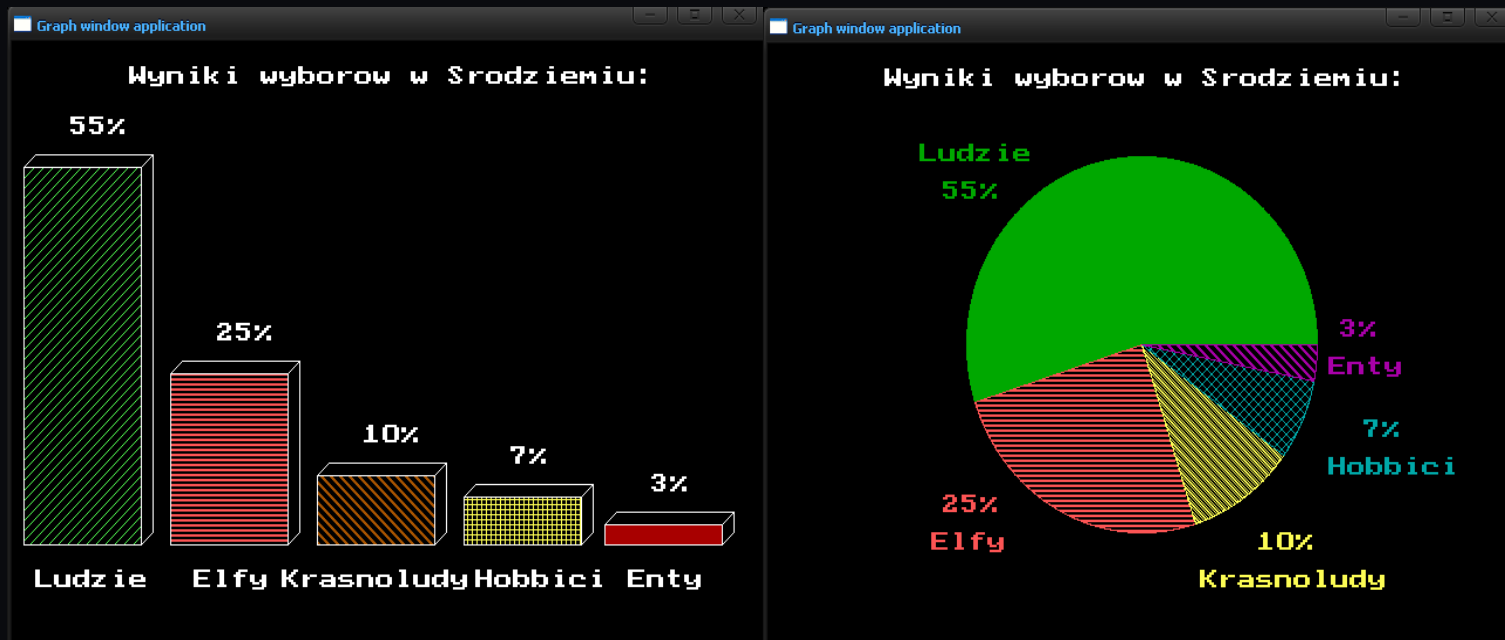


# Moduł graph w Pascalu

Wykład: uses graph, InitGraph, Detect, SetGraphMode, kolory, wypełnienia, linie, SetFillStyle, FloodFill, SetLineStyle, SetTextStyle, OutTextXY, SetTextJustify, GetMaxX, ClearDevice, MoveTo, LineTo, PutPixel, Rectangle, Circle, Bar3D, Ellipse, Arc, PieSlice

# MODUŁ GRAPH W PASCALU

Biblioteka dostępna po dodaniu klauzuli: `uses Graph;` służy zazwyczaj do graficznej prezentacji wyników obliczeń:





# INICJOWANIE TRYBU GRAFICZNEGO VGAHi 640x480 W DEV-PASCALU

```
uses Crt,Graph;  
  
var sterownik,tryb:smallint;  
  
begin  
  
sterownik:=VGA; tryb:=VGAHi;  
InitGraph(sterownik,tryb,'');  
  
//tutaj instrukcje graficzne  
  
CloseGraph;  
end.
```

W Dev-Pascalu może pozostać puste, we Free Pascalu należy tutaj podać ścieżkę do katalogu BGI np. C:\fpc\BGI

# INICJOWANIE TRYBU GRAFICZNEGO W WYSOKIEJ ROZDZIELCZOŚCI

```
uses Crt,Graph;  
  
var sterownik,tryb:smallint;  
  
begin  
  
sterownik:=Detect; tryb:=Detect;  
InitGraph(sterownik,tryb,'');  
  
//tutaj instrukcje graficzne  
  
CloseGraph;  
end.
```

Nastąpi autodetekcja  
karty i trybu graficznego  
wykorzystywanego na  
danym komputerze;  
Zamiast Detect można  
wpisać 0



# INICJOWANIE TRYBU GRAFICZNEGO Z UŻYCIEM STEGRAPHMODE()

```
uses Crt, Graph;
```

```
var sterownik, tryb: smallint;
```

```
begin
```

```
sterownik:=Detect; tryb:=Detect;
```

```
InitGraph(sterownik, tryb, '');
```

```
CloseGraph;
```

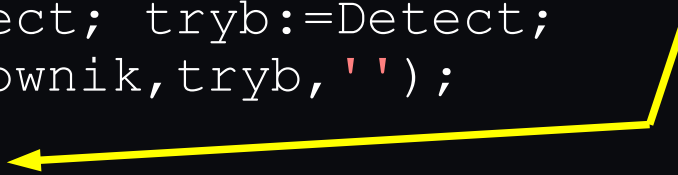
```
SetGraphMode(4);
```

```
//tutaj instrukcje graficzne
```

```
CloseGraph;
```

```
end.
```

Za pomocą funkcji  
SetGraphMode()  
ustawiamy wybraną dla  
programu rozdzielczość,  
pierwotne okno w  
wysokiej rozdzielczości  
zamykamy tuż po inicjacji





# KRÓTKI PRZEGLĄD STEROWNIKÓW...

Można używać zamiennie numerów i nazw:

- 0 = Detect
- 1 = CGA
- 2 = MCGA
- 3 = EGA
- 4 = EGA64
- 5 = EGAMono
- 6 = IBM8514
- 7 = HercMono
- 8 = ATT400
- 9 = VGA
- 10 = PC3270

## ...ORAZ TRYBÓW GRAFICZNYCH

EGALo	= 640 x 200,	16 kolorów
EGAHi	= 640 x 350,	16 kolorów
EGA64Lo	= 640 x 200,	16 kolorów
EGA64Hi	= 640 x 350,	4 kolory
EGAMonoHi	= 640 x 480,	2 kolory ( czarno-białe )
HercMonoHi	= 720 x 348,	2 kolory ( czarno-białe )
ATT400C0	= 320 x 200,	4 kolory z palety nr 0
ATT400C1	= 320 x 200,	4 kolory z palety nr 1
ATT400C2	= 320 x 200,	4 kolory z palety nr 2
ATT400C3	= 320 x 200,	4 kolory z palety nr 3
ATT400Med	= 640 x 200,	2 kolory ( czarno-białe )
ATT400Hi	= 640 x 400,	2 kolory ( czarno-białe )
VGALo	= 640 x 200,	16 kolorów
VGAMed	= 640 x 350,	16 kolorów
VGAHi	= 640 x 480,	16 kolorów
PC3270Hi	= 720 x 350,	2 kolory ( czarno-białe )
IBM8514Lo	= 640 x 480,	256 kolorów
IBM8514Hi	= 1024 x 768,	256 kolorów

To w zasadzie lekcja historii, bo wielu z nich nasze nowoczesne karty, a już zwłaszcza monitory nie obsługują 😊

**WNIOSEK:** pozwól na autodetekcję karty 😊

# PRZEGLĄD PODSTAWOWYCH KOLORÓW

Black	0	Czarny
Blue	1	Niebieski
Green	2	Zielony
Cyan	3	Morski
Red	4	Czerwony
Magenta	5	Fioletowy
Brown	6	Brązowy
LightGray	7	Jasnoszary
DarkGray	8	Ciemnoszary
LightBlue	9	Jasnoniebieski
LightGreen	10	Jasnozielony
LightCyan	11	Jasnomorski
LightRed	12	Jasnoczerwony
LightMagenta	13	Jasnofioletowy
Yellow	14	żółty
White	15	Biały

Przykłady:

```
SetColor(yellow);
```

```
SetFillStyle(8,Cyan);
```



# PRZEGLĄD PODSTAWOWYCH WYPEŁNIEŃ

Można używać zamiennie numerów i nazw:

- 0 = EmptyFill → wypełnienie kolorem tła
- 1 = SolidFill → wypełnienie ciągłe
- 2 = LineFill → wypełnienie pogrubionymi liniami poziomymi
- 3 = LtSlashFill → wypełnienie liniami pochyłymi w prawo
- 4 = SlashFill → wypełnienie pogrubionymi liniami pochyłymi w prawo
- 5 = BkSlashFill → wypełnienie pogrubionymi liniami pochyłymi w lewo
- 6 = LtBkSlashFill → wypełnienie liniami pochyłymi w lewo
- 7 = HatchFill → wypełnienie siatką pionową
- 8 = xHatchFill → wypełnienie siatką ukośną
- 9 = InteerleaveFill → wypełnienie liniami splecionymi
- 10 = VideDotFill → wypełnienie kropkami
- 11 = CloseDotFill → wypełnienie zagęszczonymi kropkami

**Przykłady: `SetFillStyle(HatchFill,Cyan); FloodFill(450,110,BkSlashFill);`**

# PRZEGLĄD RODZAJÓW LINII


Można używać zamiennie numerów i nazw:

- 0 = SolidLn → ciągła
- 1 = DottedLn → kropkowana
- 2 = CenterLn → symetryczna
- 3 = DashedLn → przerywana
- 4 = UserBitLn → o wzorze zdefiniowanym przez użytkownika

Rodzaje grubości linii to:

- 1 = NormWidth → linia normalna
- 3 = ThickWidth → linia pogrubiona

Przykłady: `SetLineStyle(SolidLn,0,ThickWidth);`



PRZEGLĄD  
WYBRANYCH  
FUNKCJI  
GRAFICZNYCH



**SetColor(Kolor);**

Ustawia kolor pędzla dla wszystkich kolejnych figur na kolor podany w nawiasie.

**SetFillStyle(Wzorzec ,Kolor);**

Ustawia kolor pędzla oraz wypełnienie dla wszystkich kolejnych figur.

**FloodFill(x,y,Kolor);**

Wypełnia przestrzeń wypełnieniem ustawionym przez SetFillStyle. Zmienna kolor określa kolor ograniczający przestrzeń wypełnianą, czyli ustala obszar graniczny wypełnienia.

**SetLineStyle (Rodzaj, Wzor, Grubosc)**

Zmienia rodzaj linii oraz jej grubość.

## SetTextStyle(Krój, Kierunek, Rozmiar);

Pierwszy argument określa krój pisma (wymagany jest plik \*.chr zawierający czcionki → w DevPascalu jest on niedostępny):

- 0 = DefaultFont,
- 1 = TriplexFont,
- 2 = SmallFont,
- 3 = SansSerifFont,
- 4 = GothicFont.

Drugi z parametrów określa kierunek wyprowadzania tekstu.

- 0 = HorizDir - wyprowadzanie poziome od lewej do prawej,
- 1 = VertDir - wyprowadzanie pionowe z góry na dół.

Do określania rozmiaru pisma należy zastosować liczby 1..10

```
OutTextXY(x, y, 'Hello World!');
```

Wywołanie tej procedury powoduje wyprowadzenie na ekranie tekstu w punkcie o współrzędnych x i y.

```
SetTextJustify(Poziom, Pion);
```

Procedura ta umożliwia wyrównywanie lub centrowanie wyprowadzanego łańcucha względem wskaźnika ekranu lub określonego punktu. Predefiniowane stałe:

POZIOM: 1 = CenterText  
0 = LeftText  
2 = RightText

PION: 0 = BottomText  
2 = TopText



## GetMaxX(); GetMaxY();

Funkcje te zwracają informację o aktualnej rozdzielczości ekranu (szerokość i wysokość w pikselach).

## ClearDevice();

Po zainicjowaniu trybu graficznego dostępna jest cała powierzchnia ekranu. Jego zawartość może być w dowolnym miejscu programu oczyszczona poprzez wywołanie bezparametrowej procedury ClearDevice, która oprócz czyszczenia ekranu resetuje także przypisane wszystkim parametrom graficznym wartości.

## MoveTo(x, y);

Przesuwa wskaźnik ekranu (graficzny kursor) do punktu o współrzędnych (x,y). Wskaźnik ekranu jest zawsze odniesiony do punktu (0,0) → lewy górny róg bieżącego okna.

## LineTo (x,y);

Spowoduje narysowanie odcinka od miejsca aktualnego położenia graficznego wskaźnika ekranu do punktu o podanych w wywołaniu współrzędnych (x,y), po czym wskaźnik ekranu zostanie przesunięty do końca narysowanego odcinka.

## Line (x1,y1,x2,y2);

Wywołanie procedury spowoduje narysowanie odcinka od punktu ekranu o współrzędnych (x1,y1) do punktu o współrzędnych (x2,y2). Aktualne położenie graficznego wskaźnika ekranu nie ulega przy tym zmianie.

## PutPixel (x,y,kolor);

Rysuje piksel o zadanej kolorze na ekranie w miejscu określonym współrzędnymi x,y.

```
Rectangle(x1, y1, x2, y2);
```

Narysuj prostokąt, gdzie punkt  $(x_1, y_1)$  to lewy górny narożnik prostokąta, a  $(x_2, y_2)$  prawy dolny.

```
Bar3D(x1, y1, x2, y2, depth, true);
```

Narysuj prostopadłościan, gdzie punkt  $(x_1, y_1)$  to lewy górny narożnik przedniej ściany, a  $(x_2, y_2)$  prawy dolny.

Parametr `depth` określa głębokość prostopadłościanu - długość odcinka w rzucie osi Z na płaszczyznę.

```
Circle(x, y, r)
```

Rysuje okrąg o środku w punkcie  $x, y$  oraz promieniu o długości  $r$ .



**Ellipse(x,y,0,360,Rx,Ry);**

Narysuj elipsę o środku w punkcie (x,y), następnie podane są kąty w stopniach (od 0 do 360 daje pełną elipsę), Rx i Ry to dwa promienie wodzące elipsy

**Arc(x,y,a,b,R);**

Rysuj wycinek okręgu o środku w punkcie x,y oraz promieniu o długości R. Wycinek powstaje od wielkości kąta w stopniach a do b – np. a=0; b=180 daje półokrąg.

**PieSlice(320,240,a,b,R);**

Rysuj wycinek koła o środku w punkcie x,y oraz promieniu o długości R. Wycinek powstaje od wielkości kąta w stopniach a do b – np. a=0; b=180 daje półkole.