

Тема 2:

Основні прийоми програмування графіки на Паскалі

План:

- 2.1. Графічний режим Паскаля
- 2.2. Робота з покажчиком в графічному режимі
- 2.3. Малювання основних графічних примітивів і фігур
- 2.4. Зафарбовування фігур

Графічний модуль

Turbo Pascal – файл **graph.tpu**;

Pascal ABC – файл **graph.abc**.

Підключення:

```
Uses Graph;
```

Ініціалізація графічного режиму:

```
InitGraph(var Driver, Mode: integer;  
Path: string);
```

де `Driver` – змінна, яка визначає тип графічного драйвера;

`Mode` – змінна, яка задає режим роботи графічного адаптера;

`Path` – ім'я файлу драйвера і шлях до нього.

Типи графічних драйверів:

Driver	Значення
Detect	0
CGA	1
MCGA	2
EGA	3
EGA64	4
EGAMono	5
HercMono	6
ATT400	8
VGA	9
PC3270	10

Графічні режими:

Mode	Розр. здатність	Кольор. палітра	Кільк. сторінок
CgaHi=4	640□200	2 кольори	1
McgaMed=4	640□200	2 кольори	1
McgaHi=5	640□480	2 кольори	1
EgaLo=0	640□200	16 кольорів	4
EgaHi=1	640□350	16 кольорів	2
Ega64Lo=0	640□200	16 кольорів	1
Ega64Hi=1	640□350	4 кольори	1
HercMonoH1=0	720□348		2
VgaLo=0	640□200	16 кольорів	4
VgaMed=1	640□350	16 кольорів	2
VgaHi=2	640□480	16 кольорів	1
IBM8514Lo=0	640□480	256 кольорів	
IBM8514Hi=1	1024□768	256 кольорів	

Перевірка результату ініціалізації:

GraphResult

GrOk = 0; {нет ошибок}

GrInitGraph = -14..-1

Завершення роботи графічного режиму

CloseGraph

Структура програми:

- Підключення графічного модуля;
- Визначення графічного драйвера;
- Встановлення графічного режиму;
- Ініціалізація графічного режиму;
- Побудови;
- Закриття графічного режиму.

Приклад заготовки программы:

```
Program primer;  
Uses Graph;  
Var d , m : integer : {переменные для установки драйвера, режима}  
Begin  
  d:= detect;  
  InitGraph(d, m, "");  
  If GraphResult =0 then {если инициализация прошла успешно}  
  begin  
    <описание всех ваших построений>  
    closeGraph ;  
  end  
  else writeln ('произошла ошибка при инициализации графики');  
end .
```

На практиці використовується

$rx \square ry$

розрізняльна здатність:

$\square 320 \square 200;$

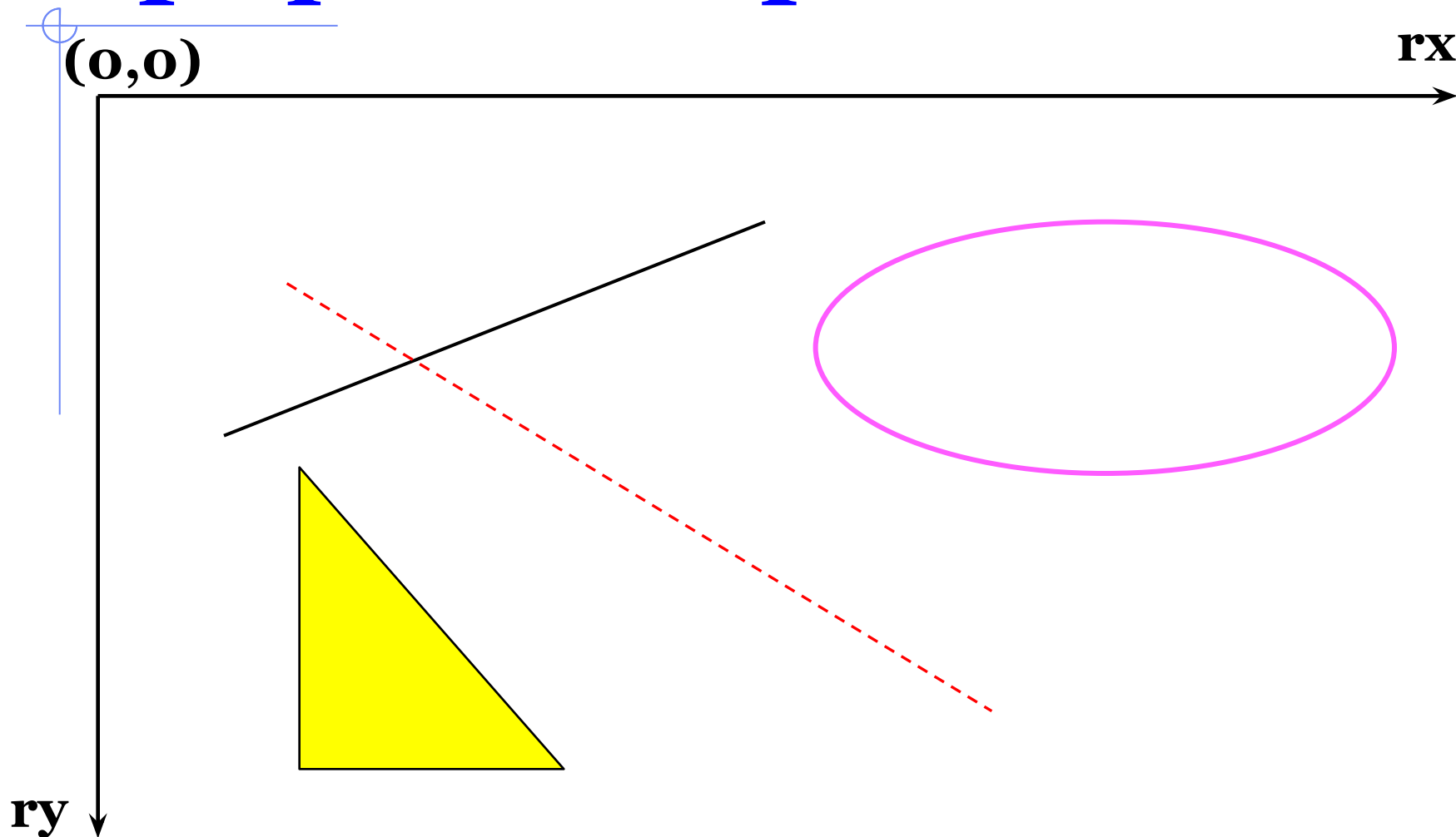
$\square 640 \square 480;$

$\square 800 \square 600;$

$\square 1024 \square 768;$

$\square 1280 \square 1024.$

Графічні координати



Переведення координат (x, y) в графічні (xg, yg)

$$xg = sx \square [x] + dx,$$

$$yg = ry - sy \square [y] - dy$$

де $[x]$ і $[y]$ – цілі частини x та y ,

sx та sy – масштабні множники

$$sx = (rx - 1) \setminus x_{\max}; \quad sy = (ry - 1) \setminus y_{\max} \quad (x_{\max} \text{ і}$$

y_{\max} – максимальні значення

геометричних координат)

dx, dy – розмір зсуву.

Визначення розміру екрану

Функції:

GetMaxX: Integer;

GetMaxY: Integer

Визначення поточних координат точки

Функції:

GetX: Integer;

GetY: Integer

Переміщення поточного показчика

Процедури:

MoveTo(X, Y: Integer);

MoveRel(DX, DY: Integer)

Очищення зображення

Процедури:

ClearDevice;

ClearViewPort

Константи кольору – змінна С

0	Чорний	8	Темно-сірий
1	Синій	9	Світло-синій
2	Зелений	10	Світло-зелений
3	Блакитний	11	Світло-блакитний
4	Червоний	12	Рожевий
5	Фіолетовий	13	Малиновий
6	Коричневий	14	Жовтий
7	Світло-сірий	15	Білий

Встановлення поточного кольору

Процедури:

SetBkColor(C: Word)

SetColor(C: Word)

Повернення поточного кольору

Функції:

GetBkColor

GetColor

GetPixel(X, Y: Integer)

Відображення точки на екрані

Процедура:

PutPixel(X, Y: Integer, C: Word)

MoveTo(X, Y: Integer)

Приклад



Виведення текстового рядка в графічному режимі

Процедури:

OutText(Str: String)

OutTextXY(X, Y: Integer, Str: String)

Настроювання параметрів форматування тексту

Процедури

SetTextJustify(h, v: word)

SetTextStyle(f: word; d: word; S: Char)

Приклад

```
SetColor(3);  
  for i:=0 to 10 do  
  begin  
    SetTextStyle(i,0,4);  
    OutText('pascal');
```



```
end;
```

Параметри лінії задає

процедура:

SetLineStyle (Type, Pattern, Thick: word)

де Type – тип лінії значення: 0, 1, 2, 3, 4

Pattern=0

Thick – товщина лінії в пікселях

Побудова прямої

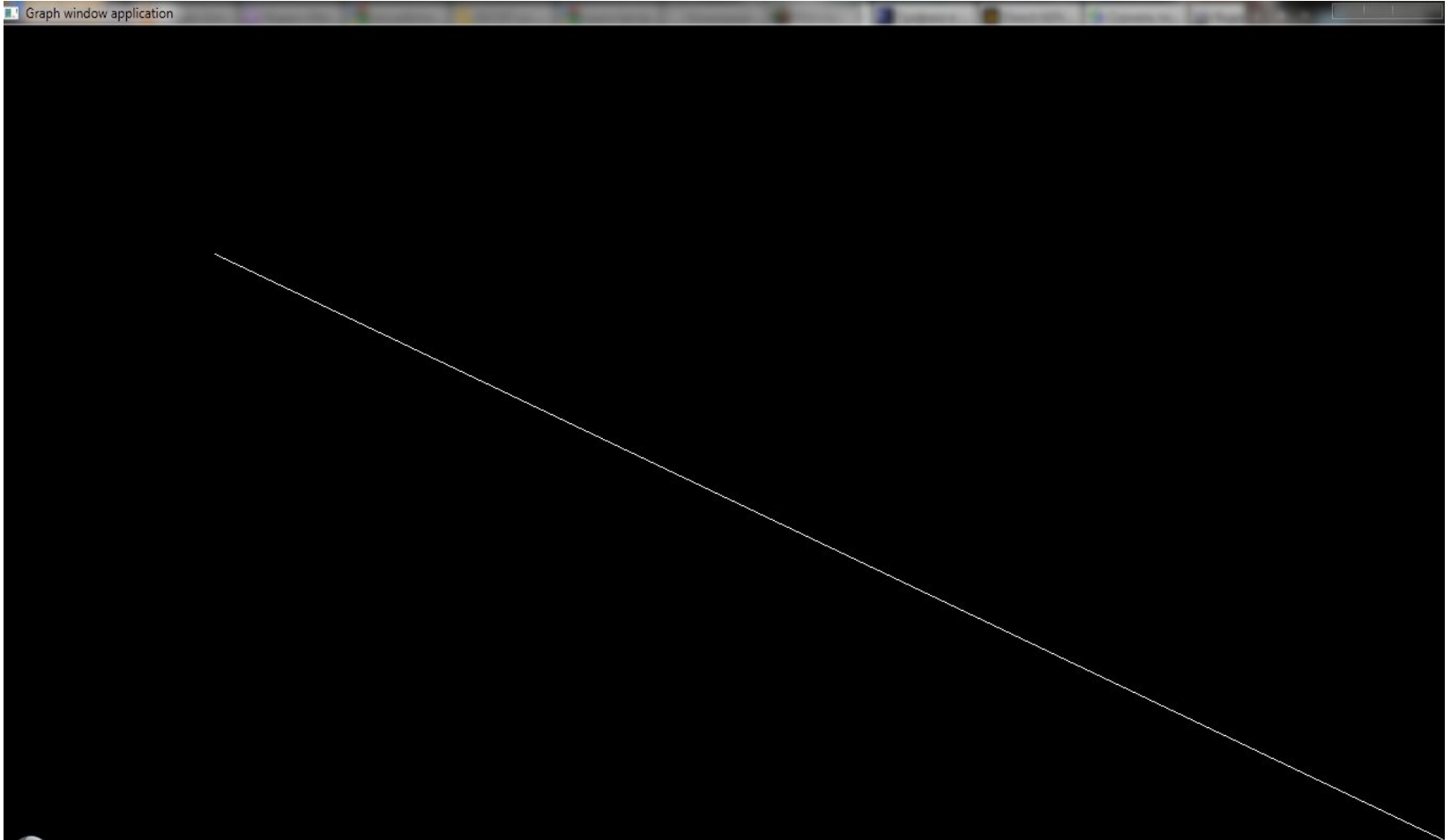
Процедури:

Line(X1, Y1, X2, Y2 : Integer);

LineTo(X, Y: Integer);

LineRel(Dx, Dy)

Приклад



Приклад:

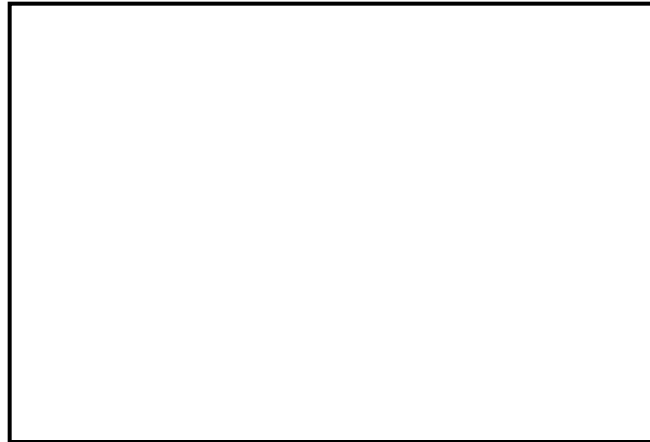


Побудова прямокутника

Процедура:

Rectangle (X1, Y1, X2, Y2: Integer)

(X1, Y1)



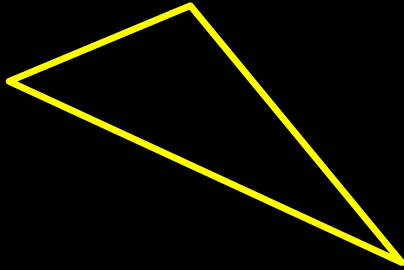
(X2, Y2)

Побудова багатокутника

Процедура:

```
DrawPoly(NumPoints : Word; Var  
PolyPoints)
```

Приклад:



Побудова кривих

Процедури:

Circle(X, Y: integer; R: word)

Arc(X, Y, Ugol_begin, Ugol_end, R: integer)

Побудова кривих

Процедури:

Ellipse(X,Y:integer; Ugol_begin,
Ugol_end, Rx, Ry: word)

Sector (X,Y: integer; Ugol_begin,
Ugol_end, Rx, Ry: word)

Визначення стилю та кольору зафарбовування

Процедура:

SetFillStyle(Style, Color: Word)

Стилi зафарбовування

0	заливка кольором фону
1	суцільне зафарбовування поточним кольором
2	горизонтальне штрихування
3	штрихування під кутом 45° вліво тонкими лініями
4	штрихування під кутом 45° вліво потовщеними лініями
5	штрихування під кутом 45° управо потовщене

Стилi зафарбовування

6	штрихування пiд кутом 45° управо тонке
7	заповнення +++++
8	заповнення пiд кутом 45° рiдкою косою клiткою
9	заповнення пiд кутом 45° частою косою клiткою
10	заповнення рiдкими крапками
11	заповнення частими крапками
12	узор визначається користувачем

Зафарбовування довільної замкненої фігури

Процедура:

FloodFill(X, Y, Border:word)

Процедури для побудови зафарбованих фігур:

Bar

Bar3D

PieSlice

Sector

FillEllipse