

Лекції 19-20.



Програмування графіки

*Мельник О.О.
доцент кафедри
програмного забезпечення систем*



План:

1. Графічний режим.
2. Стандартні графічні процедури і функції.
3. Зображення і стилі.
4. Мультиплікація.
5. Вивід текстової інформації.
6. Обробка помилок.



Рекомендована література:

1. Turbo Pascal 7.0: Пер. с нем. / Винфрид и Фолькер Кассера. – К.: Издательство «ДиаСофт», 2003.
2. Грызлов В.И., Грызлова Т.П. Турбо Паскаль 7.0. – 4-е изд., испр. – М.: ДМК Пресс, 2005.
3. Паскаль для персональных компьютеров: Справ. Пособие / Ю.С. Бородич, А.Н. Вальвачев, А.И. Кузьмич. – Мн.: Выш. Шк.: БФ ГИТМП «Ника», 1991.
4. Паскаль. Программирование на языке высокого уровня: Учебник для вузов / Т.А. Павловская. – СПб: Питер, 2003.
5. Программирование в среде Turbo Pascal 7.0 / Марченко А.И., Марченко Л.А.: Под ред.: Тарасенко В.П. – 8-е изд. – К.: ВЕК+, СПб: КОРОНА принт, 2004.
6. Зеленьяк О.П. Практикум программирования на Turbo Pascal. Задачи, алгоритмы и решения. – СПб.: ООО «ДиаСофтЮП», 2002.



Графічний режим

Формування зображення на екрані ПЕОМ відбувається за допомогою дисплейного адаптера.

Дисплей може працювати у *текстових* або *графічних* режимах. Текстові режими підтримує модуль *Crt*, графічні – модуль *Graph*.

Модуль Graph реалізує бібліотеку підпрограм, яка складається з понад 90 графічних процедур і функцій (побудова графічних примітивів, різноманітні види заповнення і типи ліній, шрифти тощо).



Графічний режим

У *графічному режимі* зображення на екрані будується у вигляді растра: поле екрана розбито на елементарні точки (піксели), кожна з яких має свій колір.

Графічний режим визначається кількістю точок растра і кількістю кольорів у палітрі.

GetMaxX, *GetMaxY* – стандартні функції модуля Graph, які повертають відповідно максимальні координати по осям X та Y в залежності від поточного режиму відеоадаптера.



Графічний режим

Координати екрана у графічному режимі





Графічний режим

Графічна система підтримує *поточний вказівник* (графічний курсор). Вказівник має властивості текстового курсору, але на відміну від нього є невидимим.

Для визначення поточних координат вказівника використовуються функції *GetX*, *GetY*, для його переміщення без виводу зображень на екрані – процедури *MoveTo*, *MoveRel*.



Графічний режим

Ініціалізація графічного режиму виконується процедурою

InitGraph (Driver, Mode, Path);

Driver:integer - тип драйвера:

Detect = 0; {автоматичне визначення типу}

CGA = 1; EGA = 3; VGA = 9;

Mode:integer - режим роботи:

{Адаптер VGA}

VGA_{Lo} = 0; {640*200, 16 кольорів, 4 стор.}

VGA_{Med} = 1; {640*350, 16 кольорів, 2 стор.}

VGA_{Hi} = 2; {640*480, 16 кольорів, 1 стор.}



Графічний режим

Path:string - ім'я файлу драйвера та шлях.

Для моніторів EGA, VGA, SVGA використовується, як правило, драйвер **egavga.bgi**, для моніторів CGA – **cga.bgi** .

Якщо драйвер розміщено у робочому каталозі, шлях до нього задається порожнім рядком **Path=""**.

Закриття графічного режиму проводиться процедурою **CloseGraph;**



Графічний режим

Приклад. Намалювати лінію від лівого верхнього до правого нижнього кута екрана.

```
Program LineDemo;  
uses Graph;  
var  
    gd, gm : integer;  
BEGIN  
    gd:= detect;  
    InitGraph (gd, gm, "");  
    Line (0, 0, GetMaxX, GetMaxY);  
    Readln;  
    CloseGraph  
END.
```



Графічний режим

Процедури та функції встановлення режиму

Назва	Призначення
GetGraphMode	Повертає код поточного режиму роботи адаптера
SetGraphMode (Mode)	Встановлює новий графічний режим.
DetectGraph (Driver, Mode)	Повертає тип драйверу та режим його роботи
GetDriverName	Повертає ім'я поточного графічного драйверу
GetMaxMode	Для поточного драйвера повертає максимальний номер можливого режиму



Стандартні графічні процедури і функції

Лінії та точки

Назва процедури	Призначення
PutPixel ($X, Y, Color$)	В координати (X, Y) (відносно лівого кута вікна або екрану) ставиться точка кольору $Color$
GetPixel (X, Y)	Повертає номер кольору точки (X, Y)
Line ($X1, Y1, X2, Y2$)	Виводить відрізок прямої від ($X1, Y1$) до ($X2, Y2$)
LineTo (X, Y)	Виводить відрізок прямої від поточного вказівника (курсору) до точки (X, Y)
LineRel (Dx, Dy)	Виводить відрізок прямої від курсору до точки, яка визначається через приріст координат Dx, Dy



Стандартні графічні процедури і функції

Графічні примітиви

Назва процедури	Призначення
Circle (X, Y, R)	Креслить коло з центром у точці (X, Y) радіуса R
Ellipse ($X, Y, BegA, EndA, RX, RY$)	Креслить еліптичну дугу: (X, Y) - координати центру; $BegA, EndA$ - початковий та кінцевий кути дуги; RX, RY - горизонтальний та вертикальний радіуси.
FillEllipse (X, Y, RX, RY)	Обводить лінією та зафарбовує еліпс



Стандартні графічні процедури і функції

Графічні примітиви

Назва процедури	Призначення
Sector <i>(X, Y, BegA, EndA, RX, RY)</i>	Креслить та заповнює еліптичний сектор: <i>X, Y, RX, RY</i> - див. Ellipse; <i>BegA, EndA</i> - початковий та кінцевий кути еліптичного сектору
PieSlice <i>(X, Y, BegA, EndA, R)</i>	Креслить та заповнює сектор кола



Стандартні графічні процедури і функції

Графічні примітиви

Назва процедури	Призначення
<i>Arc</i> (<i>X</i> , <i>Y</i> , <i>BegA</i> , <i>EndA</i> , <i>R</i>)	Креслить дугу кола: (<i>X</i> , <i>Y</i>) - центр; <i>BegA</i> , <i>EndA</i> - початковий та кінцевий кути дуги (в градусах, відлік проти годинникової стрілки), <i>R</i> - радіус.
GetArcCoords (<i>var Coords</i> : ArcCoordsType)	Повертає координати трьох точок: центру, початку та кінця дуги. Type ArcCoordsType = record <i>X</i> , <i>Y</i> : Integer; <i>Xstart</i> , <i>Ystart</i> : Integer; <i>Xend</i> , <i>Yend</i> : Integer; end;



Стандартні графічні процедури і функції

Графічні примітиви

Назва процедури	Призначення
Bar (<i>X1, Y1, X2, Y2</i>)	Заповнює прямокутну область екрану
Bar3D (<i>X1, Y1, X2, Y2, Depth, Top</i>)	Креслить тривимірне зображення паралелепіпеду і зафарбовує передню грань. <i>Depth</i> : word – висота тривимірного контуру (“глибина”). Якщо <i>Top</i> =True — верхня грань креслиться, інакше -ні.



Стандартні графічні процедури і функції

Графічні примітиви

Назва процедури	Призначення
Rectangle (<i>X1, Y1, X2, Y2</i>)	Креслить прямокутник (<i>X1, Y1</i>) — (<i>X2, Y2</i>)
DrawPoly (<i>N; var Points</i>)	Креслить контур багатокутника, де <i>N</i> - кількість точок (включаючи обидві крайні); <i>Points</i> : PointType , містить координати точок переходу: Type PointType = record <i>x, y</i> : Word end;



Зображення і стилі

Кольори та палітри

Назва процедури	Призначення
<i>SetColor (Color)</i>	Встановлює колір для ліній та символів
<i>GetColor</i>	Повертає код активного кольору
<i>GetMaxColor</i>	Повертає максимально доступний код кольору
<i>SetBkColor (Color)</i>	Встановлює колір фону
<i>GetBkColor</i>	Повертає код кольору фону



Зображення і стилі

Кольори та палітри

Назва процедури	Призначення
GetPalette (var <i>PaletteInfo</i> : <i>PaletteType</i>)	Повертає розмір та кольори активної палітри. const MaxColors = 15; type PaletteType = record Size: Word; {кількість кольорів в палітрі} Colors : array [0..MaxColors] of ShortInt {номери кольорів в палітрі} end ;



Зображення і стилі

Кольори та палітри

Назва процедури	Призначення
SetPalette(<i>N, Color</i>)	Замінює один з кольорів палітри на новий колір. <i>N</i> - номер кольору в палітрі, <i>Color</i> - номер встановленого кольору.
SetAllPalette(<i>var Palette</i>)	Змінює одночасно декілька кольорів палітри. <i>Palette</i> - нетипізований параметр. Перший байт - довжина палітри (<i>N</i>), наступні <i>N</i> байт - нові кольори палітри (значення дорівнює -1, якщо колір не міняється).
GetPaletteSize	Повертає розмір палітри
GetDefaultPalette (<i>var Palette : PaletteType</i>)	Повертає структуру палітри за умовчанням (<i>PaletteType</i> - див. <i>GetPalette</i>).



Зображення і стилі

Заповнення та стилі

Назва процедури	Призначення
SetFillStyle(Fill, Color)	Встановлює стиль заповнення: <i>Fill</i> - тип, <i>Color</i> - колір. <i>Fill</i> : Const EmptyFill = 0; {заповнення фоном} SolidFill = 1; {суцільне заповнення} LineFill = 2; { ----- } LtSlashFill = 3; { //////////////// } SlashFill = 4; { //////////////// - напівжирні } BkSlashFill = 5; { \\\\\\\\\\\\\\\\\\\\\\\ - напівжирні } LtBkSlashFill = 6; { \\\\\\\\\\\\\\\\\\\\\\\ } HatchFill = 7; { ++++++++ } XhatchFill = 8; { xxxxxxxx }



Зображення і стилі

Заповнення та стилі

Назва процедури	Призначення
SetFillStyle(<i>Fill</i>, <i>Color</i>)	Встановлює стиль (тип та колір) заповнення. <i>Fill</i> - тип, <i>Color</i> - колір. <i>Fill</i> : Const InterleaveFill = 9; { прямокутна клітинна } WideDotFill = 10; { рідкі крапки } CloseDotFill = 11; { часті крапки } UserFill = 12; { визначає користувач у процедурі SetFillPattern }



Зображення і стилі

Заповнення та стилі

Назва процедури	Призначення
SetFillPattern (<i>Pattern</i> : FillPatternType; <i>Color</i>)	Встановлює зріз штрихування та колір. Type FillPatternType = array [1..8] of Byte; Кожен байт задає 8 пікселів рядка заповнення.
GetFillPattern (var <i>Pattern</i> : FillPatternType)	Повертає зріз заповнення



Зображення і стилі

Заповнення та стилі

Назва процедури	Призначення
GetFillSettings(var PattInfo: FillSettingsType)	Повертає стиль заповнення (активний). Type FillSettingsType = record Pattern : Word; {взірець} Color : Word end;
FloodFill(X, Y, Border)	Заповнює замкнену фігуру, використовуючи стиль заповнення. (X, Y) - координати довільної точки фігури, <i>Border</i> - колір граничної лінії



Зображення і стилі

Заповнення та стилі

Назва	Призначення
SetLineStyle (<i>T</i> , <i>Pat</i> , <i>Thick</i>)	<p>Встановлює стиль ліній: <i>T</i> - тип, <i>Pat</i> – вірець, <i>Thick</i> – товщина ліній.</p> <p>Константи типу ліній (<i>T</i>):</p> <ul style="list-style-type: none">SolidLn = 0; {суцільна}DottedLn = 1; {крапки}CenterLn = 2; {штрих-пунктирна}DashedLn = 3; {пунктирна}UserBitLn = 4; {визначена користувачем} <p><i>Pat</i> враховується лише для ліній, визначених користувачем (16 біт задають 16 пікселів і формують тип лінії).</p> <p><i>Thick</i> = {1, 3}.</p> <p>1 - товщина в 1 піксел, 3 - товщина в 3 піксели.</p>



Зображення і стилі

Заповнення та стилі

Назва процедури	Призначення
GetLineStyle (var StyleInfo: LineSettingsType)	Повертає стиль ліній. Type LineSettingsType = record LineStyle : Word; Pattern : Word; Thickness : Word end;
SetWriteMode (Mode)	Встановлює спосіб взаємодії нових ліній з існуючими. Mode = 0 - накладання на існуючі, Mode = 1 - накладання згідно операції XOR. Режим використовується процедурами Drawpoly, Line, LineRel, LineTo, Rectangle.



Мультиплікація

Процедури та функції

Назва	Призначення
ClearDevice	Очистка екрану, курсор переміщується в точку (0,0), екран заповнюється кольором фону
ClearViewport	Очистка вікна (або екрану), курсор переміщується в точку (0,0), вікно заповнюється кольором номер 0 палітри
SetViewport <i>(X1, Y1, X2, Y2, Clip)</i>	Встановлює для графічного виводу поточну область або вікно. (<i>X1, Y1</i>) – лівий верхній, (<i>X2, Y2</i>) – правий нижній кут області перегляду. Курсор переміщується в точку (0,0). <i>Clip</i> : boolean



Мультиплікація

Процедури та функції

Назва	Призначення
GetAspectRatio (X,Y)	Повертає 2 числа для оцінки співвідношення сторін екрану
SetAspectRatio (X,Y)	Встановлює масштабний коефіцієнт відношення сторін графічного екрану
SetActivePage (P)	Активізує сторінку відеопам'яті P (нумерація сторінок з нуля)
SetVisualPage (P)	На екран виводиться сторінка номер P



Вивід текстової інформації

У графічному режимі для виводу тексту використовуються *растрові* і *штрихові шрифти*.

Символ **растрового** шрифту визначається знакоміщем 8x8 елементів.

Штриховий шрифт визначається серією векторів і зберігається в окремому файлі з розширенням **.CHR**.

При збільшенні розміру символів штрихові шрифти мають перевагу над растровими через кращу роздільну здатність та якість.



Вивід текстової інформації

Процедури та функції

Назва	Призначення
OutText (<i>Txt</i>)	Виводить текст, починаючи з позиції курсору
OutTextXY (<i>X, Y, Txt</i>)	Виводить текст, починаючи з координат (<i>X, Y</i>)
SetTextStyle (<i>Font, Direct, Size</i>)	Встановлює стиль текстового виводу на графічний екран. <i>Font</i> - номер шрифту, <i>Direct</i> - код напрямку, <i>Size</i> - код розміру шрифту.



Вивід текстової інформації

Процедури та функції

Назва	Призначення
OutText (<i>Txt</i>)	Виводить текст, починаючи з позиції курсору
OutTextXY (<i>X, Y, Txt</i>)	Виводить текст, починаючи з координат (<i>X, Y</i>)
SetTextStyle (<i>Font, Direct, Size</i>)	Встановлює стиль текстового виводу на графічний екран. <i>Font</i> - номер шрифту, <i>Direct</i> - код напрямку, <i>Size</i> - код розміру шрифту.



Вивід текстової інформації

Font

Const

DefaultFont	=0	растровий шрифт	
TriplexFont	=1	Потрійний шрифт	(TRIP.CHR)
SmallFont	=2	Зменшений шрифт	(LITT.CHR)
SansSerifFont	=3	Прямий шрифт	(SANS.CHR)
GothicFont	=4	Готичний шрифт	(GOTH.CHR)
	=5	‘рукописний шрифт’	(SCRI.CHR)
	=6	одноштриховий шрифт типу Courier	(SIMP.CHR)
	=7	нахилений шрифт <i>Times Italic</i>	(TSCR.CHR)
	=8	шрифт типу Times Roman	(LCOM.CHR)
	=9	шрифт типу Courier	(EURO.CHR)
	=10	збільшеного розміру великий двоштриховий шрифт	(BOLD.CHR)



Вивід текстової інформації

Direct

Const

HorizDir = 0;

VertDir = 1;

Size

може приймати значення 1..10 .



Вивід текстової інформації

Процедури та функції

Назва	Призначення
SetTextJustify (<i>Horiz, Vert</i>)	<p>Задає вирівнювання тексту відносно курсору (або координат). <i>Horiz</i> – горизонтальне вирівнювання, <i>Vert</i> – вертикальне вирівнювання.</p> <p>Const</p> <p>LeftText = 0; { курсор ліворуч від тексту } CenterText = 1; { симетрично } RightText = 2; { курсор праворуч від тексту } BottomText = 3; { курсор знизу від тексту } TopText = 4; { курсор зверху від тексту }</p>



Вивід текстової інформації

Процедури та функції

Назва	Призначення
SetUserCharSize <i>(X1, Y1, X2, Y2)</i>	Змінює розмір символів згідно заданих пропорцій. Наприклад, щоб збільшити ширину символу двічі, потрібно задати X1=2, X2=1. Стандартний розмір задається процедурою SetTextStyle.
TextWidth (Txt)	Повертає довжину тексту, що виводиться, в пікселях
TextHeight(Txt)	Повертає висоту шрифту в пікселях



Вивід текстової інформації

Процедури та функції

Назва	Призначення
GetTextSettings (var <i>TextInfo</i> : TextSettingsType)	Повертає стиль та вирівнювання тексту Type TextSettingsType = record Font : Word; Direction : Word; CharSize : Word; Horiz : Word; Vert : Word; end;



Обробка помилок

Внутрішні помилки модуля Graph повертаються функцією **GraphResult**. Функція повертає код помилки, котрий вказує на стан, який виник після останньої графічної операції.

Визначені такі коди помилок:

grOk	= 0;	{ немає помилок }
grInitGraph	= - 1;	{ не ініційовано графічний режим }
grNotDetected	= - 2;	{ не визначено тип драйвера }
grFileNotFind	= - 3;	{ не знайдено графічний драйвер }
grInvalidDriver	= - 4;	{ неправильний тип драйвера }
grNoLoadMem	= - 5;	{ немає пам'яті для завантаження драйвера }
grNoScanMem	= - 6;	{ переповнення пам'яті при скануючому заповненні }
grNoFloodMem	= - 7;	{ переповнення пам'яті при заповненні }
grFontNotFound	= - 8;	{ не знайдено файл зі шрифтом }



Обробка помилок

grFontNotMem	= - 9;	{ немає пам'яті для розміщення шрифту }
grInvalidMode	= - 10;	{ неправильний графічний режим }
grError	= - 11;	{ загальна помилка }



Дякую за увагу!
