

Урок 1. Процедури і функції модуля Graph (для роботи з пікселями, колом, дугами)

Мета: навчити встановлювати графічний екран дисплея та використовувати існуючі засоби (методи) зображення графічної інформації (процедури модуля Graph), формувати абстрактне мислення та просторові уявлення, розвивати творчі здібності, виховувати комп'ютерну грамотність, акуратність підчас використання малюнків та креслень.

Обладнання: персональні комп'ютери, опорний конспект до заданої теми, роздатковий матеріал.

Тип уроку: урок формування умінь і навичок.

Форма організації уроку.

Робота в комп'ютерному класі – лекційно - практичне заняття;

- пояснення нового матеріалу;
- парна та індивідуальна робота за комп'ютером.

Хід уроку

1. Актуалізація опорних знань.

Стандартний стан комп'ютера після його вмикання, а також до моменту запуску програми із середовища Турбо Паскаля відповідає роботі екрану дисплея в текстовому режимі. Для роботи програми в графічному режимі треба виконати *ініціалізацію (встановлення)* графічного режиму.

Після завершення роботи програми в графічному режимі комп'ютер повертається в текстовий режим.

Питання до учнів:

1. Скільки символів у рядку і в стовпці містить екран дисплея у текстовому режимі?
2. За допомогою яких процедур можна виконати найпростіші графічні побудови в текстовому режимі?
3. Який модуль містить ці процедури?
4. Як під'єднати модуль Crt ?
5. Як виконати вставляння символу псевдографіки в текст?

2. Пояснення нового матеріалу.

Система координат у графічному режимі не відповідає системі координат текстового режиму (80x25). Графічний екран дисплея складається з точок, які можна засвічувати певним кольором чи гасити, в результаті чого на екрані утворюється деяке зображення. Точки називаються **пікселями**. Розташування точок визначається їхніми координатами. Можливе задання різних графічних режимів(розміри екрану) для кожного з **адаптерів-електронної плати**.

Тому будь-яка програма, яка використовує графічні можливості комп'ютера повинна *ініціалізувати (встановлювати) графічний режим роботи дисплейного адаптера* .

Налаштування процедур на роботу з конкретним адаптером досягається за допомогою підключення потрібного графічного драйвера. Драйвер – це спеціальна програма, яка здійснює

управління тими чи іншими технічними засобами комп'ютера.

Графічний драйвер управляє дисплейним адаптером в графічному режимі.

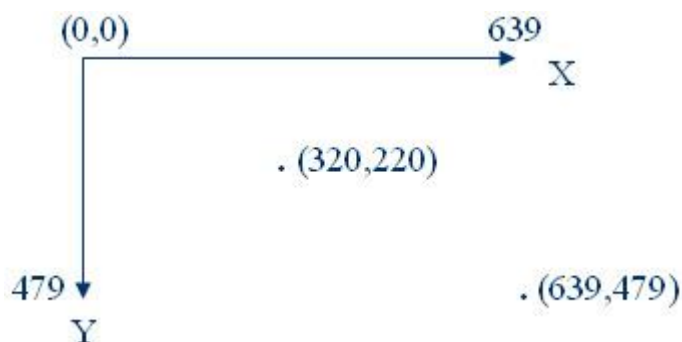
Графічні драйвери розроблені фірмою Borland для всіх типів адаптерів. Вони знаходяться на диску в вигляді файлу з розширенням BGI (від англ. Borland Graphics Interface – графічний інтерфейс фірми Borland).

Н-д: CGA.BGI, EGA.VGA.BGI – драйвер для адаптерів EGA і VGA.

Для кожного із адаптерів можливе задання різних графічних режимів (розміри екрану).

Для VGA адаптера розміри екрану в залежності від режиму екрану можуть бути наступні: 640x200, 640x350, 640x480.

Розглянемо екран, який має 640 точок у горизонтальному напрямку (X) і 480 у вертикальному (Y). Початок відліку точок є у лівому верхньому куті екрана.



Для виводу графічних зображень на екран на початку програми слід приєднати командою **Uses** до програми модуль **Graph**, який містить більш 50 графічних процедур і функцій. (*Uses Graph*)

Потрібні файли: Turbo. tpl, Graph.tpu, Egavga.bgi.

Робота програми починається з ініціалізації (встановлення) графічного режиму процедурою **InitGraph** і завершується процедурою **CloseGraph**

Розглянемо і проаналізуємо програму: (роздатковий матеріал).

```
USES Graph
VAR
  GraphDriver:integer; { змінна, що задає тип адаптера }
  GraphMode:integer; { змінна, що задає режим роботи адаптера( екрану) }
BEGIN
  {автоматичне визначення типу адаптера}
  GraphDriver:= Detect;
  { автоматичний вибір найкращого режиму екрану (640 x 480) }
  InitGraph( GraphDriver,GraphMode, 'C:TP7' );
  { 'C:TP7' – може мати вид '' , якщо драйвери в активному каталозі }
  if GraphResult < > 0 then
    begin
      writeln('Помилка графіки');
      Halt; { Стоп }
    End;
  { GraphResult - ф-я, повертає код останньої виконаної графічної операції }
  { набуває значення (-14) - 0, якщо 0-помилки немає }
  { Текст програми }

      Setbkcolor(0);                               {колір фону }
```

```

Setfillstyle(1,cyan);Bar(0,0,639,479); {зафарбований прямокутник }
SetColor(black); { колір майбутнього зображення}
Rectangle(50,100,550,300); { прямокутник}
CloseGraph { Закриття графічного режиму. }

```

End.

Починається програма з виклику процедури InitGraph, яка автоматично виявляє апаратні засоби і завантажує відповідний графічний драйвер поміщений в Tr7. Якщо в процесі завантаження пройшла помилка, то на екрані появляється повідомлення про помилку.

CloseGraph - в процесі свого виконання ця процедура очищає пам'ять яка була задіяна під драйвери, шрифти і проміжкові дані і відновлює той режим адаптера, який був до ініціалізації графічного режиму.

Розглянемо процедури модуля Graph призначені для графічних побудов. Всі процедури, які ми будемо вивчати для зручності користування розділемо на декілька груп.

I. Стандартні процедури і функції модуля Graph для роботи з кольорами.

SetBkColor (колір) - задання кольору фону;

SetColor (колір) - задання кольору майбутнього зображення;

SetFillStyle (стиль заповнення, колір) - задання способу заповнення замкнутої області:
Стиль заповнення

0 - кольором фону	EmptyFill
1 - суцільне кольором зображення	SolidFill
2 - горизонтальними лініями	LineFill
3 - похилими лініями	LtslashFill
4 - похилими товстими лініями	SlashFill
5 - похилими зворотніми товстими лініями	BkslashFill
6 - похилими зворотніми лініями	LtbkslashFill
7 - прямокутною горизонтальною штриховкою	HatchFill
8 - косою штриховкою	XhatchFill
9 - косою перекриваючою штриховкою	InterleaveFill
10 - рідкими крапками	WideDotFill
11 - щільне заповнення крапками	CloseDotFill

Колір

0 - чорний	Black
1 - синій	Blue
2 - зелений	Green
3 - блакитний	Cyan
4 - червоний	Red
5 - фіолетовий	Magenta
6 - коричневий	Brown
7 - світло-сірий	LightGray
8 - темно-сірий	DarkGray
9 - яскраво-синій	LightBlue
10 - яскраво-зелений	LightGreen
11 - яскраво-блакитний	LightCyan

12 - яскраво-червоний	LightRed
13 - яскраво-фіолетовий	LightMagenta
14 - жовтий	Yellow
15 - білий	White

FloodFill (x,y, колір межі) - заповнення замкненої області з внутрішньою точкою (x,y) ;
Приклад:

```
SetColor(2);
Circle(320,220,100);
SetFillStyle(1,6);
FloodFill(320,220,2);
```

CleaDevice – процедура, яка очищає екран зафарбовуючи його кольором фону

GetDevice – функція, яка повертає максимально допустимі номени кольору для заданого режиму

Чорний колір !!!!!!!!!!! Чорний колір набуває значення кольору фону!!!!!!

Існує помилкове твердження, що не можна контури зображення малювати чорним кольором, його просто не видно!

Щоб цього недоліку уникнути в програмі або не використовують процедуру

SetBkColor (колір), а зафарбовують весь екран дисплея прямокутником потрібного кольору, або використовують **SetBkColor (Black)**, з параметром Black, а потім малюють прямокутник на весь екран потрібного кольору.

Приклад програми, що демонструє можливі кольори (роздатковий матеріал):

```
USES Graph;
VAR
    GD:integer; {тип адаптера}
    GM:integer; {режим екрану}
BEGIN
    GD:= Detect; {автоматичне визначення типу адаптера}
    InitGraph(GD,GM,''); {автоматичний вибір режиму екрану}
    if GraphResult <> 0 then Halt; {GraphResult - повертає код останньої}
    виконаної графічної операції }
    For i:=0 to GetMaxColor do
        Begin
            SetBkColor(i);
            ClearDevice;
            Readln;
        End;
    CloseGraph;
END.
```

3. Закріплення матеріалу. Робота за комп'ютерами

Завдання 1 Виконати програму, яка демонструє можливі кольори графічного режиму, модифікувати дану програму, використовуючи генератор випадкових чисел і озвучення кожного кольору іншою нотою.

```
USES Graph, Crt;
VAR
```

```
GD:integer;
GM:integer;
BEGIN
GD:= Detect;
InitGraph(GD,GM, '');
if GraphResult <> 0 then Halt;

Randomize;
For i:=1 to 20 do
  Begin
    SetBkColor (random(i));
    Sound(I*5);
    Delay(2000);
    Nosound;
    ClearDevice;
  End;
CloseGraph;

END.
```

4. Підведення підсумків

Питання до класу:

1. Яка процедура встановлює графічний режим?
2. Як відрізняється система координат в текстовому і в графічному режимі?
3. Яка процедура закриває графічний режим?
4. Назвати процедуру для задання кольору фону.
5. Назвати процедуру для задання кольору майбутнього зображення.
6. Яка відмінність між процедурами SetFillStyle і FloodFill ?

5. Домашнє завдання

- Вивчити всі процедури розглянуті на уроці.