

*ADNSU*

*Fakültə : İnformasiya texnologiyaları və  
idarəetmə*

*İxtisas : Kompüter elmləri*

*Qrup : 647.6*

---

*Müəllimə : Kərimova Sevinc*

*Tələbə : Mehdiyeva Sevinc*

*Mövzu : Qrafiki rejim. Sadə qrafiki  
proqramlar.*

# Qrafiki rejim

## Sadə qrafiki proqram



C dilində qrafiki proqram «sendviç» strukturuna bənzəyir.



Şəkillərin çəkilməsi üçün xüsusi pəncərə açan sadə program tərtib edək:

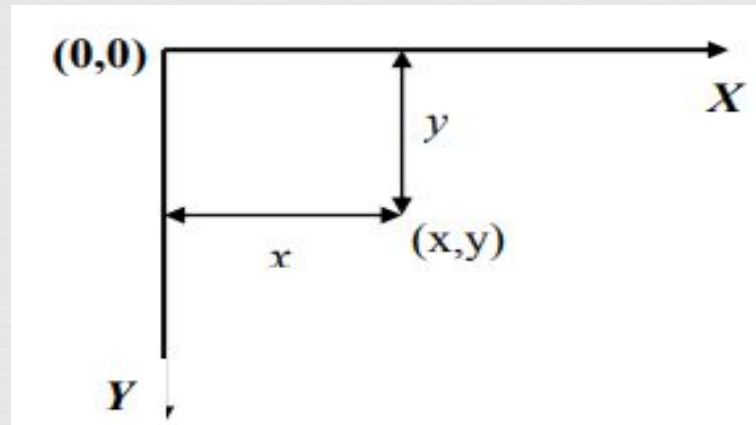
```
#include <graphics.h>
#include <conio.h>
main ()
{
initwindow ( 400, 300 ); // 400x300 ölçüdə qrafiki pəncərənin açılması
// .... burada şəkil çəkmək olar
getch ();
closegraph (); // pəncərəni bağla
}
```

- ❖ Qrafiki funksiyalardan istifadə etmək **graphics.h** başlıq faylını qoşmaq lazımdır.
- ❖ **Initwindow** funksiyası şəkil çəkmək üçün əlavə pəncərə açır. Mötərizədə piksellərlə pəncərənin enini və hündürlüyünü göstərmək lazımdır.
- ❖ **Closegraph** funksiyasının köməyi ilə qrafiki rejim dayandırılır.

# Nöqtələrin koordinatları:

Ekranı şəkil çəkmək üçün koordinatları təyin etməyi bacarmaq lazımdır.

- Koordinatların başlanğıcı, yəni  $(0,0)$  nöqtəsi pəncərənin sol yuxarı küncündə yerləşir.
- Riyazi koordinat sistemindən fərqli olaraq **X** oxu sağa, **Y** oxu isə aşağı istiqamətləndirilib.
- İstənilən nöqtə üçün **x** koordinatı – pəncərənin sol, **y** koordinatı isə yuxarı sərhədinə kimi məsafədir.



# □ RƏNG



16 standart rəng üçün ədədi və simvolik işarələr təyin olunub:

|   |           |           |    |              |                |
|---|-----------|-----------|----|--------------|----------------|
| 0 | BLACK     | qara      | 8  | DARKGRAY     | tünd boz       |
| 1 | BLUE      | göy       | 9  | LIGHTBLUE    | açıq göy       |
| 2 | GREEN     | yaşıl     | 10 | LIGHTGREEN   | açıq yaşıl     |
| 3 | CYAN      | mavi      | 11 | LIGHTCYAN    | açıq mavi      |
| 4 | RED       | qırmızı   | 12 | LIGHTRED     | açıq qırmızı   |
| 5 | MAGENTA   | bənövşəyi | 13 | LIGHTMAGENTA | açıq bənövşəyi |
| 6 | BROWN     | qəhvəyi   | 14 | YELLOW       | sarı           |
| 7 | LIGHTGRAY | açıq boz  | 15 | WHITE        | ağ             |

Bundan əlavə, bütün rəng palitrasından istifadə etmək olar (True Color rejimi, təbii rəng). Bu halda istənilən rəng 3 komponentdən ibarət olur: qırmızı (**R**), yaşıl (**G**) və göy (**B**). Rəngin hər komponenti 0 – 255 aralığında dəyişən tam ədəddir (cəmi 256 variant). Beləliklə RBG kombinasiyası üçün  $256^3=16777216$  rəng alınır. Rəngləri vermək üçün **COLOR** funksiyasından istifadə edilir. Bu funksiyanın 3 parametri var. Onlar **R**, **G**, **B** rənglərinin intensivliyini təyin edirlər və mötərizədə vergüllə ayrılmış yazılırlar. Məsələn, bəzi rəngləri belə təyin etmək olar:

|                       |           |
|-----------------------|-----------|
| COLOR (0,0,0)         | qara      |
| COLOR (255,0,0)       | qırmızı   |
| COLOR (0, 255, 0)     | yaşıl     |
| COLOR (0, 0, 255)     | göy       |
| COLOR (255, 255, 255) | ağ        |
| COLOR (100, 100, 100) | boz       |
| COLOR (255, 0, 255)   | bənövşəyi |

**setcolor** funksiyası xətlərin rəngini təyin edir:

```
setcolor ( 10 ); // açıq yaşıl rəngi təyin etmək
```

Bu əmrdən sonra çəkilən bütün xətlər, düzbucaqlar, çevrələr və s. açıq yaşıl rəngli olacaqdır. Bu funksiya rəngi rəng palitrasından da götürə bilər:

```
setcolor ( COLOR(255, 0, 255) ); // bənövşəyi rəngi təyin etmək
```

Şəkil çəkmək üçün açılmış pəncərə ağ rəngdə olur.

## **Ayrı-ayrı piksellərlə işləmə:**

Şəkil çəkmək üçün standart funksiyalardan istifadə edilir. **putpixel** funksiyasından istifadə edərək hər piksel üçün onun rəngini vermək olar:

```
putpixel ( x, y, 14 ); // (x,y) nöqtəsini sarı rəngləmək
```

**getpixel** funksiyası vasitəsi ilə pəncərədə yerləşən istənilən pikselin rəngini təyin etmək olar.

```
n = getpixel ( x, y ); // (x,y) nöqtəsinin rəngini n dəyişənə yazmaq
```

# Xətlər

Parçanı **line** əmri vasitəsi ilə çəkmək olar:

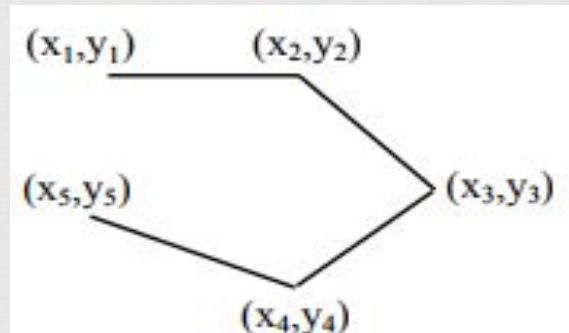
```
line ( x1, y1, x2, y2 ); // (x1,y1)-(x2,y2) parçası
```

Parçanı başqa üsulla çəkmək olar. Əvvəlcə **moveto** əmrindən istifadə edərək kursoru **(x1,y1)** nöqtəsinə gətirmək, sonra isə **lineto** əmri vasitəsi ilə **(x2,y2)** nöqtəsinə qədər parça çəkmək:

```
moveto ( x1, y1 ); // kursor (x1,y1) nöqtəsinə gətirilir  
lineto ( x2, y2 ); // (x2,y2) nöqtəyə qədər xətt çəkilir
```

**lineto** əmrindən sonra kursor növbəti **(x2,y2)** nöqtəsinə keçir. Qırıq xətlərin çəkilməsində bu əmrdən istifadə etmək daha əlverişlidir:

```
setcolor (12); // qırmızı rəng  
moveto (x1, y1); // kursoru (x1,y1) nöqtəsinə gətir  
lineto (x2, y2); // 2-ci nöqtəyə qədər parça çəkilir  
lineto (x3, y3); // 3-cü nöqtəyə qədər parça çəkilir  
lineto (x4, y4); // .....  
lineto (x5, y5);
```





# Düzbucaqlar

Düzbucağı çəkmək üçün iki qarşı bucağın koordinatlarını vermək lazımdır (adətən sol yuxarı və sağ aşağı bucaqlar seçilir). Konturun rəngi **setcolor** funksiya vasitəsi ilə təyin olunur, düzbucağın özü isə **rectangle** əmri vasitəsi ilə çəkilir.

```
setcolor ( 9 );  
rectangle (x1, y1, x2, y2);
```

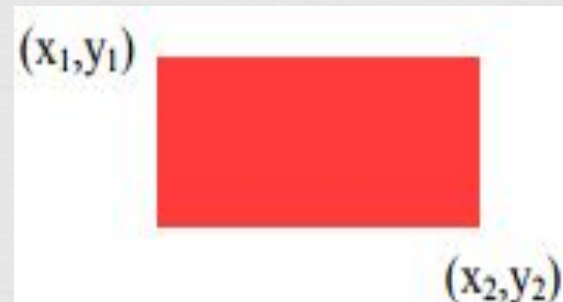


Rəngli düzbucağı çəkmək üçün **bar** əmrindən istifadə edilir. Rəngi və rəng çəkilməsinin üslubunu (ing. fill style) təyin etmək üçün **setfillstyle** funksiyasından istifadə etmək lazımdır.

```
setfillstyle ( 1, 12 ); // 1-ci üslub, 12-ci rəng  
bar (x1, y1, x2, y2);
```

**setfillstyle** funksiyanın 2 parametri var. Birinci parametr rəng çəkmə üslubunu, ikinci parametr isə rəngi təyin edir.

|         |  |                  |
|---------|--|------------------|
| 0       |  | rəng yoxdur      |
| 1       |  | bircinsli rəng   |
| 3,4,5,6 |  | əyri xətlər      |
| 7,8     |  | tor              |
| 9,10,11 |  | nöqtəli naxışlar |



# Çevrə



Çevrəni çəkmək üçün **circle** funksiyasından istifadə edirlər:

```
setcolor ( COLOR (0, 255, 0) ); // yaşıl rəng  
circle (x, y, R );
```

circle funksiyasının üç parametri var: birinci iki parametr mərkəzin koordinatlarıdır, üçüncü parametr isə - radiusdur. Parametrlər kimi ədədlər, dəyişənlərin adları və ya cəbri ifadələr ola bilər:

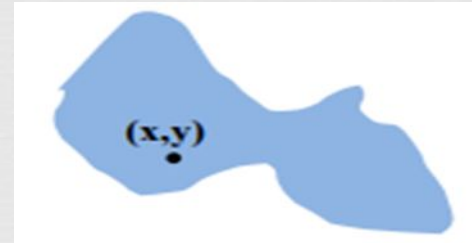
```
circle (200, y0+25, R );
```

# İxtiyari oblastın rənglənməsi:



Bəzən ixtiyari çəkilmiş fiquru rəngləmək lazım gəlir. Bunu **floodfill** funksiyası vasitəsi ilə etmək olar:

```
setfillstyle ( 1 , 11 ); // üslub 1, rəng 11  
floodfill (x, y, 0); // 0 rənginin sərhədinə kimi rəngləmək
```



Fiquru rəngləmək üçün onun daxilində istənilən nöqtənin  $(x,y)$  koordinatlarını bilmək lazımdır. Bundan başqa, fiqurun sərhədi bir rəngli və kəsilməz olmalıdır. Sərhədin rəngi **floodfill** funksiyasının axırıncı parametridir.

# Yazılar



**outtextxy** funksiyası pəncərənin istənilən yerində yazını çıxartmağa imkan verir. Bu funksiyaya yazının yuxarı sol küncünün (x,y) koordinatlarını vermək lazımdır. Yazının rəngi **setcolor** funksiyası ilə tənzimlənir:

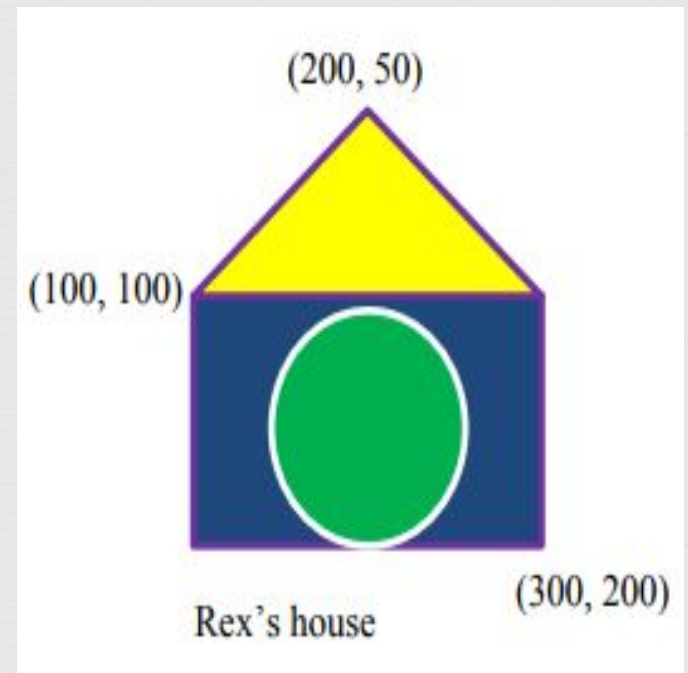
```
setcolor ( 9 );  
outtextxy ( x, y, "Bakı" );
```

(x,y)  
•  
**Bakı**

# Proqram nümunəsi

Standart qrafiki funksiyalardan istifadə edərək evcik çəkən bir proqram yazıq.

```
#include <graphics.h>
#include <conio.h>
main ()
{
    initwindow (440, 300);
    setfillstyle (1,9);
    bar (100, 100, 300, 200); // göy düzbucaq
    setcolor (13);           // bənövşəyi kontur
    rectangle (100, 100, 300, 200);
    moveto (100, 100);     // damın çəkilməsi
    lineto (200,50);
    lineto (300, 100);
    setfillstyle (1,14);    // dam sarı rəngdə
    floodfill (200, 75, 13);
    setcolor (15);
    circle (200, 150, 50); // ağ çevrə
    setfillstyle (1, 10);
    floodfill (200, 150, 150); // çevrənin içi yaşıl
    setcolor (12);
    outtextxy (100, 230, "Rex's house");
    getch();
    closegraph();
}
```



*Diqqətinizə görə  
təşəkkürlər!*

