Работа с графикой на языке Assembler.

Задание:

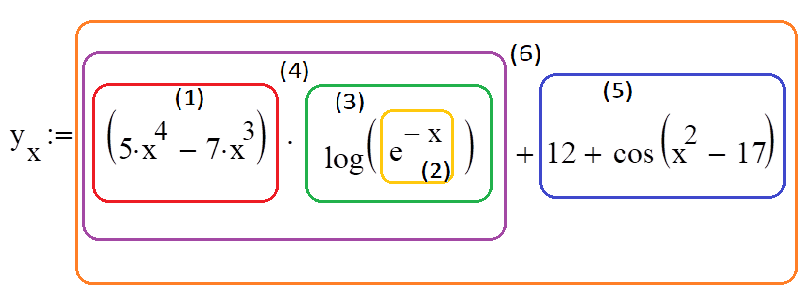
1) 

2) 

3) При щелчке ПКМ запоминается цвет объекта по которому щелкнули, при щелчке ЛКМ этот цвет применяется к объекту

**Задание 1.**

**Модель:**

****

**Код:**

;-----------

choise\_color db "Выберите цвет графика:",0

pen\_color dd 00000000h

button\_color dw ?

color\_button\_black db "Черный",0

color\_button\_blue db "Голубой",0

color\_button\_green db "Зеленый",0

color\_button\_red db "Красный",0

color\_button\_yellow db "Желтый",0

color\_button\_gray db "Серый",0

;-----------

;

color\_button\_id1 equ 2

color\_button\_id2 equ 3

color\_button\_id3 equ 4

color\_button\_id4 equ 5

color\_button\_id5 equ 6

color\_button\_id6 equ 7

Black equ 0000000h

Blue equ 0FF0000h

Green equ 000FF00h

Red equ 00000FFh

Yellow equ 000FFFFh

Gray equ 0080808h

;

;------------------------{

invoke CreateWindowEx,WS\_EX\_LEFT, ADDR STATIC,ADDR choise\_color,\

WS\_CHILD or WS\_VISIBLE,\

220,5,200,25,hWnd,NULL,hInstance,NULL

invoke CreateWindowEx,NULL, ADDR ButtonClassName,ADDR color\_button\_black,\

WS\_CHILD or WS\_VISIBLE or BS\_DEFPUSHBUTTON,\

220,35,100,25,hWnd,color\_button\_id1,hInstance,NULL

invoke CreateWindowEx,NULL, ADDR ButtonClassName,ADDR color\_button\_blue,\

WS\_CHILD or WS\_VISIBLE or BS\_DEFPUSHBUTTON,\

220,65,100,25,hWnd,color\_button\_id2,hInstance,NULL

invoke CreateWindowEx,NULL, ADDR ButtonClassName,ADDR color\_button\_green,\

WS\_CHILD or WS\_VISIBLE or BS\_DEFPUSHBUTTON,\

220,95,100,25,hWnd,color\_button\_id3,hInstance,NULL

invoke CreateWindowEx,NULL, ADDR ButtonClassName,ADDR color\_button\_red,\

WS\_CHILD or WS\_VISIBLE or BS\_DEFPUSHBUTTON,\

330,35,100,25,hWnd,color\_button\_id4,hInstance,NULL

invoke CreateWindowEx,NULL, ADDR ButtonClassName,ADDR color\_button\_yellow,\

WS\_CHILD or WS\_VISIBLE or BS\_DEFPUSHBUTTON,\

330,65,100,25,hWnd,color\_button\_id5,hInstance,NULL

invoke CreateWindowEx,NULL, ADDR ButtonClassName,ADDR color\_button\_gray,\

WS\_CHILD or WS\_VISIBLE or BS\_DEFPUSHBUTTON,\

330,95,100,25,hWnd,color\_button\_id6,hInstance,NULL

;------------------------}

.ELSEIF uMsg==WM\_COMMAND

mov eax,wParam

.IF ax==LINE\_RB

mov flg,0

.ELSEIF ax==POINT\_RB

mov flg,1

.ELSE

.IF ax==ButtonID

shr eax,16

.IF ax==BN\_CLICKED

invoke GetWindowText,hwndEditMinX,ADDR buffer,25

invoke StrToInt, ADDR buffer

mov minX,eax

invoke GetWindowText,hwndEditMaxX, ADDR buffer,25

invoke StrToInt, ADDR buffer

mov maxX,eax

invoke GetWindowText,hwndEditStep,ADDR buffer,25

invoke StrToFloat, ADDR buffer, ADDR step

;Создаем доп. окно

invoke CreateWindowEx,WS\_EX\_CLIENTEDGE,ADDR ClassName1,ADDR AppNam1e,\

WS\_OVERLAPPEDWINDOW,CW\_USEDEFAULT,\

CW\_USEDEFAULT,800,600,0,NULL,\

hInstance,NULL

mov hwnd1,eax

invoke ShowWindow, hwnd1, SW\_SHOWNORMAL

invoke UpdateWindow, hwnd1

.ENDIF

.ENDIF

;---------{

.IF ax==color\_button\_id1

shr eax,16

.IF ax==BN\_CLICKED

mov pen\_color,Black

.ENDIF

.ENDIF

.IF ax==color\_button\_id2

shr eax,16

.IF ax==BN\_CLICKED

mov pen\_color,Blue

.ENDIF

.ENDIF

.IF ax==color\_button\_id3

shr eax,16

.IF ax==BN\_CLICKED

mov pen\_color,Green

.ENDIF

.ENDIF

.IF ax==color\_button\_id4

shr eax,16

.IF ax==BN\_CLICKED

mov pen\_color,Red

.ENDIF

.ENDIF

.IF ax==color\_button\_id5

shr eax,16

.IF ax==BN\_CLICKED

mov pen\_color,Yellow

.ENDIF

.ENDIF

.IF ax==color\_button\_id6

shr eax,16

.IF ax==BN\_CLICKED

mov pen\_color,Gray

.ENDIF

.ENDIF

;---------}

.ENDIF

.ELSE

invoke DefWindowProc,hWnd,uMsg,wParam,lParam

ret

.ENDIF

xor eax,eax

ret

WndProc endp

count proc

;(первая скобка) (1)

mov [tmp], 5

fld x

fmul x

fmul x

fmul x

fimul tmp

mov [tmp], 7

fld x

fmul x

fmul x

fimul tmp

fsubp

;e^-x

fld x

fchs

fldl2e

fmul

fld st

frndint

fsub st(1), st

fxch st(1)

f2xm1

fld1

fadd

fscale

fstp st(1)

;log(e^-x) (2)

fldlg2

fxch

fyl2x

;(1)\*(2) (3)

fmulp

;12+cos(x^2-17) (4)

fld x

fmul x

mov [tmp], 17

fisub tmp

fcos

mov [tmp], 12

fiadd tmp

;(3) \* (4)

faddp

;Считаем X и Y, как координату пикселя

fmul ScaleY

fild OffsetY

fsubr

fistp intY

fld x

fmul ScaleX

fiadd OffsetX

fistp intX

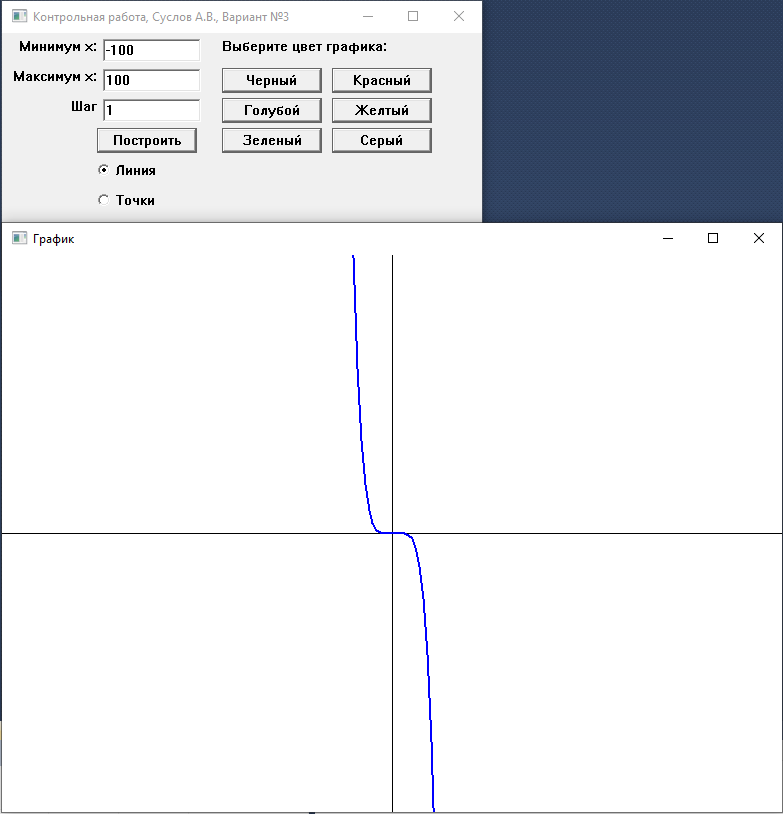
ret

count endp

@enddd:

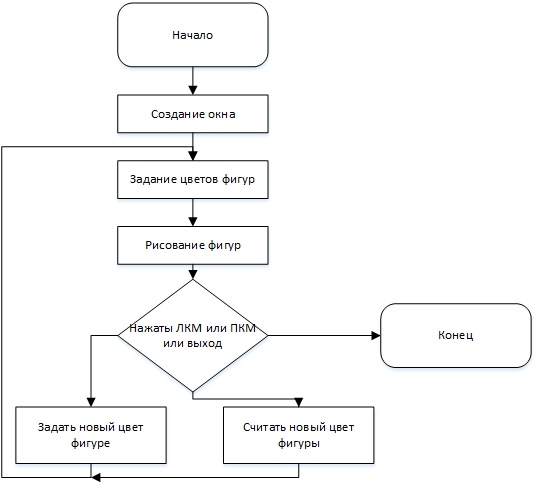
end start

**Демонстрация:**



**Задание 2,3**

**Модель:**



**Код:**

WndProc proc hWnd:HWND, uMsg:UINT, wParam:WPARAM, lParam:LPARAM

LOCAL hOld:HDC

LOCAL hMemDC: HDC

LOCAL hBmp: HDC

LOCAL ps:PAINTSTRUCT

LOCAL rc:RECT

LOCAL hdc:HDC

.IF uMsg==WM\_DESTROY

invoke PostQuitMessage, NULL

ret

.ELSEIF uMsg==WM\_CREATE

;1 треугольник

mov polygon1.x, 150

mov polygon1.y, 150

mov polygon1[type POINT].x, 350

mov polygon1[type POINT].y, 150

mov polygon1[2\*type POINT].x, 350

mov polygon1[2\*type POINT].y, 50

;второй треугольник

mov polygon2.x, 150

mov polygon2.y, 350

mov polygon2[type POINT].x, 350

mov polygon2[type POINT].y, 350

mov polygon2[2\*type POINT].x, 350

mov polygon2[2\*type POINT].y, 450

.ELSEIF uMsg==WM\_PAINT

invoke GetClientRect, hWnd, ADDR rc

invoke BeginPaint, hWnd, ADDR ps

mov hdc, eax

;Контекст устройства в памяти

invoke CreateCompatibleDC, hdc

mov hMemDC, eax

;Bitmap для рисования

invoke CreateCompatibleBitmap, hdc, rc.right, rc.bottom

mov hBmp, eax

;Присоединяем bitmap к DC

invoke SelectObject, hMemDC, hBmp

mov hOld, eax

;Заливка рабочей области окна

invoke FillRect, hMemDC, ADDR rc, 0

;board

invoke SelectObject, hMemDC, color\_taken

invoke Rectangle, hMemDC, 5, 5, 20, 20

invoke SelectObject, hMemDC, color\_board

invoke Rectangle, hMemDC, 25, 25, 680, 670

;Рисуем фигуры

invoke SelectObject, hMemDC, color\_border

invoke SelectObject, hMemDC, color\_rectangle\_3

invoke Rectangle, hMemDC, 400, 300, 600, 480

invoke SelectObject, hMemDC, color\_first\_triangle

invoke Polygon, hMemDC, ADDR polygon1, 3

invoke SelectObject, hMemDC, color\_second\_triangle

invoke Polygon, hMemDC, ADDR polygon2, 3

invoke SelectObject, hMemDC, color\_rectangle

invoke Rectangle, hMemDC, 50, 150, 150, 350

invoke SelectObject, hMemDC, color\_quadr

invoke Rectangle, hMemDC, 150, 150, 350, 350

invoke SelectObject, hMemDC, color\_circle

invoke Ellipse, hMemDC, 250, 150, 450, 350

invoke SelectObject, hMemDC, color\_circle\_2

invoke Ellipse, hMemDC, 350, 150, 650, 450

invoke SelectObject, hMemDC, color\_rectangle\_2

invoke Rectangle, hMemDC, 500, 60, 575, 350

invoke SelectObject, hMemDC, color\_circle\_3

invoke Ellipse, hMemDC, 300, 400, 500, 650

invoke SelectObject, hMemDC, color\_rectangle\_4

invoke Rectangle, hMemDC, 60, 500, 620, 650

;Рисуем палитру

invoke SelectObject, hMemDC, color\_palitra\_1

invoke Rectangle, hMemDC, 700, 50, 750, 100

invoke SelectObject, hMemDC, color\_palitra\_2

invoke Rectangle, hMemDC, 700, 110, 750, 160

invoke SelectObject, hMemDC, color\_palitra\_3

invoke Rectangle, hMemDC, 700, 170, 750, 220

invoke SelectObject, hMemDC, color\_palitra\_4

invoke Rectangle, hMemDC, 700, 230, 750, 280

invoke SelectObject, hMemDC, color\_palitra\_5

invoke Rectangle, hMemDC, 700, 290, 750, 340

invoke SelectObject, hMemDC, color\_palitra\_6

invoke Rectangle, hMemDC, 700, 350, 750, 400

invoke SelectObject, hMemDC, color\_palitra\_7

invoke Rectangle, hMemDC, 700, 410, 750, 460

invoke SelectObject, hMemDC, color\_palitra\_8

invoke Rectangle, hMemDC, 700, 470, 750, 520

invoke SelectObject, hMemDC, color\_palitra\_9

invoke Rectangle, hMemDC, 700, 530, 750, 580

invoke SelectObject, hMemDC, color\_palitra\_10

invoke Rectangle, hMemDC, 700, 590, 750, 640

;

;Двойная буферизация

invoke BitBlt, hdc, 0, 0, rc.right, rc.bottom, hMemDC, 0, 0, SRCCOPY

invoke SelectObject, hMemDC, hOld

invoke DeleteObject, hBmp

invoke DeleteDC, hMemDC

invoke EndPaint, hdc, ADDR ps

;-----------------------

.ELSEIF uMsg==WM\_RBUTTONDOWN

mov eax,lParam

mov posx,ax

shr eax,16

mov posy,ax

.IF wParam==MK\_RBUTTON ;левая кнопка мыши

.IF posx>150 ;quadr

.IF posx<350

.IF posy>150

.IF posy<350

mov eax,[color\_quadr]

mov [color\_taken],eax

mov posx,0

.ENDIF

.ENDIF

.ENDIF

.ENDIF

;|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

.ELSEIF uMsg==WM\_LBUTTONDOWN ;правая кнопка мыши

mov eax,lParam

mov posx,ax

shr eax,16

mov posy,ax

.IF wParam==MK\_LBUTTON

.IF posx>150 ;quadr

.IF posx<350

.IF posy>150

.IF posy<350

mov eax,[color\_taken]

mov [color\_quadr],eax

mov posx,0

.ENDIF

.ENDIF

.ENDIF

.ENDIF

invoke InvalidateRect, hWnd, 0, FALSE

.ENDIF

.ELSE

invoke DefWindowProc, hWnd, uMsg, wParam, lParam

ret

.ENDIF

xor eax, eax

ret

WndProc endp

**Демонстрация:**

