

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**«Вятский государственный университет»**  
**(«ВятГУ»)**

Факультет автоматики и вычислительной техники  
Кафедра электронных вычислительных машин

Отчет по лабораторной работе №3  
по дисциплине «Интерфейсы периферийных устройств»

Выполнил студент группы ИВТ-42 \_\_\_\_\_/Рзаев А. Э./

Проверил старший преподаватель кафедры ЭВМ \_\_\_\_\_/Гагарский К. Н./

Киров 2018

## 1 Задание на лабораторную работу

Составить на языке assembler программу вывода на экран монитора графическое изображение в 4-м режиме, состоящее из всех 4 цветов и палитр. Изображение должно иметь возможность двигаться по экрану при нажатии стрелок клавиатуры и менять палитру при нажатии клавиши.

## 2 Теория

Вывод изображения на экран осуществляется в видеорежиме №4 (CGA). Экран рассматривается как совокупность точек (пикселей) – наименьших элементов изображения. Вывод изображения на экран осуществляется путем управления цветом каждой точки экрана.

В этом режиме экран монитора представляет из себя:

-320 точек по горизонтали

-200 точек по вертикали

Вывод пикселя на экран задается путем указания строки в dx, столбца в cx и вызывания прерывания int 10h.

## 3 Исходный код программы

Программа вывода изображения на экран представлена на рисунке 1

```
.286
.model small
.data
    ; первый квадрат
    points dw 45, 15, 105, 15, 1
           dw 45, 15, 45, 75, 2
           dw 105, 15, 105, 75, 3
           dw 45, 75, 105, 75, 3

    ; ребра
    dw 45, 15, 30, 30, 3
    dw 105, 15, 90, 30, 2
    dw 105, 75, 90, 90, 1
    dw 45, 75, 30, 90, 2

    ; второй квадрат
    dw 30, 30, 90, 30, 1
    dw 30, 30, 30, 90, 2
    dw 90, 30, 90, 90, 3
    dw 30, 90, 90, 90, 3
.stack 256h
.code
```

```

; рисование точки
; cdecl
; drawPoint(x, y, c)
; [bp+8] - c, биты точки
; [bp+6] - координата y
; [bp+4] - координата x
drawPoint proc near
    push bp
    mov bp, sp

    pusha
    mov al, byte ptr [bp+8]
    mov ah, 0Ch
    mov cx, word ptr [bp+4]
    mov dx, word ptr [bp+6]
    int 10h
    popa

    pop bp
    ret
drawPoint endp

; рисование линии
; cdecl
; drawLine(x1, y1, x2, y2, c)
; [bp+12] - c
; [bp+10] - y2
; [bp+8] - x2
; [bp+6] - y1
; [bp+4] - x1
drawLine proc near
    push bp
    mov bp, sp
    sub sp, 12
    ;[bp - 2] = e
    ;[bp - 4] = i
    ;[bp - 6] = s1
    ;[bp - 8] = s2
    ;[bp - 10] = f
    ;[bp - 12] = tmp

    pusha
    ; извлечение аргументов
    mov ax, word ptr [bp+4]
    mov bx, word ptr [bp+6]
    mov cx, word ptr [bp+8]
    mov dx, word ptr [bp+10]
    mov si, word ptr [bp+12]

    ; расчет Dx, Dy
    ; cx = Dx
    mov word ptr [bp-6], 1 ; s1 = 1
    sub cx, ax ; Dx = x2 - x1
    cmp cx, 0
    jge skipAbsDx ; Dx >= 0?

```

```

        imul cx, -1          ; Dx = -Dx
        mov word ptr [bp-6], -1 ; s1 = -1
skipAbsDx:
; dx = Dy
mov word ptr [bp-8], 1      ; s2 = 1
sub dx, bx                  ; Dy = y2 - y1
cmp dx, 0
jge skipAbsDy              ; Dy >= 0?
        imul cx, -1          ; Dy = -DY
        mov word ptr [bp-8], -1 ; s1 = -1
skipAbsDy:

; расчет флага f
mov word ptr [bp-10], 0     ; f = 0
cmp dx, cx
jng skipSwapDxDy           ; Dy > Dx?
        xchg dx, cx          ; Dy <=> Dx
        mov word ptr [bp-10], 1 ; f = 1
skipSwapDxDy:
mov word ptr [bp-2], dx     ; e = Dy
mov word ptr [bp-4], 1      ; i = 1

loopCond:
        mov di, cx           ; di = Dx
        add di, dx           ; di += Dy
        cmp word ptr [bp-4], di
        jg loopEnd           ; i <= Dy + Dx?
loopBegin:
        ; putpixel(x, y, c)
        push si ; c
        push bx ; y
        push ax ; x
        call drawPoint
        add sp, 6

; поправка e
mov di, word ptr [bp-2] ; di = e
mov word ptr [bp-12], di ; tmp = di

shl word ptr [bp-12], 1 ; tmp = tmp * 2
cmp word ptr [bp-12], cx
jl ifLess                  ; tmp(2e) < Dx?
jnl ifNotLess
ifLess:
        add word ptr [bp-2], dx ; e += dy
        mov di, word ptr [bp-10] ; di = f
        mov word ptr [bp-12], di ; tmp = f
        cmp word ptr [bp-12], 0
        jnz inc1Y            ; f == 1?
        jz inc1X
inc1Y:
        add bx, word ptr [bp-8] ; y += s2
        jmp endInc1
inc1X:
        add ax, word ptr [bp-6] ; x += s1

```

```

        jmp endInc1
    endInc1:
        jmp endIf1
ifNotLess:
    sub word ptr [bp-2], cx ; e -= dx

    mov di, word ptr [bp-10] ; di = f
    mov word ptr [bp-12], di ; tmp = f
    cmp word ptr [bp-12], 0
    jnz inc2X          ; f == 1?
    jz inc2Y
inc2Y:
    add bx, word ptr [bp-8] ; y += s2
    jmp endInc2
inc2X:
    add ax, word ptr [bp-6] ; x += s1
    jmp endInc2
endInc2:
    jmp endIf1
endIf1:
    add word ptr [bp-4], 1 ; i += 1
    jmp loopCond
loopEnd:

    popa

    add sp, 12
    pop bp

    ret
drawLine endp

; рисуем фигуру
drawFigure proc near
    mov di, 100 ; Dx
    mov dx, 40 ; Dy
    drawLoop1:
        ; рисуем фигуру
        mov cx, 12
        lea bx, points
        mov si, 0

        drawLoop2:
            push word ptr [bx][si+8]
            mov ax, word ptr [bx][si+6]
            add ax, dx
            push ax
            mov ax, word ptr [bx][si+4]
            add ax, di
            push ax
            mov ax, word ptr [bx][si+2]
            add ax, dx
            push ax
            mov ax, word ptr [bx][si+0]
            add ax, di

```

```

        push ax
        call drawLine
        add sp, 10
        add si, 10
        loop drawLoop2

; считывание нажатой клавиши
mov ah, 00h
int 16h

; очистка экрана
push ax
mov cx, 12
lea bx, points
mov si, 0

drawLoop3:
    push 0
    mov ax, word ptr [bx][si+6]
    add ax, dx
    push ax
    mov ax, word ptr [bx][si+4]
    add ax, di
    push ax
    mov ax, word ptr [bx][si+2]
    add ax, dx
    push ax
    mov ax, word ptr [bx][si+0]
    add ax, di
    push ax
    call drawLine
    add sp, 10
    add si, 10
    loop drawLoop3
pop ax

; обработка клавиш
cmp ah, 4Bh ; влево
je decX_
cmp ah, 4Dh ; вправо
je incX_
cmp ah, 48h ; вверх
je decY_
cmp ah, 50h ; вниз
je incY_
cmp ah, 01 ; выход
je drawLoop1End
jmp endInt
incX_:
    inc di
    jmp endInt
decX_:
    dec di
    jmp endInt
incY :

```

```

                inc dx
                jmp endInt
            decY_:
                dec dx
                jmp endInt
            endInt:
                jmp drawLoop1
        drawLoop1End:

        ret
drawFigure endp

main:
    mov ax, @data
    mov ds, ax

    ; переход в графический режим
    mov al, 04h
    mov ah, 00h
    int 10h

    ; установка базового адреса
    mov ax, 0B800h
    mov es, ax

    ; установка палитры
    mov ah, 0Bh
    mov bx, 0100h ; первая палитра
    ;mov bx, 0101h ; вторая палитра
    int 10h

    pusha
    call drawFigure
    popa

    ; переход в текстовый режим
    mov al, 03h
    mov ah, 00h
    int 10h

    ; выход из программы
    mov ax, 4c00h
    int 21h
end main

```

Рисунок 1 – Программа вывода изображения

## 4 Экранные формы

Экранная форма разработанной программы представлена на рисунке 2.

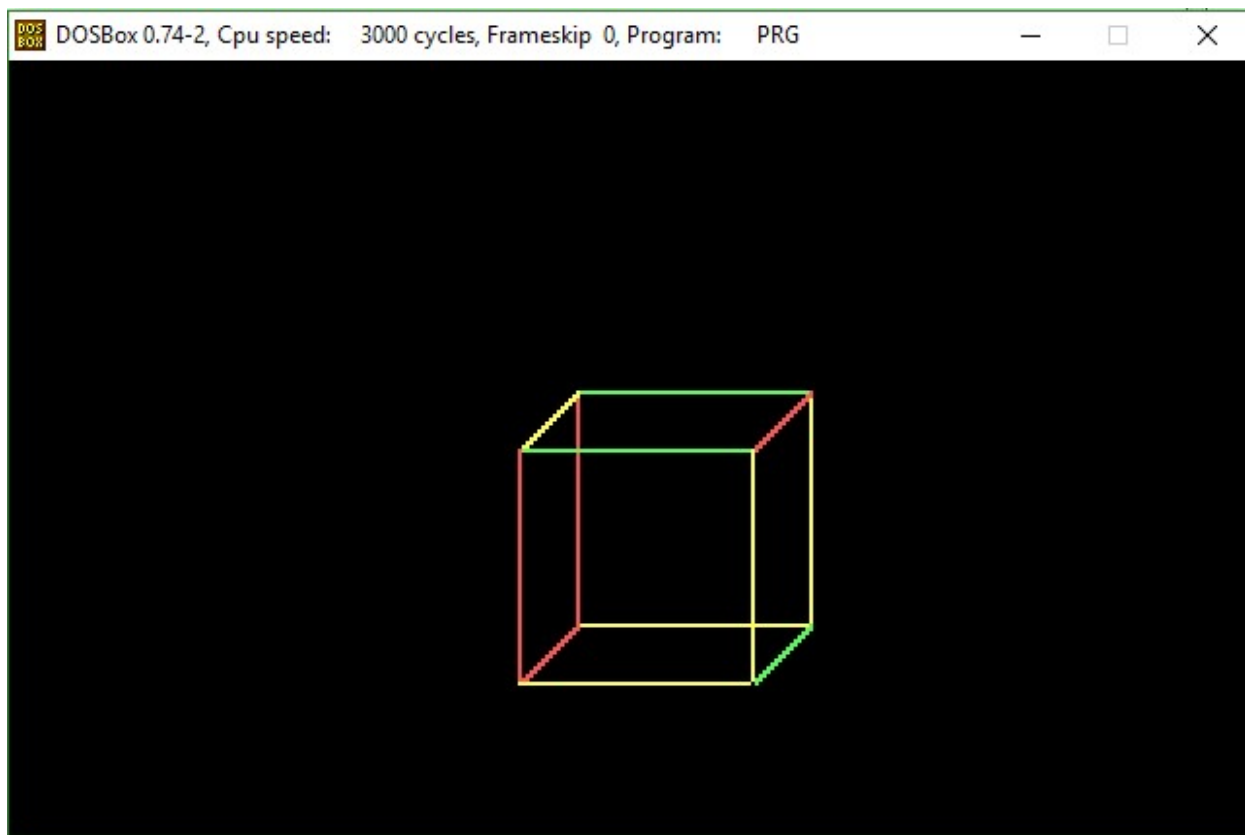


Рисунок 2 – Экранная форма программы

## 5 Вывод

В ходе выполнения лабораторной работы была разработана программа, демонстрирующая работу с видеопамятью. Был подробно изучен 4-й видеорежим и различные виды цветов и палитры. Так же были изучены функции работы с клавиатурой через прерывание `int 16h`. Результатом выполнения лабораторной работы стала программа, выводящая на экран изображение с возможностью его передвижения по экрану при помощи стрелок клавиатуры. Смена палитры происходит при нажатии клавиши пробел.