Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение   
высшего профессионального образования   
«Алтайский государственный технический университет им. И.И. Ползунова»

Факультет информационных технологий

Кафедра прикладной математики

Отчет защищен с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Л. И. Сучкова

(подпись) (и.о., фамилия)

“\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_2014 г.

дата

Отчет по лабораторной работе №4

по дисциплине Периферийные устройства

Студент группы ПИ-21 С. Ю. Искуснов

и.о., фамилия

Преподаватель к.т.н., профессор Л. И. Сучкова\_

должность, ученое звание и.о., фамилия

Барнаул 2014

Задание:

Написать приложение, осуществляющее:

1. Вывод текста с использованием не менее чем двух типов шрифтов.

2. Вывод графических примитивов (прямоугольники, эллипсы, дуги и т.п.) с использованием различных перьев и кистей для заливки;

3. Вывод мини-мультфильма по сюжету. Сюжет мультиплика-ции придумывается студентом и согласовывается с преподавателем с целью одинакового уровня сложности различных вариантов сюжетов. Самостоятельно решить задачу в соответствии с индивидуальным ва-риантом.

4. Изучить средства загрузки DLL, оформить 2-3 функции выво-да данных на графическое устройство как DLL. Функции для библио-теки подобрать самостоятельно.

Код программы:

.386

.model flat, stdcall

option casemap:none

;Библиотеки

include d:\masm32\include\windows.inc

include d:\masm32\include\user32.inc

include d:\masm32\include\kernel32.inc

include d:\masm32\include\gdi32.inc

includelib d:\masm32\lib\user32.lib

includelib d:\masm32\lib\kernel32.lib

includelib d:\masm32\lib\gdi32.lib

;Прототипы

WinMain proto :DWORD,:DWORD,:DWORD,:DWORD

DrawStrings proto :HWND,:HDC

DrawGraphics proto :HWND,:HDC

DrawAnimation proto :HWND,:HDC

DrawMenu proto :HWND,:HDC

.data

ClassName db "Form1", 0 ;Название класса

AppName db "PL4", 0 ;Название приложения

hInstance HINSTANCE ? ;Дескриптор приложения

CommandLine LPSTR ? ;Дескриптор консоли

hAnimation HGDIOBJ 0 ;Дескриптор картинки

x0 DWORD 0 ;x координата начала чтения в источнике

y0 DWORD 125 ;y координата начала чтения в источнике

x DWORD 10 ;x координата начала записи в приемнике

y DWORD 50 ;y координата начала записи в приемнике

w DWORD 120 ;Ширина чтения\записи картинки

h DWORD 120 ;Высота чтения\записи картинки

flag DWORD 0 ;Флаг режима отрисовки

;Строки

menu1 db "1.Fonts",0

menu2 db "2.Graphics",0

menu3 db "3.Animation",0

menureset db "Space to reset",0

text1 db "The first type of font",0

text2 db "The second type of font",0

text3 db "The third type of font",0

font1 db "Times New Roman",0

font2 db "Lucida Console",0

font3 db "Courier New",0

.code

start:

invoke GetModuleHandle, NULL

mov hInstance, eax

invoke GetCommandLine

mov CommandLine, eax

invoke WinMain, hInstance, NULL, CommandLine, SW\_SHOWDEFAULT

invoke ExitProcess, eax

WinMain proc hInst:HINSTANCE, hPrevInst:HINSTANCE, CmdLine:LPSTR, CmdShow:DWORD

LOCAL wc:WNDCLASSEX

LOCAL msg:MSG

LOCAL hWnd:HWND

;Заполнение стpуктуpы wc

mov wc.cbSize,SIZEOF WNDCLASSEX

mov wc.style, CS\_BYTEALIGNWINDOW

mov wc.lpfnWndProc, OFFSET WndProc

mov wc.cbClsExtra,NULL

mov wc.cbWndExtra,NULL

push hInstance

pop wc.hInstance

mov wc.hbrBackground,COLOR\_WINDOW+1

mov wc.lpszMenuName,NULL

mov wc.lpszClassName,OFFSET ClassName

invoke LoadIcon,NULL,IDI\_APPLICATION

mov wc.hIcon,eax

mov wc.hIconSm,eax

invoke LoadCursor,NULL,IDC\_ARROW

mov wc.hCursor,eax

invoke RegisterClassEx, addr wc ; pегистpация нашего класса окна

invoke CreateWindowEx, NULL,

ADDR ClassName,

ADDR AppName,

WS\_OVERLAPPEDWINDOW, ;Стиль окна

CW\_USEDEFAULT,

CW\_USEDEFAULT,

CW\_USEDEFAULT,

CW\_USEDEFAULT,

NULL, ;Дескриптор родительского окна

NULL,

hInst, ;Дескриптор приложения

NULL

mov hWnd,eax

invoke ShowWindow, hWnd, SW\_SHOWNORMAL ; отобpазить окно

invoke UpdateWindow, hWnd ; обновить клиентскую область

invoke LoadBitmap, hInstance, 1 ; загрузка дескриптора приложения

mov hAnimation, eax

invoke InvalidateRect, hWnd, NULL, TRUE

.WHILE TRUE ;Цикл обработки сообщений

invoke GetMessage, ADDR msg,NULL,0,0

.BREAK .IF (!eax)

invoke TranslateMessage, ADDR msg

invoke DispatchMessage, ADDR msg

.ENDW

mov eax,msg.wParam ; сохpанение возвpащаемого значения в eax

ret

WinMain endp

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

LOCAL hDC:HDC

LOCAL PS:PAINTSTRUCT

.if uMsg==WM\_DESTROY ; сообщение о закрытии окна

invoke PostQuitMessage,NULL ; отправить сообщение о выходе из пpогpаммы

.elseif uMsg==WM\_KEYUP

.if wParam==VK\_SPACE

mov flag,0

mov x0,0

mov y0,125

mov x,0

.elseif wParam==VK\_1

mov flag,1

.elseif wParam==VK\_2

mov flag,2

.elseif wParam==VK\_3

mov flag,3

.endif

invoke InvalidateRect, hWnd, NULL, TRUE

.elseif uMsg==WM\_PAINT

invoke BeginPaint, hWnd, addr PS

mov hDC, eax

.if flag==1

invoke DrawStrings,hWnd,hDC

.elseif flag==2

invoke DrawGraphics,hWnd,hDC

.elseif flag==3

invoke DrawAnimation,hWnd,hDC

invoke InvalidateRect, hWnd, NULL, TRUE

.else

invoke DrawMenu, hWnd, hDC

.endif

invoke EndPaint, hWnd, addr PS

.else

invoke DefWindowProc,hWnd,uMsg,wParam,lParam

ret

.endif

xor eax,eax

ret

WndProc endp

DrawMenu proc hWnd:HWND, hDC:HDC

local lx:DWORD

local ly:DWORD

mov lx,250

mov ly,100

invoke TextOut,hDC,lx,ly,addr menu1,7

add ly,20

invoke TextOut,hDC,lx,ly,addr menu2,10

add ly,20

invoke TextOut,hDC,lx,ly,addr menu3,11

add ly,40

sub lx,5

invoke TextOut,hDC,lx,ly,addr menureset,14

xor eax,eax

ret

DrawMenu endp

DrawStrings proc hWnd:HWND, hDC:HDC

local font: LOGFONT

local hOld: DWORD

local hOldColor: DWORD

local hOldBgColor: DWORD

;Первый шрифт

mov font.lfHeight,40 ;высота

mov font.lfWidth,20 ;средняя ширина

mov font.lfEscapement,30 ;наклон текста

mov font.lfOrientation,40 ;наклон букв

mov font.lfWeight,1000 ;толщина

mov font.lfItalic,FALSE ;курсивного

mov font.lfUnderline,FALSE ;подчеркнутого

mov font.lfStrikeOut,FALSE ;перечеркнутого

mov font.lfCharSet,DEFAULT\_CHARSET ;начертание (набор символов логического шрифта)

mov font.lfOutPrecision,OUT\_DEFAULT\_PRECIS ;Поручает отображения логического шрифта выбрать шрифт TrueType, когда система содержит многочисленные шрифты с тем же именем

mov font.lfClipPrecision,CLIP\_DEFAULT\_PRECIS;(Определяет поведение по умолчанию отсечения.)

;точность отсечения определяет, как обрезать символы, которые частично вне области отсечения.

mov font.lfQuality,DEFAULT\_QUALITY ;внешний вид не имеет значения

mov font.lfPitchAndFamily,FF\_ROMAN ;шаг и семейство шрифта (Шрифты с переменной шириной штриха (пропорционального) и с засечками. MS Serif )

invoke lstrcpy, addr font.lfFaceName, addr font1 ; копируем имя гарнитуры шрифта

invoke CreateFontIndirect,addr font ;подборка наиболее подходящего шрифта и возвращение его дескриптора

invoke SelectObject,hDC,eax

mov hOld,eax

invoke SetTextColor,hDC,008895ECh

mov hOldColor,eax

invoke SetBkColor,hDC,00969888h

mov hOldBgColor,eax

invoke TextOut,hDC,200,50,addr text1,22

;Второй шрифт

mov font.lfHeight,15 ;высота

mov font.lfWidth,5 ;средняя ширина

mov font.lfEscapement,-30 ;наклон текста

mov font.lfOrientation,-40 ;наклон букв

mov font.lfWeight,100 ;толщина

mov font.lfItalic,TRUE ;курсивного

mov font.lfUnderline,FALSE ;подчеркнутого

mov font.lfStrikeOut,TRUE ;перечеркнутого

mov font.lfCharSet,DEFAULT\_CHARSET ;начертание (набор символов логического шрифта)

mov font.lfOutPrecision,OUT\_TT\_ONLY\_PRECIS ;Поручает отображения логического шрифта выбрать шрифт TrueType, когда система содержит многочисленные шрифты с тем же именем

mov font.lfClipPrecision,CLIP\_LH\_ANGLES ;(Определяет поведение по умолчанию отсечения.)

;точность отсечения определяет, как обрезать символы, которые частично вне области отсечения.

mov font.lfQuality,DRAFT\_QUALITY ; внешний вид не имеет значения

mov font.lfPitchAndFamily,FF\_SWISS ; шаг и семейство шрифта (Шрифты с переменной шириной штриха (пропорционального) и с засечками. MS Serif )

invoke lstrcpy, addr font.lfFaceName, addr font2 ; копируем имя гарнитуры шрифта

invoke CreateFontIndirect,addr font ;подборка наиболее подходящего шрифта и возвращение его дескриптора

invoke SelectObject,hDC,eax

mov hOld,eax

invoke SetTextColor,hDC,00284F5Bh

mov hOldColor,eax

invoke SetBkColor,hDC,00E8EB52h

mov hOldBgColor,eax

invoke TextOut,hDC,200,100,addr text2,23

;Третий шрифт

mov font.lfHeight,25 ;высота

mov font.lfWidth,15 ;средняя ширина

mov font.lfEscapement,0 ;наклон текста

mov font.lfOrientation,0 ;наклон букв

mov font.lfWeight,800 ;толщина

mov font.lfItalic,TRUE ;курсивного

mov font.lfUnderline,TRUE ;подчеркнутого

mov font.lfStrikeOut,FALSE ;перечеркнутого

mov font.lfCharSet,DEFAULT\_CHARSET ;начертание (набор символов логического шрифта)

mov font.lfOutPrecision,OUT\_STRING\_PRECIS ;Поручает отображения логического шрифта выбрать шрифт TrueType, когда система содержит многочисленные шрифты с тем же именем

mov font.lfClipPrecision,CLIP\_TT\_ALWAYS ;(Определяет поведение по умолчанию отсечения.)

;точность отсечения определяет, как обрезать символы, которые частично вне области отсечения.

mov font.lfQuality,PROOF\_QUALITY ; внешний вид не имеет значения

mov font.lfPitchAndFamily,FF\_MODERN ; шаг и семейство шрифта (Шрифты с переменной шириной штриха (пропорционального) и с засечками. MS Serif )

invoke lstrcpy, addr font.lfFaceName, addr font3 ; копируем имя гарнитуры шрифта

invoke CreateFontIndirect,addr font ;подборка наиболее подходящего шрифта и возвращение его дескриптора

invoke SelectObject,hDC,eax

mov hOld,eax

invoke SetTextColor,hDC,006946EFh

mov hOldColor,eax

invoke SetBkColor,hDC,00E443EBh

mov hOldBgColor,eax

invoke TextOut,hDC,200,150,addr text3,22

invoke SelectObject,hDC,hOld

invoke DeleteObject,eax

invoke SetTextColor,hDC,hOldColor

invoke SetBkColor,hDC,hOldBgColor

xor eax,eax

ret

DrawStrings endp

DrawGraphics proc hWnd:HWND, hDC:HDC

local hPen:DWORD

local hBrush:DWORD

local hOldPen:DWORD

local hOldBrush:DWORD

local pen:LOGPEN

local brush:LOGBRUSH

local p1:POINT

local p2:POINT

local p3:POINT

;прямоугольник

mov brush.lbStyle,BS\_HATCHED

mov brush.lbColor,008895ECh

mov brush.lbHatch,HS\_BDIAGONAL

invoke CreateBrushIndirect,addr brush

mov hBrush,eax

invoke SelectObject,hDC,eax

mov hOldBrush,eax

invoke Rectangle,hDC,10,10,100,100

;эллипс

mov brush.lbStyle,BS\_SOLID

mov brush.lbColor,00E443EBh

invoke CreateBrushIndirect,addr brush

mov hBrush,eax

invoke SelectObject,hDC,hBrush

invoke DeleteObject,eax

invoke Ellipse,hDC,10,200,150,150

;прямоугольник с закругленными краями

mov brush.lbStyle,BS\_SOLID ;сплошная кисть

mov brush.lbColor,006946EFh

invoke CreateBrushIndirect,addr brush

mov hBrush,eax

invoke SelectObject,hDC,hBrush

invoke DeleteObject,eax

invoke RoundRect,hDC,400,10,150,200,15,10

;треугольник

invoke CreatePen,PS\_DASH,5,008895ECh

mov hPen,eax

invoke SelectObject,hDC,hPen

mov hOldPen,eax

mov p1.x,200

mov p1.y,200

mov p2.x,400

mov p2.y,200

mov p3.x,100

mov p3.y,300

mov brush.lbStyle,BS\_HATCHED

mov brush.lbColor,00add094h

mov brush.lbHatch,HS\_VERTICAL

invoke CreateBrushIndirect,addr brush

mov hBrush,eax

invoke SelectObject,hDC,hBrush

invoke DeleteObject,eax

invoke Polygon,hDC,ADDR p3,3

invoke SelectObject,hDC,hOldPen

invoke DeleteObject,eax

invoke SelectObject,hDC,hOldBrush

invoke DeleteObject,eax

xor eax,eax

ret

DrawGraphics endp

DrawAnimation proc hWnd:HWND, hDC:HDC

local hOld:HGDIOBJ

local memDC:HDC

local rect:RECT

invoke GetWindowRect,hWnd,addr rect

mov eax,rect.right

sub eax,rect.left

push eax

invoke CreateCompatibleDC,hDC

mov memDC, eax

invoke SelectObject, memDC, hAnimation

mov hOld, eax

invoke BitBlt,hDC,x,y,w,h,memDC,x0,y0,SRCCOPY

invoke SelectObject, memDC, hOld

invoke DeleteDC, memDC

.if y0<200

add x0, 120

.endif

add x, 25

.if x0>900

mov x0, 0

add y0, 129

.endif

.if y0>200

mov y0, 125

.endif

pop eax

.if x>eax

mov x, 10

.endif

invoke Sleep, 200

xor eax,eax

ret

DrawAnimation endp

end start