МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»

(БГТУ им. В.Г. Шухова)

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Дисциплина: Архитектура ВС

Индивидуальное домашнее задание

Тема: «Создание Windows-приложения на ассемблере»

Выполнил:

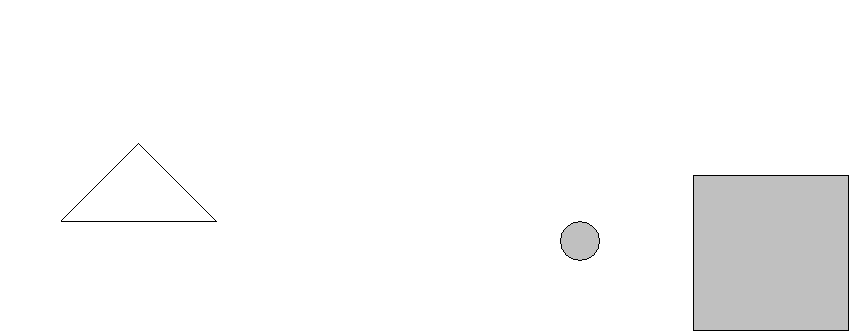
ст. группы ПВ–31

Заановская А.И.

Проверил: Осипов О.В.

Белгород 2018

Цель работы: создание оконного графического приложения на ассемблере.



.386

.model flat, stdCALL

option casemap : none

include C:\Users\Acer\Desktop\LABS\ABC\masm32\include\windows.inc

include C:\Users\Acer\Desktop\LABS\ABC\masm32\include\kernel32.inc

include C:\Users\Acer\Desktop\LABS\ABC\masm32\include\user32.inc

include C:\Users\Acer\Desktop\LABS\ABC\masm32\include\gdi32.inc

includelib C:\Users\Acer\Desktop\LABS\ABC\masm32\lib\user32.lib

includelib C:\Users\Acer\Desktop\LABS\ABC\masm32\lib\kernel32.lib

includelib C:\Users\Acer\Desktop\LABS\ABC\masm32\lib\gdi32.lib

.data?

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

hInstance dd ?

commandline dd ?

.data

ClassName db "simpleWinClass",0

AppName db "RGZ",0

areaX SDWORD ?

areaY SDWORD ?

areaR SDWORD ?

areaX1 SDWORD ?

areaY1 SDWORD ?

areaR1 SDWORD ?

maxX SDWORD ?

maxY SDWORD ?

areaI SDWORD 170 ;площадь 1

areaI1 SDWORD 200 ;площадь 2

otstup SDWORD 60

sx SDWORD ?

sy SDWORD ?

SX SDWORD ?

SY SDWORD ?

ballX SDWORD ?

ballY SDWORD ?

ballR SDWORD 20

balldX SDWORD 5

balldY SDWORD 5

deltX SDWORD 5

deltY SDWORD 5

ballS SDWORD 9

ballA SDWORD 60

ballA\_ SDWORD ?

temp SDWORD ?

c180 SDWORD 180

c100 SDWORD 100

c90 SDWORD 90

c5 SDWORD 5

cP1 SDWORD 5

c2 SDWORD 2

c1 SDWORD 1

c0 SDWORD 0

t1 SDWORD ?

t2 SDWORD ?

t3 SDWORD ?

hdc SDWORD ?

memBit SDWORD ?

hBitmap SDWORD ?

ps PAINTSTRUCT <?>

.code

start:

;Точка входа в приложение.

INVOKE GetModuleHandle, NULL ;получим handle экземпляра программы

MOV hInstance, EAX

INVOKE GetCommandLine ;получим адрес командной строки

MOV commandline, EAX

INVOKE WinMain, hInstance, NULL, commandline, SW\_SHOWDEFAULT ;вызов основной программы (в стандарте Windows)

INVOKE ExitProcess, EAX ;выход из программы с кодом в EAX

;;Инициализация окна.

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

LOCAL wc:WNDCLASSEX

LOCAL msg:MSG

LOCAL hwnd: HWND

MOV wc.cbSize, SIZEOF WNDCLASSEX

MOV wc.style, CS\_HREDRAW or CS\_VREDRAW

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, hInst, 500

MOV wc.hIcon, EAX

MOV wc.hIconSm, EAX

INVOKE LoadCursor, NULL, IDC\_ARROW

MOV wc.hCursor, EAX

INVOKE RegisterClassEx, ADDr wc

INVOKE CreateWindowEx, NULL, ADDR ClassName, ADDR AppName,\

13303808, CW\_USEDEFAULT,\

CW\_USEDEFAULT, CW\_USEDEFAULT,CW\_USEDEFAULT, NULL, NULL,\

hInst, NULL

MOV hwnd, EAX

INVOKE LoadMenu, hInst, 600

INVOKE SetMenu, hwnd, EAX

INVOKE ShowWindow, hwnd, SW\_SHOWNORMAL

INVOKE UpdateWindow, hwnd

.WHILE TRUE

INVOKE GetMessage, ADDR msg, NULL, 0, 0

.BREAK .IF (!EAX)

INVOKE TranslateMessage, ADDR msg

INVOKE DispatchMessage, ADDR msg

.ENDW

MOV EAX, msg.wParam

RET

WinMain endp

;Работа программы

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

;################# Закрытие окна

.if uMsg == WM\_DESTROY

INVOKE DeleteObject, memBit

INVOKE PostQuitMessage, NULL

; Создание окна

.elseif uMsg==WM\_CREATE

;устанавливаем таймер

FILD c1

FILD c100

FDIV

FST temp

INVOKE SetTimer, hWnd, 1, c2, 0

; получение максимальных размеров окон

INVOKE GetSystemMetrics, SM\_CXSCREEN

MOV maxX, EAX

INVOKE GetSystemMetrics, SM\_CYSCREEN

MOV maxY, EAX

;получение контекста

INVOKE GetDC, hWnd

MOV hdc, EAX

; Рисовать будем в памяти

INVOKE CreateCompatibleDC, hdc

MOV memBit, EAX

INVOKE CreateCompatibleBitmap, hdc, maxX, maxY

MOV hBitmap, EAX

INVOKE SelectObject, memBit, hBitmap

; Отрисовка

.elseif uMsg==WM\_PAINT

;Отрисовка фигур

INVOKE PatBlt, memBit,0,0, maxX, maxY, WHITENESS

INVOKE GetStockObject, c1

INVOKE SelectObject, memBit, EAX

;Квадрат

FILD areaY ;загрузить целое число со знаком в стек

FILD areaR

FADD

FISTP temp ;преобразовывает число из вершины стека в целое со знаком и записывает его в приёмник

PUSH temp

FILD arEAX

FILD areaR

FADD

FISTP temp

PUSH temp

FILD areaY

FILD areaR

FSUB

FISTP temp

PUSH temp

FILD arEAX

FILD areaR

FSUB

FISTP temp

PUSH temp

PUSH memBit

CALL Rectangle

;треугольник

PUSH 0

FILD areaR

FILD otstup

FADD

FISTP temp

PUSH temp

PUSH temp

PUSH memBit

CALL MOVeToEx

FILD areaR

FADD ST, ST

FILD otstup

FADD

FISTP temp

PUSH temp

PUSH otstup

PUSH memBit

CALL LineTo

PUSH 0

FILD areaR

FILD otstup

FADD

FISTP temp

PUSH temp

PUSH temp

PUSH memBit

CALL MOVeToEx

FILD areaR

FADD ST, ST

FILD otstup

FADD

FISTP temp

PUSH temp

PUSH temp

PUSH memBit

CALL LineTo

PUSH 0

PUSH temp

PUSH temp

PUSH memBit

CALL MOVeToEx

PUSH temp

PUSH otstup

PUSH memBit

CALL LineTo

;Шарик 1

FILD ballY

FILD ballR

FADD

FISTP temp

PUSH temp

FILD ballX

FILD ballR

FADD

FISTP temp

PUSH temp

FILD ballY

FILD ballR

FSUB

FISTP temp

PUSH temp

FILD ballX

FILD ballR

FSUB

FISTP temp

PUSH temp

PUSH memBit

CALL Ellipse

;Отрисовка из памяти в окно

INVOKE BeginPaint, hWnd, offset ps

MOV hdc, EAX

INVOKE BitBlt, hdc,0,0, sx,sy, memBit,0,0, SRCCOPY

INVOKE EndPaint, hWnd, offset ps

; Получение размеров окна

.elseif uMsg==WM\_SIZE

XOR EAX, EAX

MOV ax, word ptr lParam

MOV sx, EAX

MOV EAX, lParam

SHR EAX, 16

MOV sy, EAX

;расчет размеров квадрата

MOV EAX, sy

MOV EBX, sx

.if sx>EAX

MOV temp,EAX

.else

MOV temp, EBX

.endif

FILD c2

FLD areaI

FMUL

FLD temp

FXCH

FSUB

FILD c2

FDIV

FSTP areaR

MOV EAX, c2

.if areaR < EAX

MOV areaR, EAX

.endif

FLD areaR

FLD areaI

FADD

FSTP areaY

FLD sx

FLD areaI

FSUB

FLD areaR

FSUB

FSTP arEAX

MOV ballX, 300

MOV ballY, 56

;Срабатывание таймера

.elseif uMsg==WM\_TIMER

MOV EBX, balldX

ADD ballX, EBX

MOV ECX, sx

SUB ECX, ballR

;правый край

.if ballX>ECX

SUB ballX, EBX

MOV EBX, deltX

NEG EBX

MOV balldX, EBX

.endif

MOV ECX, ballR

;левый край

.if ballX<ECX

ADD ballX, EBX

MOV EBX, deltX

MOV balldX, EBX

.endif

;внутри квадрата

MOV ECX, areaY

SUB ECX, areaR;верхний край квадрата

SUB ECX, ballR;c учетом радиуса шарика

.if (ballY > ECX)

MOV EDX, areaY

ADD EDX, areaR;нижний край квадрата

ADD EDX, ballR

.if (ballY < EDX)

MOV ECX, arEAX

SUB ECX, areaR

SUB ECX, ballR

.if ballX>ECX

MOV ECX, arEAX

ADD ECX, areaR

ADD ECX, ballR

.if ballX<ECX

NEG balldX

MOV EBX, balldX

ADD ballX, EBX

.endif

.endif

.endif

.endif

MOV EBX, balldY

ADD ballY, EBX

MOV ECX, sy

SUB ECX, ballR

;верхний край

.if ballY>ECX

SUB ballY, EBX

MOV EBX, deltY

NEG EBX

MOV balldY, EBX

.endif

MOV ECX, ballR

;нижний край

.if ballY<ECX

ADD ballY, EBX

MOV EBX, deltY

MOV balldY, EBX

.endif

;внутри квадрата

MOV ECX, arEAX

SUB ECX, areaR;верхний край квадрата

SUB ECX, ballR;c учетом радиуса шарика

.if (ballX > ECX)

MOV EDX, arEAX

ADD EDX, areaR;нижний край квадрата

ADD EDX, ballR

.if (ballX < EDX)

MOV ECX, areaY

SUB ECX, areaR

SUB ECX, ballR

.if ballY>ECX

MOV ECX, areaY

ADD ECX, areaR

ADD ECX, ballR

.if ballY<ECX

NEG balldY

MOV EBX, balldY

ADD ballY, EBX

.endif

.endif

.endif

.endif

;у треугольника

FILD areaR

FADD ST, ST

FILD ballR

FADD

FILD otstup

FADD

FISTP temp

MOV ECX, temp

.if ballY < ECX

SUB ECX, ballR

SUB ECX, ballR

.if ballY > ECX

MOV ECX, otstup

SUB ECX, ballR

.if ballX > ECX

FILD areaR

FADD ST, ST

FILD ballR

FADD

FILD otstup

FADD

FISTP temp

MOV ECX, temp

.if ballX < ECX

MOV EBX, deltY

MOV balldY, EBX

.endif

.endif

.else

FILD areaR

FILD otstup

FADD

FILD ballR

FSUB

FISTP temp

MOV ECX, temp

.if ballX > ECX

MOV ECX, ballX

SUB ECX, ballR

.if ballY > ECX

MOV EBX, deltX

MOV balldX, EBX

MOV EBX, deltY

NEG EBX

MOV balldY, EBX

.endif

.else

FILD areaR

FILD otstup

FADD

FIST temp

FILD ballX

FSUB st, st(1)

FILD ballR

FADD

FILD areaR

FADD st, st

FILD otstup

FADD

FSUB st, st(2)

FMULp st(1), st

FILD otstup

FSUB st, st(2)

FDIVp st(1), st

FADD

FISTP temp

MOV ECX, temp

.if ballY > ECX

MOV EBX, deltX

NEG EBX

MOV balldX, EBX

MOV EBX, deltY

NEG EBX

MOV balldY, EBX

.endif

.endif

.endif

.endif

INVOKE InvalidateRect, hWnd, 0,0

.else

INVOKE DefWindowProc, hWnd, uMsg, wParam,lParam

RET

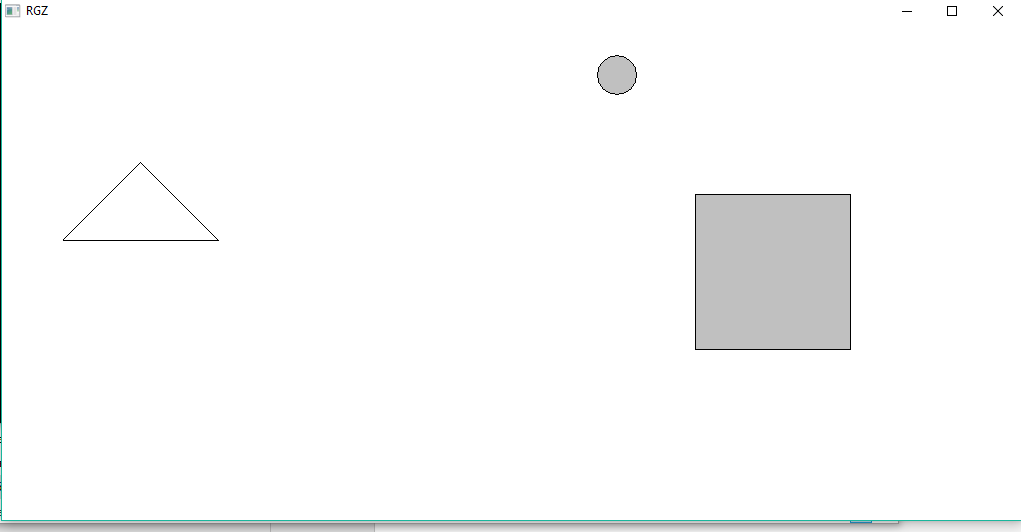
.endif

XOR EAX, EAX

RET

WndProc endp

END START



Вывод: были получены и закреплены навыки создания оконного приложения Windows на ассемблере.