**Министерство науки и высшего образования Российской Федерации**

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина»

Кафедра «ЭВМ»

Машинно-зависимые языки программирования

**Отчет о лабораторной работе № 5**

по теме

«Вычисление математической функции в

оконном приложении»

**Выполнили:**

Студенты гр. 045

Анохин В.А.

Вашкулатов Н.А.

**Проверил:**

Доц. Муратов Е.Р

Асс. Тарасов А.С.

**Цель работы:** научиться создавать простейшее приложение с графическим интерфейсом средствами WinAPI на языке ассемблера, научиться вычислять математические функции на FPU на языке ассемблера.

**Ход работы:**

Реализовать оконное приложение выполняющее вычисление функции на FPU согласно заданию: запрограммировать функцию cos(2\*x).

**Код программы:**

.686P

.MODEL FLAT, stdcall

option casemap : none

RGBW equ 00CCCCCCh; цвет фона окна

include win.inc

include console.inc

include C:/masm32/include/masm32.inc

includelib C:/masm32/lib/masm32.lib

.data

HWND DD 0; дескриптор главного окна

HINST DD 0; дескриптор приложения

TITL DB "Программа", 0

CLASSNAME DB 'CLASS32', 0

Message MSG < ? >

WC WNDCLASS < ? >

CPBUT db 'Рассчитать', 0

CLSBTN db 'BUTTON', 0

CLSEDT db 'EDIT', 0

CAP db 'Сообщение', 0

TEXTA db 20 dup(0); текст в полях редактирования

num dq 0.0

num2 dd 0

hBut DD ? ; дескриптор кнопки

hedt1 DD ? ; дескриптор поля 1

hdc DD ? ; дескриптор контекста окна

ps PAINTSTRUCT < ? >

mess1 db 'Посчитать функцию : ', 0; надпись в окне

mess1\_len equ $ - mess1 - 1

mess2 db '=', 10 dup(' '), 0; результат суммы строковый

sum\_len equ $ - mess2 - 1

.code

START proc

; получить дескриптор приложения

INVOKE GetModuleHandle, 0

MOV HINST, EAX

; заполнить структуру окна стиль

MOV WC.style, CS\_HREDRAW + CS\_VREDRAW + CS\_GLOBALCLASS

; процедура обработки сообщений

MOV WC.lpfnWndProc, OFFSET WNDPROC;процедура обработки кода окна должна быть

;представлена в коде(пример процедуры будет приведен ниже)

MOV EAX, HINST

MOV WC.hInstance, EAX

INVOKE LoadIcon, 0, IDI\_APPLICATION;грузит иконки

MOV WC.hIcon, EAX

INVOKE LoadCursor, 0, IDC\_ARROW;грузит курсор

MOV WC.hCursor, EAX

INVOKE CreateSolidBrush, RGBW;создает кисть для заполнения фона окна

MOV WC.hbrBackground, EAX

MOV DWORD PTR WC.lpszMenuName, 0

MOV DWORD PTR WC.lpszClassName, OFFSET CLASSNAME

INVOKE RegisterClass, OFFSET WC

; создать окно зарегистрированного класса

INVOKE CreateWindowEx, 0,

OFFSET CLASSNAME,

OFFSET TITL,

WS\_CAPTION + WS\_SYSMENU + WS\_THICKFRAME + WS\_GROUP + WS\_TABSTOP,

100, ; X — координата левого верхнего угла

100, ; Y — координата левого верхнего угла

400, ; DX - ширина окна

450, ; DY — высота окна

0, 0, HINST, 0

CMP EAX, 0; проверка на ошибку

JZ END\_LOOP

MOV HWND, EAX; дескриптор окна

INVOKE ShowWindow, HWND, SW\_SHOWNORMAL; показать созданное окно

INVOKE UpdateWindow, HWND;перерисовать видимую часть окна

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

; цикл обработки сообщений

MSG\_LOOP :

INVOKE GetMessage, OFFSET Message, 0, 0, 0

CMP EAX, 0

JE END\_LOOP

INVOKE TranslateMessage, OFFSET Message

INVOKE DispatchMessageA, OFFSET Message

JMP MSG\_LOOP

END\_LOOP :

INVOKE ExitProcess, Message.wParam; выход из программы

START endp

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

; процедура окна

WNDPROC PROC hW : DWORD, Mes : DWORD, wParam : DWORD, lParam : DWORD

CMP Mes, WM\_DESTROY;будет выполняться при закрытии окна

JE WMDESTROY

CMP Mes, WM\_CREATE; при создании окна

JE WMCREATE

CMP Mes, WM\_COMMAND; общие команды(сообщения)

JE WMCOMMAND

CMP Mes, WM\_PAINT; события перерисовки формы окна

JE WMPAINT

JMP DEFWNDPROC

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

WMCREATE:; создание окна

INVOKE CreateWindowExA, 0, ; поле редактирования 1

offset CLSEDT, ; имя класса окна

offset TEXTA, ; надпись в поле

WS\_CHILD + WS\_VISIBLE, ; стиль окна

10, 50, ;x, y

60, 20, ; длина, ширина

hW, ;дескриптор окна

0, ;дескриптор меню

HINST, ;дескриптор приложения

0;lpParam

mov hedt1, eax; сохранение дескриптора

mov eax, 0

INVOKE ShowWindow, hedt1, SW\_SHOWNORMAL

INVOKE CreateWindowExA, 0, ; кнопка

offset CLSBTN, ; имя класса окна

offset CPBUT, ; надпись на кнопке

WS\_CHILD + WS\_VISIBLE, ; стиль окна кнопки

10, 90, ;x, y

100, 20, ; длина, ширина

hW, ;дескриптор окна

0, ;дескриптор меню

HINST, ;дескриптор приложения

0;lpParam

mov hBut, eax; сохранение дескриптора

mov eax, 0

INVOKE ShowWindow, hBut, SW\_SHOWNORMAL

MOV EAX, 0

JMP FINISH

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

WMCOMMAND: ; обработка нажатия кнопки

mov eax, hBut

cmp lParam, eax

jne COM\_END; команда не соответствует нажатию кнопки

INVOKE SendMessage, hedt1, WM\_GETTEXT, 20, offset TEXTA

INVOKE StrToFloat, ADDR TEXTA, ADDR num

mov num2, 2

FINIT

fld num

fild num2

fmul ST, ST(1)

fcos

fstp num

mov eax, sum\_len

INVOKE TextOutA, hdc, 80, 50, offset mess2, eax; стирание строки результата в окне

INVOKE FloatToStr, num, ADDR mess2

INVOKE LENSTR, offset mess2; определение длины результата

push eax

INVOKE TextOutA, hdc, 80, 50, offset mess2, eax; вывод результата

pop ecx; очистка строки

inc ecx

mov al, ' '

mov edi, offset mess2 + 1

CLR: mov[edi], al

inc edi

loop CLR

COM\_END :

MOV EAX, 0

JMP FINISH

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

WMPAINT:; перерисовка окна

INVOKE BeginPaint, hW, offset ps

mov hdc, eax

INVOKE SetBkColor, hdc, RGBW

mov eax, mess1\_len

INVOKE TextOutA, hdc, 10, 20, offset mess1, eax

INVOKE EndPaint, hdc, offset ps

MOV EAX, 0

JMP FINISH

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

DEFWNDPROC:; обработка сообщения по умолчанию

INVOKE DefWindowProc,

hW, Mes, wParam, lParam

JMP FINISH

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

WMDESTROY:; выход из цикла обработки сообщений

INVOKE PostQuitMessage, 0

MOV EAX, 0

FINISH: ret

WNDPROC ENDP

END START

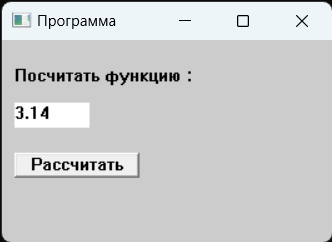


Рисунок 1 – Окно программы

**Вывод:** в ходе выполнения работы мы научились создавать простейшее приложение с графическим интерфейсом средствами WinAPI на языке ассемблера и вычислять математические функции на FPU на языке ассемблера.