**Міністерство освіти і науки України**

**Національний технічний університет України**

**«Київський політехнічний інститут імені Ігоря Сікорського»**

**Факультет інформатики та обчислювальної техніки**

**Кафедра обчислювальної техніки**

**Лабораторна робота № 7**

з дисципліни

«Системне програмування»

на тему “Модульне програмування. Використання процедур.”

Виконав: Перевірив:

студент групи ІП-05 доцент Павлов В.Г.

Кононенко Вадим Олександрович

номер залікової книжки: 0512

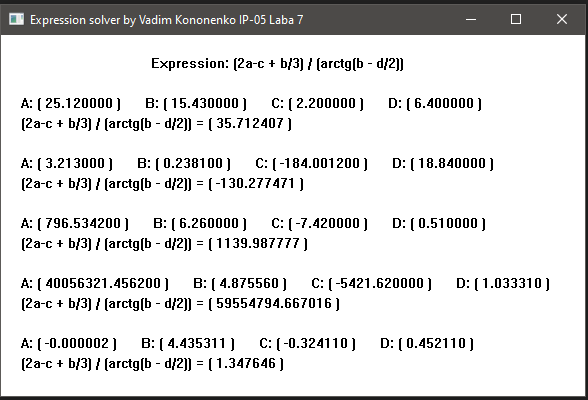
Варіант 13

Київ-2022

**Мета роботи** :

Вивчення прийомів модульного програмування, методів звернення до процедур і передачі в них параметрів.

**Результат роботи програми:**



**Контрольні розрахунки**

**First.**

**A:** (25,12) **B:** (15,43) **C:** (2,2) **D:** (6,4)

(2\*25,12 – 2,2 + 15,43/3) / arctg(15,43 – 6,4/2) = (50,24 – 2,2 + 1543/100\*1/3) / arctg(15,43 – 3,2) = (1201/25 + 1543/300) \* 1/arctg(12,23) = (3191/60) \* 1/arctg(12,23) = **35,71241**

**Second.**

**A:** (3,213) **B:** (0,2381) **C:** (-184,001200) **D:** (18,84)

(2\*3,213 + 184,0012 + 0,2381/3) / arctg(0,2381-18,84/2) = (190,4272 + 2381/30000) \* 1/arctg(0,2381 – 9,42) = (5715197/30000) \* (1 / -arctg(91819/10000)) = **-130,27747**

**Third.**

**A:** (796,5342) **B:** (6,26) **C:** (-7,42) **D:** (0,51)

(2\*796,5342 + 7,42 + 6,26/3) / arctg(6,26 – 0,51/2) = (1600,4884 + 313/150) \* 1/arctg(6,26 – 0,255) = (12019313/7500) \* 1 / arctg(1201/200) = **1139,98778**

**Fourth.**

**A:** (40056321,4562) **B:** (4,875560) **C:** (-5421,62) **D:** (1,033310)

(2\*40056321,4562 + 5421,62 + 4,87556/3) / arctg(4,87556 – 1,03331/2) = (200295161331/2500 + 121889/75000) \* 1/arctg(4,358905) = (6008854961819/75000) \* 1/arctg(871781/200000) = **5,95548\*10^7**

**Fifth.**

**A:** (-0,000002) **B:** (4,435311) **C:** (-0,324110) **D:** (0,452110)

(2\*(-0,0000023112) + 0,32411 + 4,4353111/3) / arctg(4,4353111 – 0,45211/2) = (202565861/625000000 + 4435311/30000000) \* 1/arctg(4,2092561) = (6759534041/3750000000) \* 1/arctg(42092561/10000000) = **1,34765**

**Код програми** 7-13-IP-05-Kononenko1.asm

include \masm32\include\masm32rt.inc

include \masm32\include\Fpu.inc

includelib \masm32\lib\Fpu.lib

public array\_b, array\_d, number2, count, inter\_solved3

extern res:proto

main\_solver macro index

mov eax, offset inter\_solved1

mov ebx, offset array\_a[8\*index]

mov ecx, offset number2

mov edx, offset array\_c[8\*index]

call left\_solver

push offset inter\_solved2

push number3

push dword ptr array\_b[8\*index + 4]

push dword ptr array\_b[8\*index]

call right\_solver

mov count, index

call res

fld inter\_solved1

fld inter\_solved2

fadd

fld inter\_solved3

fdivp st(1), st

fst solved

invoke crt\_sprintf, addr res\_solved\_values[100\*index],

addr solved\_value\_template,

solved

invoke crt\_sprintf, addr res\_given\_values[200\*index],

addr given\_values\_template,

array\_a[8\*index],

array\_b[8\*index],

array\_c[8\*index],

array\_d[8\*index]

endm

.data

ClassName db "simple", 0

header db "Expression solver by Vadim Kononenko IP-05 Laba 7", 0

given\_values\_template db "A: ( %f )", 10,

" B: ( %f )", 10,

" C: ( %f )", 10,

" D: ( %f )", 0

solved\_value\_template db "(2a-c + b/3) / (arctg(b - d/2)) = ( %f )", 0

expression db "Expression: (2a-c + b/3) / (arctg(b - d/2))", 0

number2 dd 2

number3 dd 3

array\_a dq 25.12, 3.213, 796.5342, 40056321.4562, -0.0000023112

array\_b dq 15.43, 0.2381, 6.26, 4.87556, 4.4353111

array\_c dq 2.2, -184.0012, -7.42, -5421.62, -0.32411

array\_d dq 6.4, 18.84, 0.51, 1.03331, 0.45211

.data?

hInstance HINSTANCE ?

CommandLine LPSTR ?

solved dq ?

count db ?

res\_given\_values db 1200 dup (?)

res\_solved\_values db 1200 dup (?)

inter\_solved1 dq ?

inter\_solved2 dq ?

inter\_solved3 dq ?

WinMain proto :dword,:dword,:dword,:dword

.code

left\_solver proc

finit

fld qword ptr [ebx]

fimul dword ptr [ecx]

fld qword ptr [edx]

fsubp st(1), st

fst qword ptr [eax]

ret

left\_solver endp

right\_solver proc

push ebp

mov ebp, esp

finit

fld qword ptr [ebp + 8]

fild dword ptr [ebp + 16]

fdivp st(1), st

mov eax, [ebp + 20]

fst qword ptr [eax]

mov esp, ebp

pop ebp

ret 16

right\_solver endp

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 wcx:WNDCLASSEX

local msg:MSG

local hwnd:HWND

mov wcx.cbSize,SIZEOF WNDCLASSEX

mov wcx.style, CS\_HREDRAW or CS\_VREDRAW

mov wcx.lpfnWndProc, OFFSET WndProc

mov wcx.cbClsExtra,NULL

mov wcx.cbWndExtra,NULL

push hInstance

pop wcx.hInstance

mov wcx.hbrBackground,COLOR\_WINDOW+1

mov wcx.lpszMenuName,NULL

mov wcx.lpszClassName,OFFSET ClassName

invoke LoadIcon,NULL,IDI\_APPLICATION

mov wcx.hIcon,eax

mov wcx.hIconSm,eax

invoke LoadCursor,NULL,IDC\_ARROW

mov wcx.hCursor,eax

invoke RegisterClassEx, addr wcx

invoke CreateWindowEx, NULL,ADDR ClassName, ADDR header, WS\_OVERLAPPEDWINDOW, CW\_USEDEFAULT, CW\_USEDEFAULT,

600,

400,

NULL,

NULL,

hInst,

NULL

mov hwnd, eax

invoke ShowWindow, hwnd,CmdShow

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

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

local hdc:HDC

local ps:PAINTSTRUCT

.if uMsg==WM\_CREATE

index = 5;

while index;

index = index - 1

main\_solver index

endm

.elseif uMsg==WM\_DESTROY

invoke PostQuitMessage, NULL

.elseif uMsg==WM\_PAINT

invoke BeginPaint, hWnd, addr ps

mov hdc, eax

invoke TextOut, hdc, 150, 20, addr expression, 43

invoke TextOut, hdc, 20, 60, addr res\_given\_values, 200

invoke TextOut, hdc, 20, 80, addr res\_solved\_values, 60

invoke TextOut, hdc, 20, 120, addr res\_given\_values[200], 200

invoke TextOut, hdc, 20, 140, addr res\_solved\_values[100], 60

invoke TextOut, hdc, 20, 180, addr res\_given\_values[400], 200

invoke TextOut, hdc, 20, 200, addr res\_solved\_values[200], 60

invoke TextOut, hdc, 20, 240, addr res\_given\_values[600], 200

invoke TextOut, hdc, 20, 260, addr res\_solved\_values[300], 60

invoke TextOut, hdc, 20, 300, addr res\_given\_values[800], 200

invoke TextOut, hdc, 20, 320, addr res\_solved\_values[400], 60

invoke EndPaint, hWnd, addr ps

.else

invoke DefWindowProc, hWnd, uMsg, wParam, lParam

ret

.endif

xor eax, eax

ret

WndProc endp

end start

**Код програми** 7-13-IP-05-Kononenko2.asm

.386

.model flat,stdcall

option casemap:none

public res

extern array\_b:qword, array\_d:qword, number2:dword, count:byte, inter\_solved3:qword

.code

res proc

finit

mov al, count

cbw

cwde

fld qword ptr array\_b[eax \* 8]; b в st(0)

fld qword ptr array\_d[eax \* 8]; d в st(0), b в st(1)

fidiv number2; d / 2 в st(0), b в st(1)

fsubp st(1), st; b - d / 2 в st(0)

fld1

fpatan; arctg(b - d / 2) в st(0)

fst inter\_solved3

ret

res endp

end

**Висновки**

При виконанні роботи я попрактикувався з прийомами модульного програмування, навчився працювати з методами звернення до процедур і передачі в них параметрів.