

Архитектуры вычислительных систем

Домашнее задание 1

Кириченко Виктория БПИ198

1 консольное приложение (Hello world)

Compile



3 passes, 0.1 seconds, 2048 bytes.

Hello World!

```
format PE console                ; Win32 portable executable console format
entry _start                      ; _start is the program's entry point

include 'win32a.inc'

section '.data' data readable writable ; data definitions

hello db "Hello World!", 0
stringformat db "%s", 0ah, 0

section '.code' code readable executable ; code

_start:
    invoke printf, stringformat, hello ; call printf, defined in msvcrt.dll
    invoke getchar                      ; wait for any key
    invoke ExitProcess, 0               ; exit the process

section '.imports' import data readable ; data imports

library kernel, 'kernel32.dll', \
    msvcrt, 'msvcrt.dll' ; link to kernel32.dll, msvcrt.dll

import kernel, \
    ExitProcess, 'ExitProcess' ; import ExitProcess from kernel32.dll

import msvcrt, \
    printf, 'printf', \
    getchar, '_fgetchar' - ; import printf and getchar from msvcrt.dll
```

- 1) Консольное приложение взято с канала byte++
- 2) Имеет подключенный компонент win32.inc
- 3) В данной программе тестовых данных нет
- 4) Сложность: уровень 1
- 5) Используется вывод текста в консоль, ожидание нажатия клавиши и выход из программы

2 консольное приложение

```
format PE console

entry start

include 'win32a.inc'

section '.data' data readable writable

    formatStr db '%s', 0
    formatNum db '%d', 0

    name rd 2
    age rd 1

    wn db 'What is your name? ', 0
    ho db 'How old are you? ', 0
    hello db 'Hello %s, you are %d years old', 0

    NULL = 0

section '.code' code readable executable

start:
    push wn
    call [printf]

    push name
    push formatStr
    call [scanf]

    push ho
    call [printf]

    push age
    push formatNum
    call [scanf]

    push [age]
    push name
    push hello
    call [printf]

    call [getch]

    push NULL

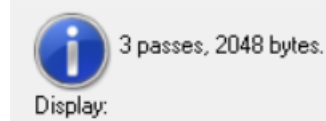
section '.idata' import data readable

library kernel, 'kernel32.dll', \
    msvcrt, 'msvcrt.dll'

import kernel, \
    ExitProcess, 'ExitProcess'

import msvcrt, \
    printf, 'printf', \
    getch, '_getch', \
    scanf, 'scanf'
```

Compile



- 1) Запрос имени и возраста пользователя и приветствие его
- 2) Работает на различных входных данных (обозначенных на скриншотах)
- 3) Сложность: уровень 2
- 4) Используется: вывод в консоль автоматического текста, текста, введенного пользователем, ожидание нажатия клавиши и завершение программы, после завершения всех функций

2 консольное приложение (скриншоты)

```
What is your name? _
```

```
What is your name? Vika  
How old are you?
```

```
What is your name? Vika  
How old are you? 18  
Hello Vika, you are 18 years old
```

- 1) Запрос имени
- 2) Вывод имени и запрос возраста
- 3) Вывод возраста, приветствие и уведомление о возрасте, ожидание нажатия любой клавиши

```
What is your name? 45  
How old are you? +  
Hello 45, you are 0 years old_
```

- Возможность ввода любого имени
- При некорректном вводе возраста, возраст считается по умолчанию равным нулю

3 консольное приложение (калькулятор)

```
format PE console

entry Start

include 'win32a.inc'

section '.data' data readable writable

    strA db 'Enter A: ', 0
    strB db 'Enter B: ', 0
    strOp db 'Enter operation: ', 0

    resStr db 'Result: %d', 0
    resMod db '/%d', 0

    spaceStr db ' %d', 0
    emptyStr db '%d', 0

    infinity db 'infinity', 0
    point db ',', 0

    A dd ?
    B dd ?
    C dd ?

    NULL = 0

section '.code' code readable executable

Start:
    push strA
    call [printf]

    push A
    push spaceStr
    call [scanf]

    push strB
    call [printf]

    push B
    push spaceStr
    call [scanf]

    push strOp
```

```
    push strOp
    call [printf]

    call [getch]

    cmp eax, 43
    jne notAdd
    mov ecx, [A]
    add ecx, [B]

    push ecx
    push resStr
    call [printf]

    jmp finish
notAdd:

    cmp eax, 45
    jne notSub
    mov ecx, [A]
    sub ecx, [B]

    push ecx
    push resStr
    call [printf]
notSub:

    cmp eax, 42
    jne notMul
    mov ecx, [A]
    imul ecx, [B]

    push ecx
    push resStr
    call [printf]

    jmp finish
notMul:

    cmp eax, 37
    jne notMod
    mov eax, [A]
    mov ecx, [B]
    mov edx, 0
```

```
    cmp [B], 0
    jne notNullDiv
    push infinity
    call [printf]

    jmp finish

notNullDiv:

    div ecx
    mov [C], edx

    push eax
    push resStr
    call [printf]

    push [C]
    push spaceStr
    call [printf]

    push [B]
    push resMod
    call [printf]

    jmp finish
notMod:

    cmp eax, 47
    jne notDiv
    mov eax, [A]
    mov ecx, [B]
    mov edx, 0

    cmp [B], 0
    jne notNullDiv1
    push infinity
    call [printf]

    jmp finish
```

- 1) Простой калькулятор с запросом двух чисел и операции, которую необходимо с ними сделать
- 2) Работает на различных входных данных(см. след слайд)
- 3) Сложность: уровень 3
- 4) Используются: вывод автоматического и пользовательского текста в консоль, сложение/умножение/деление введенных чисел

3 консольное приложение (скриншоты)

```
Enter A:
```

```
Enter A: 15  
Enter B:
```

```
Enter A: 15  
Enter B: 3  
Enter operation:
```

```
Enter A: 15  
Enter B: 3  
Enter operation: Result: 45
```

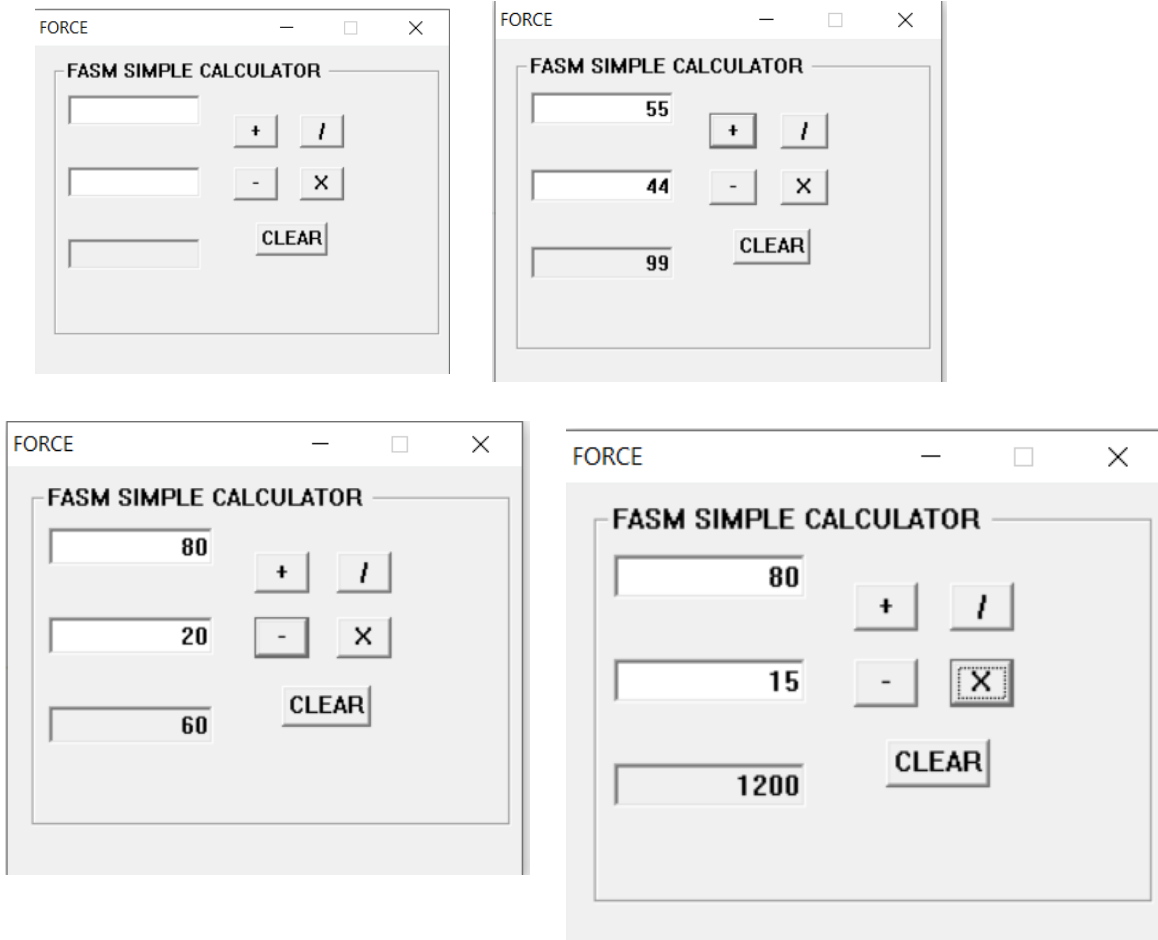
- 1) Запрос первого числа
- 2) Вывод первого числа и запрос второго
- 3) Вывод второго числа и запрос необходимой операции
- 4) Вывод результата операции

```
Enter A: -  
Enter B: 8  
Enter operation: Result: 0,000_
```

```
Enter A: д  
Enter B: Enter operation: Result: 0
```

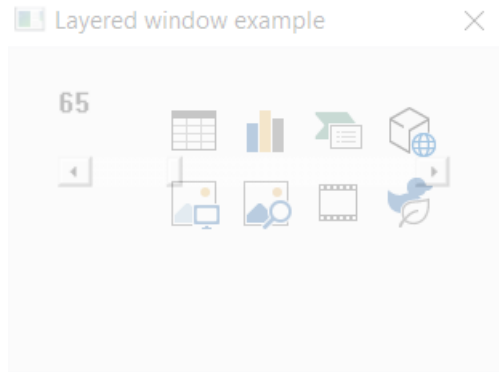
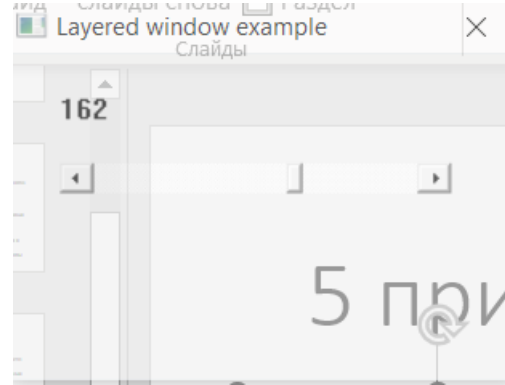
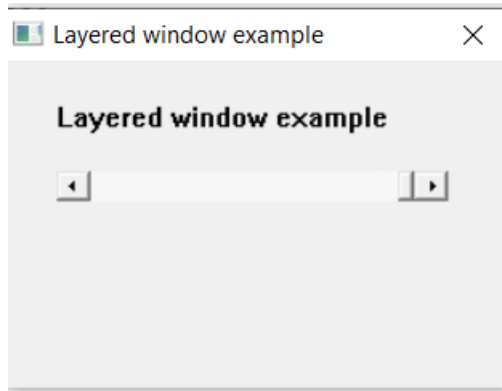
- При введении операции вместо числа, число засчитывается компилятором и запрашивается второе число, но при любой операции ответ будет 0,000
- При введении буквы, компилятор не засчитывает ввод и выводит сразу ответ 0, без возможности ввода второго числа
- При некорректном вводе операции, консольное приложение закрывается

4 приложение (калькулятор оконное)



- 1) Оконный калькулятор с простыми операциями
- 2) Корректно работает на любых входных данных (при некорректных издает соответствующее уведомление)(при попытке совершить разные операции, не меняя чисел, программа закрывается)
- 3) Сложность: уровень 4
- 4) Используется(см. прилож3) + отрисовка кнопок и textboxes

5 приложение



- 1) Оконное приложение для изменения прозрачности приложения
- 2) Работает без входных данных, изменение положения ползунка, не ломает программу
- 3) Сложность: уровень 5