# Лабораторне заняття №4

з навчальної дисципліни

Спеціалізовані мови програмування

**Python (advanced)**

на тему:

# ФУНКЦІЇ

# Мета роботи

Ознайомитися з особливостями побудови функцій в мові програмування Python 3

# Хід роботи

- Самостійно на ПК реалізувати програмний код наведений нижче

# ПРИКЛАД №1

```
# Оголошення функції hello_world
def hello_world():
    print('Hello, World!')
    print('Hello, World!')
    print('Hello, World!')
    print('Hello, World!')
    print('Hello, World!')
    print('Hello, World!')
    print('Hello, World!')

# Визов функції hello_world
hello_world()


def pass_function():
    a = int(input('>>> '))

pass_function()
```

# ПРИКЛАД №2

```
# limit - формальний параметр функції print_numbers

def print_numbers(limit):
    for i in range(limit):
        print(i)

# Тут викликається функція print_numbers, а її формальний
# параметр limit замінюється фактичним параметром 10

print_numbers(10)

range_input = int(input('>>> '))

print_numbers(range_input)
```

# ПРИКЛАД №3

```
# Функція з минулого прикладу
def print_numbers(limit):
    for i in range(limit):
        print(i)


y = int(input('Введите y: '))
print_numbers(y)

# Читаємо ввод користувача за допомогою стандартної
# функції input, конструюємо з нього число
# за допомогою стандартної функції int і записуємо в змінну n
n = int(input('... '))

# Викликаємо функцію print_numbers з фактичним параметром n
print_numbers(n)
```

# ПРИКЛАД №4

```
# Функція з минулого прикладу
def print_numbers(limit):
    for i in range(limit):
        print(i)
    print("function is completed!")

# Будь-яку логічну завершену дію потрібно поміщати в функцію
def main():
    n = int(input('Введите n: '))
    res = print_numbers(n)
    print(res)
    print('n is ' , n)
    exit  = input("y/n ?")
    if exit == 'y':
        return


# Виклик головної функції
main()
```

# ПРИКЛАД №5

```
def a():
    print('Hello, World!')

def b(sdf):
    print(sdf)
# Головну функцію бажано викликати так
# (таким чином функція буде викликана тільки, якщо
# даний файл був запущений як головний;
# це важливо для додатків, що складаються з декількох модулів)
if __name__ == '__main__':
    a()
    b('sdfdsf')
```

# ПРИКЛАД №6

```
def add_numbers(a, b):
    return a + b  # повернення суми параметрів

x = add_numbers(2, 3)
print(x)
first_num = int(input('some number >>> '))
second_num = int(input('some number >>> '))
result  = add_numbers(first_num , second_num)

print(result)
```

# ПРИКЛАД №7

```
def procedure():
    return 'I return nothing... Or I do?'

value = procedure()
print('Результат функції:', value)
```

# ПРИКЛАД №8

```
# Ця функція повертає аргумент, помножений на два,
# якщо він від’ємний, або аргумент, помножений на три,
# якщо він більше або дорівнює нулю
def function(x):
    if x < 0:
        return x * 2
    else:
        return x * 3


def main():
    # Виведення значень функції в діапазоні [-3, 3]
    for i in range(-3, 4):
        y = function(i)
        print('function(', i, ') = ', y, sep='')



# if __name__ == '__main__':
main()
```

# ПРИКЛАД №9

```
def hello(name):
    # Якщо ім'я пусте, виходимо з функції
    if not name:
        return
    print('Hello, ', name, '!', sep='')


hello('Alex')
hello('')
hello('Python')
```

# ПРИКЛАД №10

```
def add(a, b  ,c):
    return (a + b) / c


def sub(a, b):
    return a - b


# Виклик функції може бути частиною виразу
res = add(2, 3, 1.5) +  sub(2, 3)

print(res)  # => print((2 + 3)/1.5 + (2 - 3))
```

# ПРИКЛАД №11

```
# Функція, яка приймає три аргументи
def info(object, color, price):
    print('Объект:', object)
    print('Цвет:', color)
    print('Цена:', price)
    print('--------------------------------')


# передача параметрів в прямому порядку
info('red', 'pen', 1)

# передача параметрів в довільному порядку
info(price=5, object='cup', color='orange')

# # # можна змішувати обидва способи, але спочатку повинні йти параметри,
# # # які передаються в прямому порядку
info('cup', price=10, color='black')
```

# ПРИКЛАД №12

```
# Якщо параметр name не заданий, тоді name = 'undefined'
def hello(name='undefined'):
    print('Hello, ', name, '!', sep='')



hello('Python')
hello('Anton')
hello('Nastya')
hello('Sasha')
print()
hello()
```

# ПРИКЛАД №13

```
"""
В цьому прикладі розглядаються документаційні рядки
"""


def function():
    """ Рядок, що стоїть на самому початку функції (а також модуля, класу або методу),
     відіграє роль особливого виду коментарів – рядка документації (docstring).
    """
    print('function called')
function()
print(function.__doc__)
```

# ПРИКЛАД №14

```
# Функції bin, oct і hex повертають
# дане число у відповідних системах числення

number = int(input('Введіть число: '))
print('Двійкова система:        ', bin(number))
print('Восьмерична система:     ', oct(number))
print('Шістнадцятерична система:', hex(number))
```

# ПРИКЛАД №15

```
# reversed дозволяє обходити послідовність в зворотньому порядку
for i in reversed(range(5)): # range(5)  = [0 ,1,2,3,4] => reverserd([0 ,1,2,3,4]) = [4,3,2,1,0]
    print(i)
```

# ПРИКЛАД №16

```
a = 5
b = -7 + 100
c = 2

# мінімальне значення
print(min(a, b, c))

# максимальне значення
print(max(a, b, c))
print(sum(range(1200)))
```

# ПРИКЛАД №17

```
def outer_function():
    # локальне оголошення функції
def inner_function():
        print('Внутрішня функція')
    print('Зовнішня функція')
    # виклик функції
    inner_function()

# виклик зовнішьої функції
outer_function()

# inner_function()  # помилка, тут ця функція недоступна
```

# ПРИКЛАД №18

```
def function():
    print(var)  # отримання доступу до глобальної змінної

var = 'глобальна змінна'
function()
```

# ПРИКЛАД №19

```
def function():
    # визначення локальної змінної
    var = 'локальна змінна'
    # виведення значення локальної змінної на екран
    print(var)

# визначення глобальної змінної
var = 'глобальна змінна'
function()

# виведення значення глобальної змінної на екран
print(var)
```

# ПРИКЛАД №20

```
def function():
    # global вказує, що необхідно отримувати доступ до глобальної змінної
    # var, а не створювати нову локальну при спробі що-небудь їй присвоїти
    global var
    # виведення значення глобальної змінної на екран
    print(var)
    # зміна значення глобальної змінної
    var = 'нове значення'
    # виведення значення глобальної змінної на екран
    print(var)

var = 'глобальна змінна'
print(var)
function()
print(var)
```

# ПРИКЛАД №21

```
def fun(a):
    for elem in a:
        print(elem)

def add(a,b):
    return a + b

def sub(a,b):
    return a - b

def div(a,b):
    if b == 0:
        print('ділення на нуль')
        b = float(input(' >>> '))
        return  a / b
    else:
        return a / b

def mul(a,b):
    return a * b
```

# ПРИКЛАД №21 (продовження)

```
ready = True
while ready:
    first_num  = float(input('>>> '))
    operation = input(' + - * / ')
    second_num  = float(input('>>> '))
    res  = None
    if operation == '+':
        res = add(first_num , second_num)
    elif operation == '-':
        res = sub(first_num , second_num)
    elif operation == '/':
        res = div(first_num , second_num)
    elif operation == '*':
        res = mul(first_num , second_num)
    else:
        print('Дана операція відсутня в нашому калькуляторі(((')
```

# ПРИКЛАД №21 (продовження)

```
if res is not None:
        print('{} {} {} = {}'.format(first_num , operation , second_num , res))
    else:
        print('result is NONE!!!!')

    print('Бажаєте продовжити? так/ні , т/н  , yes/no , y/n ')
    response  = input(" >>> ").replace(' ' , '').lower()
    if response == 'ні' or response == 'н' or response == 'no' or response == 'n':
        print('Спасиб что считали с нами :) ')
        ready = False
    elif response == 'так' or response == 'т' or response == 'yes' or response == 'y':
        ready = True
    else:
        print('Некоректний ввод))')

print("====================")
```

# ПРИКЛАД №22

```
def function(c, d):
    # a, b – глобальні змінні; c, d – локальні змінні
    global a, b
    # зміна значення глобальної змінної
    a = 5
    # зміна значення глобальної змінної
    b = 7
    # створення локальної змінної з тим же ім'ям, що і у глобальній
    c = 10
    # створення локальної змінної з тим же ім'ям, що і у глобальній
    d = 12

a, b, c, d = 1, 2, 3, 4  # множинне присвоювання
print(a, b, c, d)  # 1 2 3 4
function(c, d) # function(3,4)
print(a, b, c, d)  # 5 7 3 4
```

# ПРИКЛАД №23

```
def outer_function():
    var = 8  # створення локальної змінної var

    def inner_function():
        # вказує, що необхідно використовувати змінну з зовнішньої функції
        nonlocal var
        print(var)  # 8
        var = 10

    print(var)  # 8
    inner_function()  # виклик внутрішньої функції
    print(var)  # 10


# створення глобальної змінної var
var = 0
print(var)  # 0
outer_function()
print(var)  # 0
```

# ПРИКЛАД №24

```
def outer_function():
    var = 8  # створення локальної змінної var

    def inner_function():
        # вказує, що необхідно використовувати глобальну змінну
        global var
        print(var)  # 0
        var = 10

    print(var)  # 8
    inner_function() #  виклик внутрішньої функції
    print(var)  # 8


# створення глобальної змінної var
var = 0
print(var) # 0
outer_function()
print(var) # 10
```

# ПРИКЛАД №25

```
# Факторіалом числа n (позначається n!) Називається добуток
# всіх натуральних чисел від 1 до n включно. 5! = 5 * 4 * 3 *2 *1 * 1
# 0! = 1 1! = 1 5! = 5* 4!(4 * (3!3 * 2!(2 * 1!(1 * 0!(0! = 1)))))


# приклад рекурсивної функції
def factorial(n):
    if n == 0:
        return 1  # умова виходу
    else:
        return n * factorial(n - 1)  # рекурсивний виклик


# обчислення факторіала числа
x = factorial(int(input("Enter the number:: ")))
print(x)
```

# ПРИКЛАД №26

```
# Числа Фібоначчі - послідовність, в якій перші два числа дорівнюють одиниці,
# а всі наступні - сумою двох попередніх. 1 1 2 3 5 8 13 21 34 55


def fib(n):
    if n == 1 or n == 2:  # умова виходу
        return 1
    else:
        # рекурсивний виклик fib(3) = > fib(2) + fib(1) , fib(4) => fib(3) (fib(2) + fib(1)) + fib(2)
        return fib(n - 1) + fib(n - 2)


index = int(input('Введіть номер числа Фібоначчі: '))
print(fib(index))
```

# Завдання на самостійну роботу

# Оформити звіт

Заняття закінчено.
Дякую за увагу!