Міністерство освіти і науки України

Національний технічний університет «ХПІ»

Навчально-науковий інститут комп’ютерних наук та інформаційних

технологій

Кафедра комп’ютерної інженерії та програмування

**ЗВІТ**

з практичного заняття №6

з дисципліни «Проектування серверних застосунків»

Виконала: студентка групи КН-919д

Федюкіна Поліна Олегівна

Перевірив: старший викладач

Лисиця Дмитро Олександрович

Харків - 2022

**Тема**: Розробка функцій та модулів користувач.

**Мета**: Розгляд особливостей розробки функцій та модулів користувача.

**Хід роботи**

Таблиця 1.1 – Індивідуальне завдання для варіанту 5(25)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Варіант | Операція | | Вид значень | Тип |
| 5(25) | Визначення середнього значення елементів списку | Букви латиниці | | Рядки, списки з елементами у вигляді чисел |

1. Розробимо програму мовою Python, яка виконує наступні функції:
2. Визначена і виконана функція func1() з аргументами у вигляді списку чисел (цілих та з плаваючою точкою);
3. Створено словник a\_dict (числом елементів не менше 8), ключі якого іменуються довільно. При цьому окремі ключі (числом не менше трьох) повинні мати однакове значення;
4. Визначено функцію func2(), яка має два аргументи, перший у вигляді словника, другий вказує значення ключа словника. Функція func2() повертає список ключів словника, значення яких збігаються із значеннями другого аргументу;
5. Перевірено роботу функції func2(), при виклику якої як перший аргумент заданий словник a\_dict, а як другий аргумент – значення, яке мають кілька ключів словника;
6. Створено список a\_list;
7. Визначено функцію func3(), яка перетворює кожен елемент заданого списку a\_list на ціле число (механізм перетворення – на розсуд студента);
8. Виконано з використанням функції func3() та методу sort() сортування елементів списку a\_list;
9. Заданий рядок str\_code, що містить невеликий фрагмент коду мовою Python і отриманий скомпільований за допомогою вбудованої функції compile() код – comp\_code;
10. Для програми та однієї з функцій програми визначено безліч глобальних імен та безліч локальних імен;

10) Створити власний модуль , що містить довільну функцію (або функції), записати цей модуль у бібліотеку Lib, завантажити його та викликати на виконання одну або кілька функцій модуля.

Пишемо код наведений нижче:

import mymod

def func1( my\_list ):

sum = 0

for i in my\_list:

if(type(i) == int or type(i) == float):

sum += i

return sum / len(my\_list)

def func2(input\_dict, value):

return\_list = []

for key, val in input\_dict.items():

if val == value: return\_list.append(key)

return return\_list

def func3(input\_list):

return\_list = []

for item in input\_list:

if(type(item) != list):

return\_list.append(int(item))

else:

for item2 in item:

return\_list.append(item2)

return return\_list

my\_list = [1, 2, 3, 4, 6]

print(func1(my\_list))

print()

a\_dict = {

1: 'a',

2: 'b',

3: 'c',

4: 'd',

5: 'e',

6: 'f',

7: 'g',

8: 'h',

9: 'a',

10: 'b',

11: 'a'

}

print(func2(a\_dict, 'a'))

print()

a\_list = list(['1123', '11', list([1, 2, 3])])

res\_list = func3(a\_list)

print(res\_list)

res\_list.sort()

res\_list.reverse()

print(res\_list)

print()

str\_code = "print(\'Hello, World!\')"

comp\_code = compile(str\_code,'<string>' , 'exec');

exec(comp\_code)

mymod.HelloWorld()

Та код зовнішнього модулю:

def HelloWorld():

print('Hello, World! From module')

Результат виконання коду наведений нижче на рис. 6.1

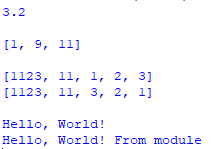


Рисунок 6.1 – Результат виконання програми

**Висновки**: під час виконання цієї практичної роботи розглянула способи та особливості розробки функцій та модулів користувача. Створила програму та підключила до неї зовнішній модуль.