Міністерство освіти і науки України Національний університет

„ Львівська політехніка ”

Кафедра ЕОМ



Звіт

з лабораторної роботи № 8

з дисципліни: “Кросплатформні засоби програмування”

ФАЙЛИ ТА ВИКЛЮЧЕННЯ У PYTHON

Варіант 24

Виконав:

Ст. групи КІ-301

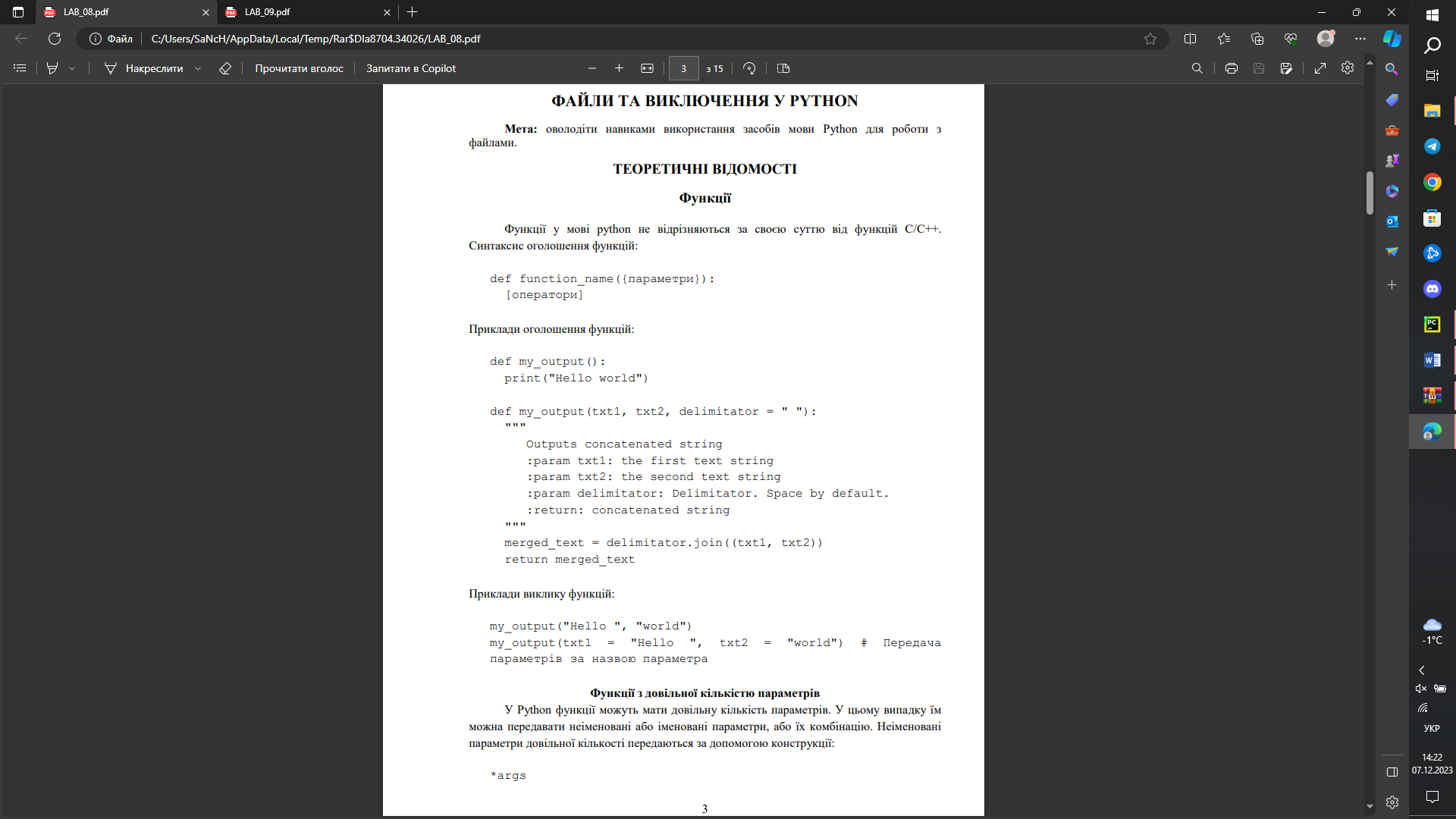
Чоланюк О.Ю.

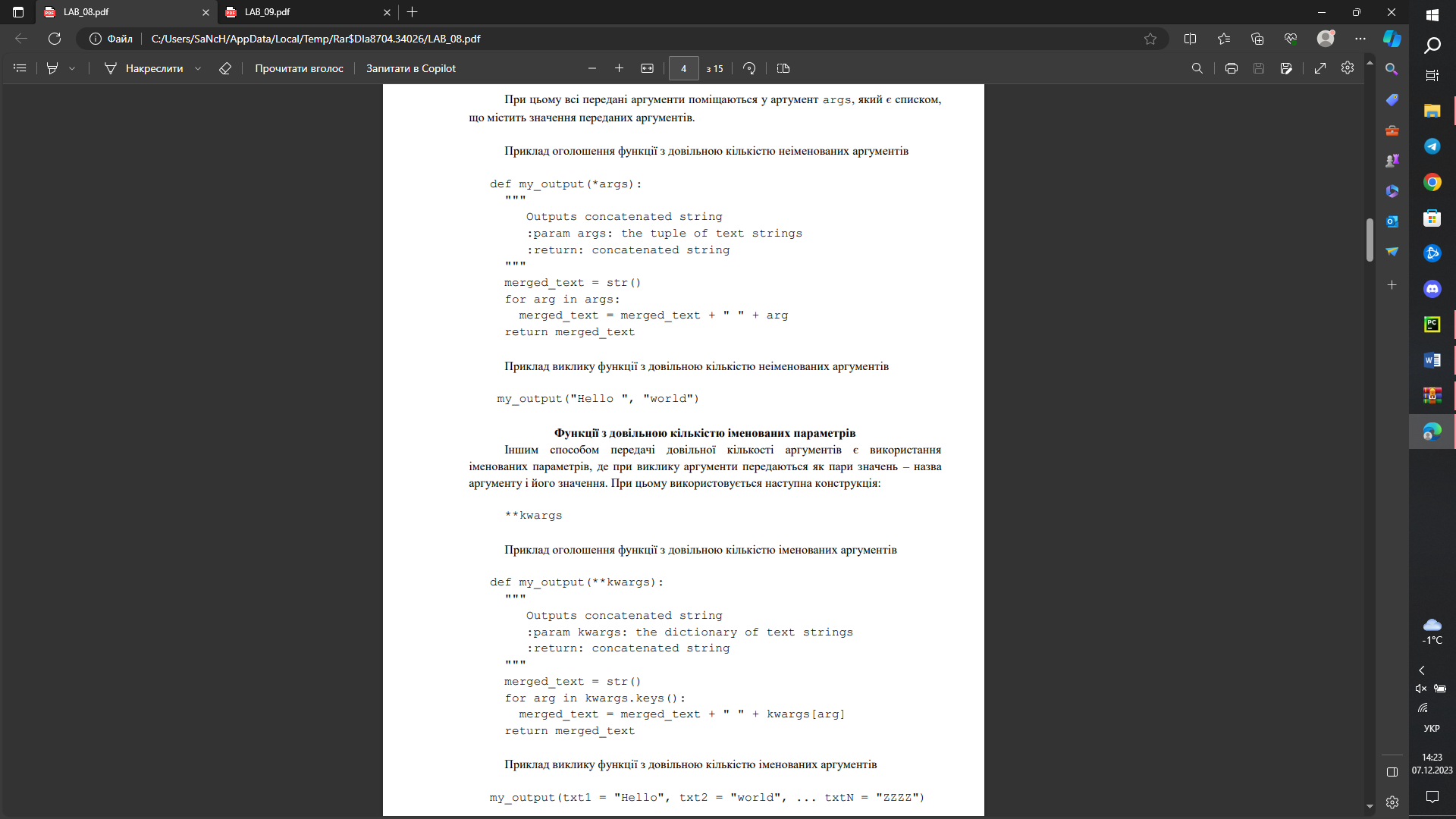
Прийняв:

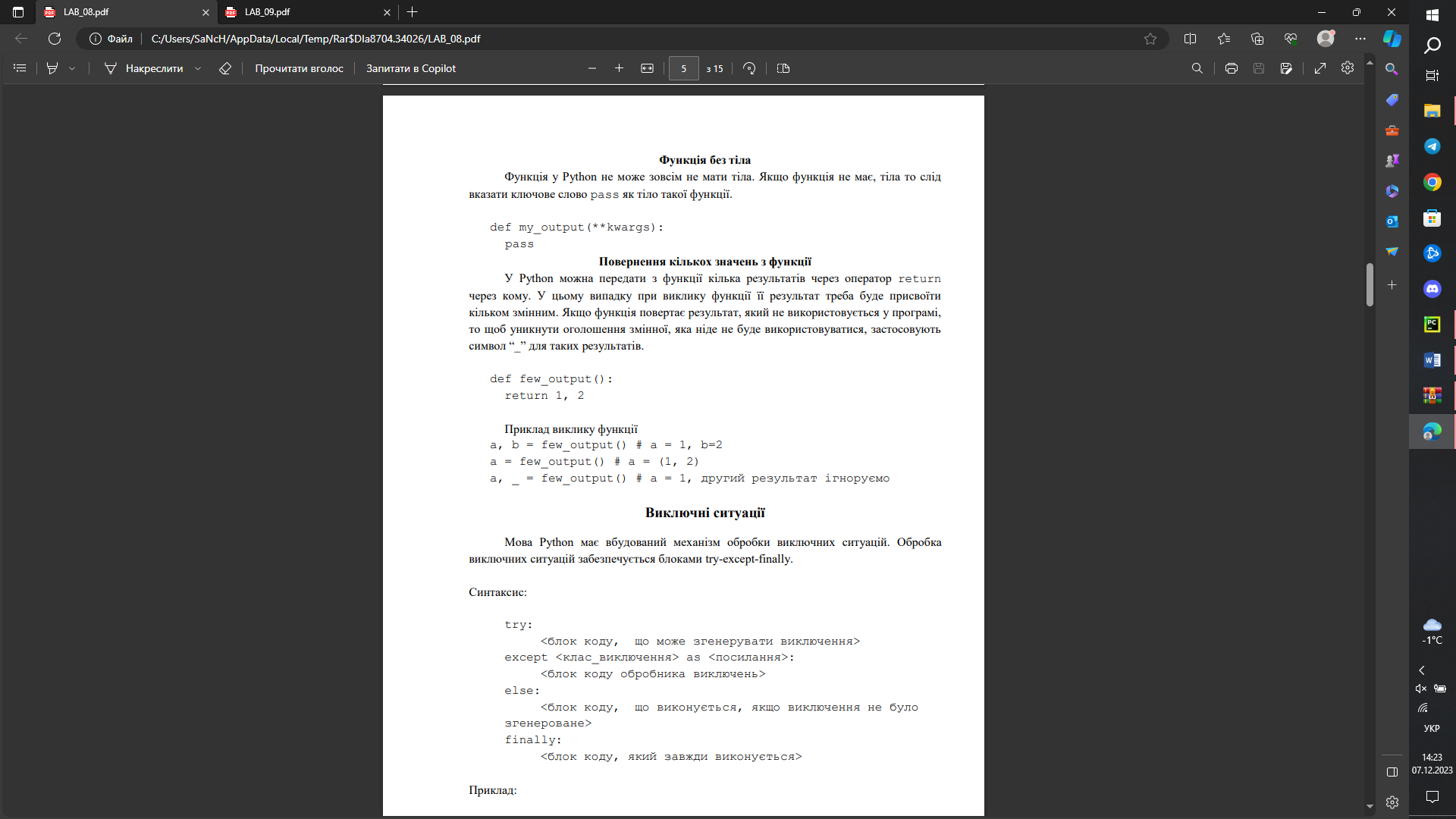
Викладач

Майдан М.В.

Львів 2023







ЗАВДАННЯ 1.

Написати та налагодити програму на мові Python згідно варіанту. Програма має задовольняти наступним вимогам:

• програма має розміщуватися в окремому модулі;

• програма має реалізувати функції читання/запису файлів у текстовому і двійковому форматах результатами обчислення виразів згідно варіанту;

• програма має містити коментарі.

2. Завантажити код на GitHub згідно методичних вказівок по роботі з GitHub.

3. Скласти звіт про виконану роботу з приведенням тексту програми, результату її виконання та фрагменту згенерованої документації та завантажити його у ВНС.

4. Дати відповідь на контрольні запитання

Варіант 24

y=sin(x-9)/(x-cos(2x))

import math  
  
class CalcException(Exception):  
 pass  
  
class Equations:  
 def calculate(self, x):  
 self.validate\_input(x)  
  
 rad = math.radians(x)  
  
 try:  
 y = math.sin(x - 9) / (x - math.cos(2 \* x))  
 print(y)  
 if math.isnan(y) or math.isinf(y):  
 raise ArithmeticError()  
 except ArithmeticError:  
 raise CalcException("Причина виліту: помилка в розрахунку формули")  
  
 return y  
  
 def validate\_input(self, x):  
 if math.isinf(x) or math.isnan(x):  
 raise CalcException("Причина виліту: некоректне введення X")  
  
 if x - math.cos(2 \* x) == 0:  
 raise CalcException("Причина виліту: ділення на нуль")

import struct  
  
class FileWrite:  
 @staticmethod  
 def write\_result\_to\_txt(file\_name, number):  
 with open(file\_name, 'w') as file:  
 file.write(f'{number} ')  
  
 @staticmethod  
 def write\_result\_to\_bin(file\_name, number):  
 with open(file\_name, 'wb') as file:  
 file.write(struct.pack('d', number))  
  
 @staticmethod  
 def read\_result\_from\_txt(file\_name):  
 with open(file\_name, 'r') as file:  
 return float(file.read())  
  
 @staticmethod  
 def read\_result\_from\_bin(file\_name):  
 with open(file\_name, 'rb') as file:  
 return struct.unpack('d', file.read())[0]

from file\_write import FileWrite  
from equations import Equations  
  
def main():  
 try:  
 x = float(input("Введіть x: "))  
 y = calculate\_y(x)  
  
 txt\_file\_name = input("Введіть назву текстового файлу: ")  
 bin\_file\_name = input("Введіть назву бінарного файлу: ")  
  
 FileWrite.write\_result\_to\_txt(txt\_file\_name, y)  
 FileWrite.write\_result\_to\_bin(bin\_file\_name, y)  
  
 print("Результат було успішно записано в файли.")  
  
 result\_from\_txt = FileWrite.read\_result\_from\_txt(txt\_file\_name)  
 print("Результат з текстового файлу: ", result\_from\_txt)  
  
 result\_from\_bin = FileWrite.read\_result\_from\_bin(bin\_file\_name)  
 print("Результат з бінарного файлу: ", result\_from\_bin)  
  
 except (ValueError, CalcException) as e:  
 print(e)  
  
def calculate\_y(x):  
 try:  
 equations = Equations()  
 return equations.calculate(x)  
 except CalcException as e:  
 print(e)  
 return float('nan')  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

