**Звіт про виконання практичних завдань до лекцій з курсу Технології програмування на мові Python**

Звіт до Теми №1

Функції та змінні

Під час виконання практичного завдання до Теми №1 було надано варіанти рішення до наступних задач:

**Перетворення рядка**

Необхідно рядок, що має вигляд "abcdefg123" перетворити наступним чином "321gfedcba", вважаючи сталою довжину рядку в 10 символів.

Хід виконання завдання:

Створюємо наш рядок. Створюємо другий рядок в який буде записано розвернутий рядок. Створюємо цикл який повторюється 10 разів. В циклі беремо останній символ нашого рядка і додаємо його до другого рядка. Виводимо розвернутий рядок.

Текст програми:

str= "abcdefg123"

str1=''

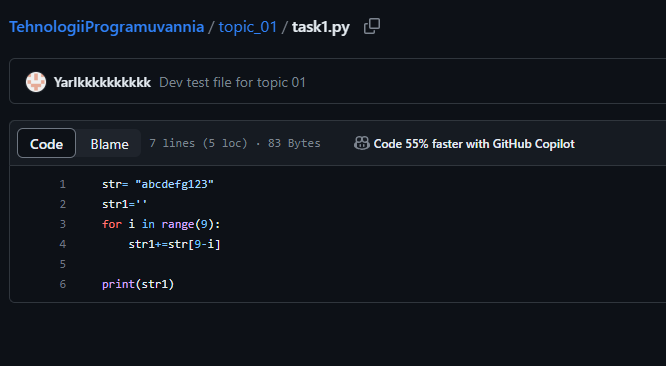
for i in range(9):

str1+=str[9-i]

print(str1)

Посилання на github: [TehnologiiProgramuvannia/topic\_01/task1.py at main · YarIkkkkkkkkkk/TehnologiiProgramuvannia (github.com)](https://github.com/YarIkkkkkkkkkk/TehnologiiProgramuvannia/blob/main/topic_01/task1.py)

Знімок екрану з посилання на github:

**Виконання функцій**

Необхідно виконати тест для функцій strip, capitalize, title, upper, lower.

Хід виконання завдання:

Створюємо наш рядок та виводимо його. Функція strip видаляє пробіли на початку та кінці рядка. Функція capitalize робить заголовною першу літеру. Функція title робить першу літеру в кожному слові заголовною. Функція upper змінює регістр всіх літер на заголовні. Функція lower робить регістр всіх літер на маленькі.

Текст програми:

str=' banaNa '

print(str)

print(str.strip())

print(str.strip().capitalize())

print(str.title())

print(str.upper())

print(str.lower())

Посилання на github: [TehnologiiProgramuvannia/topic\_01/task2.py at main · YarIkkkkkkkkkk/TehnologiiProgramuvannia (github.com)](https://github.com/YarIkkkkkkkkkk/TehnologiiProgramuvannia/blob/main/topic_01/task2.py)

Знімок екрану з посилання на github:



**Виконання функцій**

Необхідно створити функцію яка рахує дискримінант.

Хід виконання завдання:

Створюємо функцію яка приймає три аргументи та повертає результат виконання формули для дискримінанту. Зчитуємо з клавіатури наші цифри. Функцією print викликаємо нашу функцію та вказуємо три аргументи які вона використовуєю.

Текст програми:

def diskrininant(a,b,c):

return int(b)\*\*2-4\*int(a)\*int(c)

a,b,c=input('Введіть a b та c ').split()

print(diskrininant(a,b,c))

Посилання на github: [TehnologiiProgramuvannia/topic\_01/task3.py at main · YarIkkkkkkkkkk/TehnologiiProgramuvannia (github.com)](https://github.com/YarIkkkkkkkkkk/TehnologiiProgramuvannia/blob/main/topic_01/task3.py)

Знімок екрану з посилання на github:

Висновок

Під час виконання завдання я ознайомився з поняттями функції та змінної. Навчився використовувати їх на практиці.

Звіт до Теми №2

Умовний перехід

Під час виконання практичного завдання до Теми №2 було надано варіанти рішення до наступних задач:

**Завдання 1**

Написати функцію пошуку коренів квадратного рівняння використовуючи функцію розрахунку дискримінанту з попередньої теми та умовні переходи.

Хід виконання завдання:

Пишемо функцію яка знаходить дискримінант. Далі пишемо функцію яка знаходить корені рівняння, якщо дискримінант більший нуля знаходить два корені, якщо він менший нуля то рівняння не має розв’язку, інакше знаходить один корінь. Далі ми вводимо з клавіатури наші змінні та виводимо результат виконання функції.

Текст програми:

def disc(a,b,c):

D=int(b)\*\*2-4\*int(a)\*int(c)

return D

def corin (D,a,b) :

if D>0:

x1=(-int(b)+D\*\*0.5)/2\*int(a)

x2=(-int(b)-D\*\*0.5)/2\*int(a)

return(x1,x2)

elif D<0:return('None')

else:return((-int(b)+D\*\*0.5)/2\*int(a))

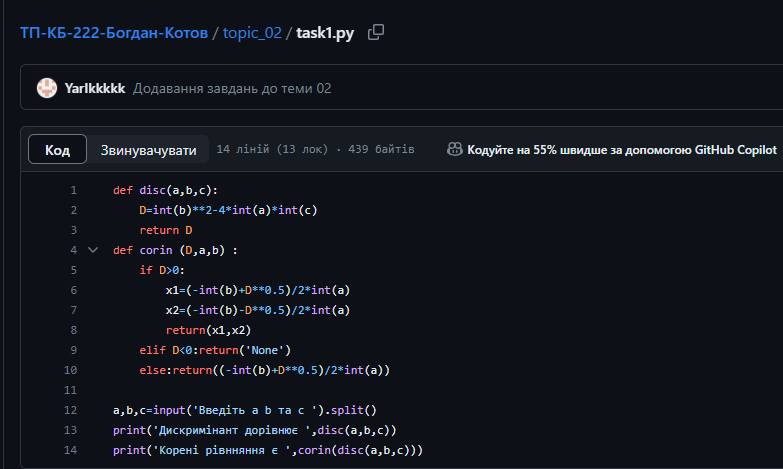
a,b,c=input('Введіть a b та c ').split()

print('Дискримінант дорівнює ',disc(a,b,c))

print('Корені рівнняння є ',corin(disc(a,b,c),a,b))

Посилання на github: [TP-KB-222-Bohdan-Kotov/topic\_02/task1.py at main · YarIkkkkkkkkkk/TP-KB-222-Bohdan-Kotov (github.com)](https://github.com/YarIkkkkkkkkkk/TP-KB-222-Bohdan-Kotov/blob/main/topic_02/task1.py)

Знімок екрану з посилання на github:



**Завдання 2**

Написати програму калькулятор використовуючи if else конструкцію.

Хід виконання завдання:

Зчитуємо з клавіатури наші числа та функцію. Потім створюємо функції додавання, віднімання, ділення, множення. Далі за допомогою if else конструкції ми перевіряємо яка саме це функція та виконуємо її і виводимо результат на екран.

Текст програми:

a=int(input('Введіть перше число '))

b=int(input('Введіть друге число '))

f=input('Введіть функцію ')

def plus(a,b):

return a+b

def minus(a,b):

return a-b

def mnoj(a,b):

return a\*b

def dil(a,b):

if b==0:

return 'Ділення на нуль'

else:

return a/b

if f=='+':

print(plus(a,b))

elif f=='-':

print(minus(a,b))

elif f=='/':

print(dil(a,b))

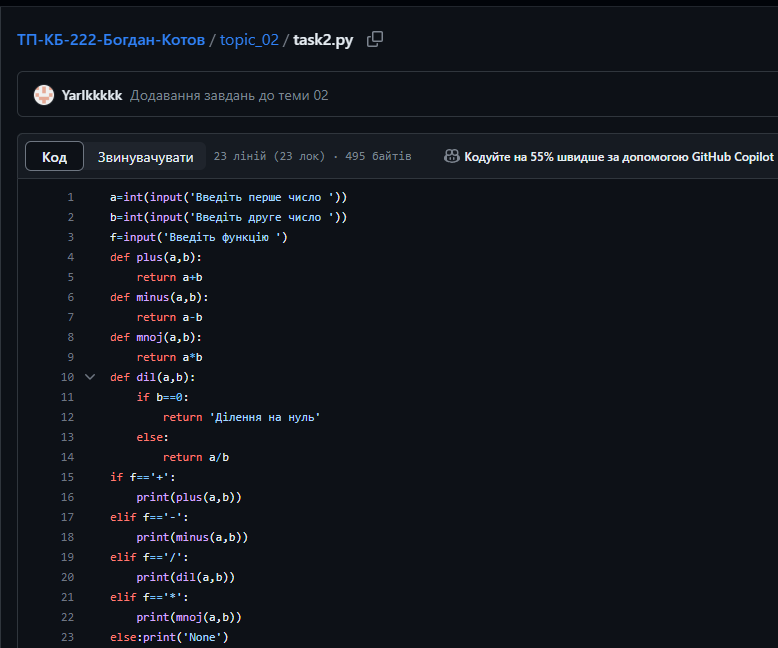
elif f=='\*':

print(mnoj(a,b))

else:print('None')

Посилання на github: [TP-KB-222-Bohdan-Kotov/topic\_02/task2.py at main · YarIkkkkkkkkkk/TP-KB-222-Bohdan-Kotov (github.com)](https://github.com/YarIkkkkkkkkkk/TP-KB-222-Bohdan-Kotov/blob/main/topic_02/task2.py)

Знімок екрану з посилання на github:



**Завдання 3**

Написати програму калькулятор використовуючи match конструкцію.

Хід виконання завдання:

Зчитуємо з клавіатури наші числа та функцію. Потім створюємо функції додавання, віднімання, ділення, множення. Далі за допомогою match конструкції ми перевіряємо яка саме це функція та виконуємо її і виводимо результат на екран.

Текст програми:

a=int(input('Введіть перше число '))

b=int(input('Введіть друге число '))

f=input('Введіть функцію ')

def plus(a,b):

return a+b

def minus(a,b):

return a-b

def mnoj(a,b):

return a\*b

def dil(a,b):

if b==0:

return 'Ділення на нуль'

else:

return a/b

match f:

case '+':print(plus(a,b))

case '-': print(minus(a,b))

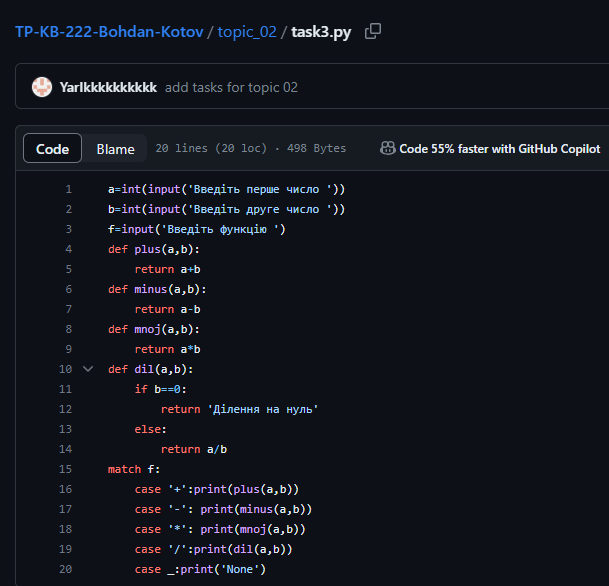
case '\*': print(mnoj(a,b))

case '/':print(dil(a,b))

case \_:print('None')

Посилання на github: [TP-KB-222-Bohdan-Kotov/topic\_02/task3.py at main · YarIkkkkkkkkkk/TP-KB-222-Bohdan-Kotov (github.com)](https://github.com/YarIkkkkkkkkkk/TP-KB-222-Bohdan-Kotov/blob/main/topic_02/task3.py)

Знімок екрану з посилання на github:



Висновок

Під час виконання завдання я ознайомився з поняттям умовного переходу, дізнався про оператори if elif else match. Навчився використовувати їх на практиці.

Звіт до Теми №3

Цикли

Під час виконання практичного завдання до Теми №3 було надано варіанти рішення до наступних задач:

**Завдання 1**

Написати програму калькулятор з постійними запитами на введення нових даних та операцій. За основу взяти програму калькулятор з попередньої теми. Реалізувати механізм завершення програми після отримання відповідної команди.

Хід виконання завдання:

Пишемо функції які виконують ділення множення додавання віднімання. Пишемо цикл який повторює запити на введення даних. Додаємо функцію яка припиняє роботу коли у рядок функцій введено q.

Текст програми:

def plus(a,b):

return a+b

def minus(a,b):

return a-b

def mnoj(a,b):

return a\*b

def dil(a,b):

if b==0:

return 'Ділення на нуль'

else:

return a/b

while True:

a=int(input('Введіть перше число '))

b=int(input('Введіть друге число '))

f=input('Введіть функцію ')

match f:

case 'q': break

case '+':print(plus(a,b))

case '-': print(minus(a,b))

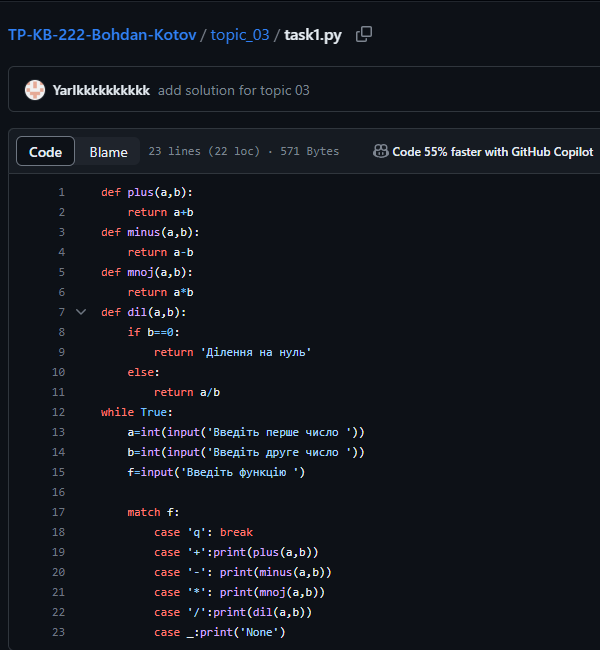
case '\*': print(mnoj(a,b))

case '/':print(dil(a,b))

case \_:print('None')

Посилання на github: [TP-KB-222-Bohdan-Kotov/topic\_03/task1.py at main · YarIkkkkkkkkkk/TP-KB-222-Bohdan-Kotov (github.com)](https://github.com/YarIkkkkkkkkkk/TP-KB-222-Bohdan-Kotov/blob/main/topic_03/task1.py)

Знімок екрану з посилання на github:



**Завдання 2**

Написати програму тестування функцій списків таких як: extend(), append(), insert(id, val), remove(val), clear(), sort(), reverse(), copy().

Хід виконання завдання:

Створюємо два списки та тестуємо функції. Функція extend() додає вміст одного списку до іншого. Функція append() додає значення в кінець списку. Функція insert(id, val) вставляє значення в певну позицію в списку. Функція remove(val) видаляє перше знайдене значення зі списку. Функція clear() очищує список. Функція sort() сортує список. Функція reverse() розвертає список. Функція copy() копіює список.

Текст програми:

sp=[5,15,16,2023]

sp1=[1,1,1,1,5]

sp.extend(sp1)

print(sp)

sp.append(666)

print(sp)

sp.insert(0, 2023)

print(sp)

sp.remove(2023)

print(sp)

sp1.clear()

print(sp1)

sp.sort()

print(sp)

sp.reverse()

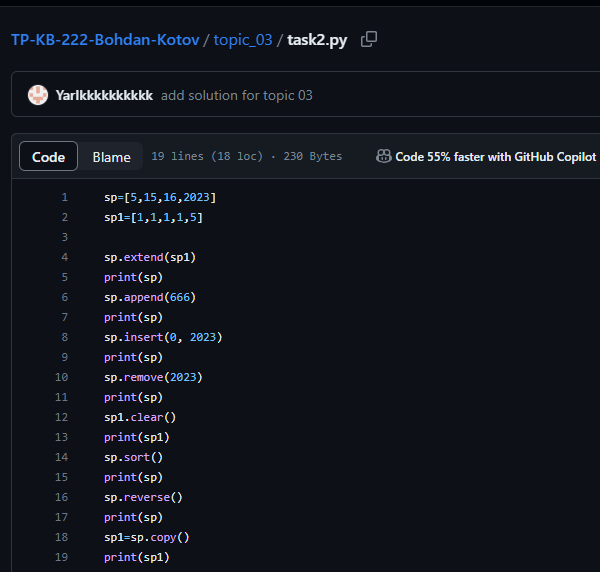
print(sp)

sp1=sp.copy()

print(sp1)

Посилання на github: [TP-KB-222-Bohdan-Kotov/topic\_03/task2.py at main · YarIkkkkkkkkkk/TP-KB-222-Bohdan-Kotov (github.com)](https://github.com/YarIkkkkkkkkkk/TP-KB-222-Bohdan-Kotov/blob/main/topic_03/task2.py)

Знімок екрану з посилання на github:



**Завдання 3**

Написати програму тестування функцій словників таких як: update(), del(), clear(), keys(), values(), items()

Хід виконання завдання:

Сворюємо два словники та тестуємо функції. Функція update() вставляє елемент у словник. Функція del() видаляє значення зі словника. Функція clear() очищує словник. Функція keys() повертає значення всіх ключів. Функція values() повертає всі значення. Функція items() повертає пари ключів та значень.

Текст програми:

car = {

"brand": "Ford",

"model": "Mustang",

"year": 1964

}

car1={

"brand": "Ford",

"model": "Mustang",

"year": 1964

}

car.update({"color": "White"})

print(car)

del car["color"]

print(car)

car1.clear()

print(car1)

x=car.keys()

print(x)

x=car.values()

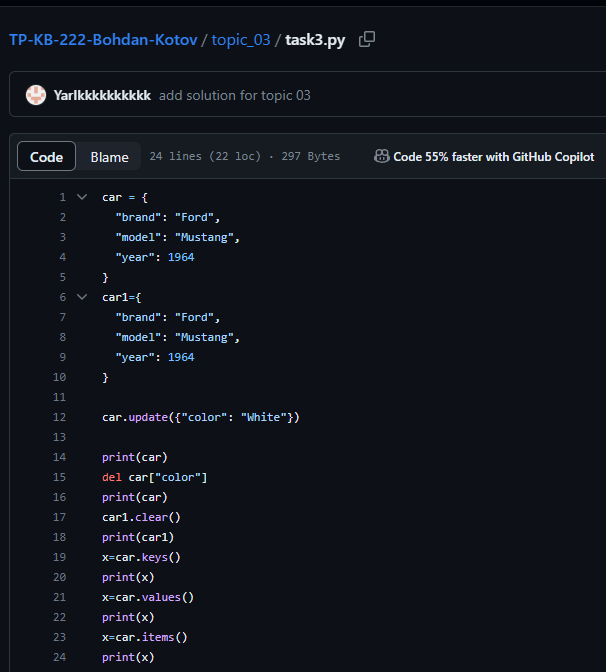
print(x)

x=car.items()

print(x)

Посилання на github: [TP-KB-222-Bohdan-Kotov/topic\_03/task3.py at main · YarIkkkkkkkkkk/TP-KB-222-Bohdan-Kotov (github.com)](https://github.com/YarIkkkkkkkkkk/TP-KB-222-Bohdan-Kotov/blob/main/topic_03/task3.py)

Знімок екрану з посилання на github:



**Завдання 4**

Маючи відсортований список, написати функцію пошуку позиції для вставки нового елементу в список.

Хід виконання завдання:

Сворюємо два список. Створюємо цикл. Додаємо функцію виходу. Перевіряємо чи число яке введено з клавіатури більше за останнє. Якщо не більше то знаходимо позицію в списку та вставляємо його туди. Вивиодимо список.

Текст програми:

sp=[1,2,5,10]

while True:

a=input('Введіть число ')

if a=='q':

break

for i in sp:

if int(a)>sp[len(sp)-1]:

sp.append(int(a))

elif int(a)<=i:

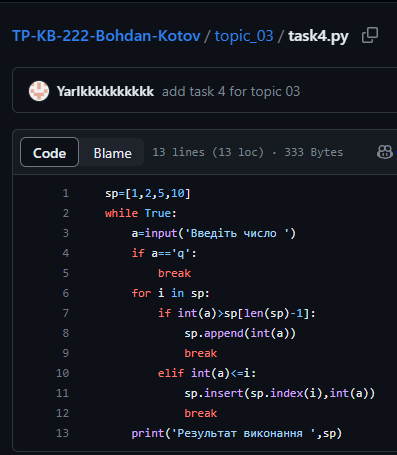
sp.insert(sp.index(i),int(a))

break

print('Результат виконання ',sp)

Посилання на github: [TP-KB-222-Bohdan-Kotov/topic\_03/task4.py at main · YarIkkkkkkkkkk/TP-KB-222-Bohdan-Kotov (github.com)](https://github.com/YarIkkkkkkkkkk/TP-KB-222-Bohdan-Kotov/blob/main/topic_03/task4.py)

Знімок екрану з посилання на github:



Висновок

Під час виконання завдання я ознайомився з поняттям циклу, дізнався про оператори while, for. Навчився використовувати їх на практиці.

Звіт до Теми №4

Виняткові ситуації

Під час виконання практичного завдання до Теми №4 було надано варіанти рішення до наступних задач:

**Завдання**

1) Розширити програму калькулятор функцією запитів даних для виконання операцій від користувача, що обробляє виняткові ситуації.

2) Розширити функцію ділення обробкою виняткової ситуації ділення на нуль

3) Ознайомитись зі списком виняткових ситуацій за посиланням

Хід виконання завдання:

Додаємо в початок коду функцію яка запитує два числа та перевіряє ці числа на виняткові ситуації. Пишемо функції які виконують множення додавання віднімання. Додаємо функцію ділення яка перевіряє виняткову ситуацію ділення на нуль, якщо це не виняткова ситуація то виконуємо ділення. Пишемо цикл який повторює запити на введення даних. Додаємо функцію яка припиняє роботу коли у рядок функцій введено q.

Текст програми:

def get\_int\_value():

while True:

try:

a = int(input("Введіть перше число: "))

b = int(input("Введіть друге число: "))

except ValueError:

print("Число не integer")

else:

break

return a,b

def plus(a,b):

return a+b

def minus(a,b):

return a-b

def mnoj(a,b):

return a\*b

def dil(a,b):

try:

dill=a/b

except ZeroDivisionError:

dill="Ділення на нуль"

else:

dill= a/b

return dill

while True:

a,b=get\_int\_value()

f=input('Введіть функцію ')

match f:

case 'q': break

case '+':print(plus(a,b))

case '-': print(minus(a,b))

case '\*': print(mnoj(a,b))

case '/':print(dil(a,b))

case \_:print('None')

Посилання на github: [TP-KB-222-Bohdan-Kotov/topic\_04/task\_1.py at main · YarIkkkkkkkkkk/TP-KB-222-Bohdan-Kotov (github.com)](https://github.com/YarIkkkkkkkkkk/TP-KB-222-Bohdan-Kotov/blob/main/topic_04/task_1.py)

Знімок екрану з посилання на github:



Висновок

Під час виконання завдання я ознайомився з поняттям виняткових ситуацій, дізнався про оператори try, except. Навчився використовувати їх на практиці.

Звіт до Теми №5

Бібліотеки

Під час виконання практичного завдання до Теми №5 було надано варіанти рішення до наступних задач:

**Завдання**

1) Гра з комп’ютером: камінь, ножиці, папір. Програма виконує запит від користувача на введення одного із значень ["rock", "scissor", "paper"]. Наступним кроком, використовуючи модуль random, програма у випадковому порядку вибирає одне із значень ["rock", "scissor", "paper"]. В залежності від умови, що камінь перемагає ножиці, ножиці перемагають папір, а папір перемагає камінь визначити переможця.

2) Програма конвертування іноземної валюти в українську гривню. Для отримання актуальних курсів валют необхідно використовувати API НБУ та модуль, що надає можливість виконувати запити до сторонніх сервісів requests. Достатня умова роботи – можливість конвертації для трьох іноземних валют EUR, USD, PLN. Користувачу надається можливість введення кількості та типу валюти, результат роботи програми – конвертоване значення в українських гривнях.

3) Використання модулів для програми калькулятор. Функції додавання, віднімання, множення та ділення перенести в файл functions.py. Функції запиту на введення даних для операцій та самих операцій перемістити в файл operations.py. Програму калькулятор реалізувати в файлі calc.py, до якого підключають файл functions.py та operations.py.

Хід виконання завдання 1:

Додаємо в початок коду функцію яка запитує наш вибір або q. Імпортуємо модуль random. Створюємо список зі значеннями. Створюємо цикл де запитуємо значення у гравця та викликаємо значення у комп’ютера. Перевіряємо хто виграв та виводимо значення яке обрав комп’ютер.

Текст програми:

def get\_znach():

while True:

player=input('Введіть ваш вибір ')

if player not in sp and player!='q':

print('Будь ласка введіть знову')

else:

return player

break

import random

sp=["rock", "scissor", "paper"]

while True:

player=get\_znach()

comp = random.choice(sp)

if player=='q':break

if player==comp:print('Draw computer chose ',comp)

elif player=='rock' and comp=='scissor':print('You WIN computer chose ',comp)

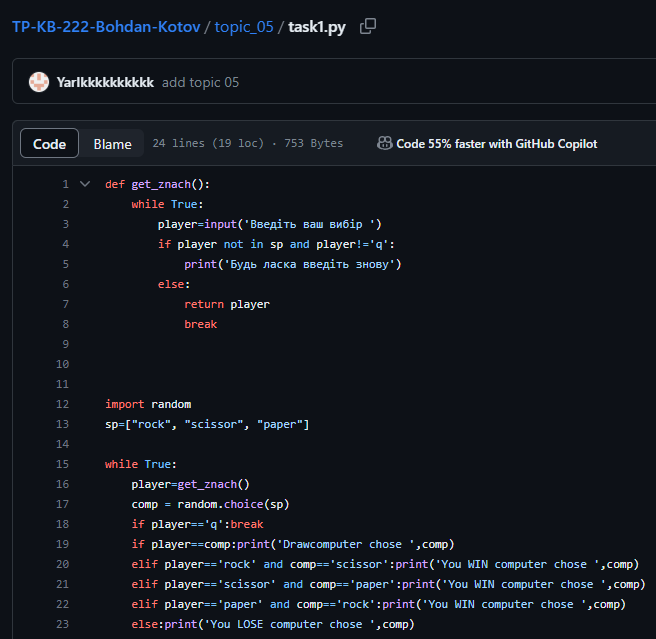
elif player=='scissor' and comp=='paper':print('You WIN computer chose ',comp)

elif player=='paper' and comp=='rock':print('You WIN computer chose ',comp)

else:print('You LOSE computer chose ',comp)

Посилання на github: [TP-KB-222-Bohdan-Kotov/topic\_05/task1.py at main · YarIkkkkkkkkkk/TP-KB-222-Bohdan-Kotov (github.com)](https://github.com/YarIkkkkkkkkkk/TP-KB-222-Bohdan-Kotov/blob/main/topic_05/task1.py)

Знімок екрану з посилання на github:



Хід виконання завдання 2:

Імпортуємо модуль requests. Виконуємо запит на сайт. Далі з усіх значень які дає нам запит ми створюємо словник у якому беремо назву валюти та курсу. Потім ми запитуємо суму та валюту. Перевіряємо чи введена валюта існує в нашому словнику. Виводимо на екран конвертовану суму.

Текст програми:

import requests

url = "<https://bank.gov.ua/NBUStatService/v1/statdirectory/exchangenew?json>"

response = requests.get(url)

data = response.json()

znach = {item["cc"]: item["rate"] for item in data}

print(znach)

while True:

suma = float(input("Введіть суму: "))

val= input("Введіть валюту для конвертації : ").upper()

if val not in znach:

print("Валюту не знайдено")

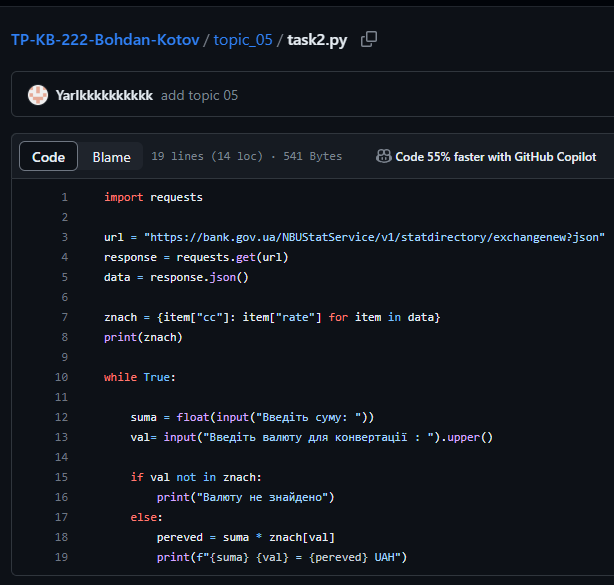
else:

pereved = suma \* znach[val]

print(f"{suma} {val} = {pereved} UAH")

Посилання на github: [TP-KB-222-Bohdan-Kotov/topic\_05/task2.py at main · YarIkkkkkkkkkk/TP-KB-222-Bohdan-Kotov (github.com)](https://github.com/YarIkkkkkkkkkk/TP-KB-222-Bohdan-Kotov/blob/main/topic_05/task2.py)

Знімок екрану з посилання на github:



Хід виконання завдання 3:

Створюємо три файли. У файлі functions прописуємо функції додавання, віднімання, ділення, множення. У файлі operations прописуємо функції запиту числа та запиту функції. У файлі calc імпортуємо все з файлів functions та operations. Викликаємо функції запиту чисел та функції. Перевіряємо яку саме функцію ввів користувач та викликаємо її.

Текст програми:

from functions import \*

from operations import \*

while True:

a,b=get\_int\_value()

f=get\_func()

match f:

case 'q': break

case '+':print(plus(a,b))

case '-': print(minus(a,b))

case '\*': print(mnoj(a,b))

case '/':print(dil(a,b))

case \_:print('None')

def get\_int\_value():

while True:

try:

a = int(input("Введіть перше число: "))

b = int(input("Введіть друге число: "))

except ValueError:

print("Число не integer")

else:

break

return a,b

def get\_func():

f=input('Введіть функцію ')

return f

def plus(a,b):

return a+b

def minus(a,b):

return a-b

def mnoj(a,b):

return a\*b

def dil(a,b):

try:

dill=a/b

except ZeroDivisionError:

dill="Ділення на нуль"

else:

dill= a/b

return dill

Посилання на github: [TP-KB-222-Bohdan-Kotov/topic\_05/calc.py at main · YarIkkkkkkkkkk/TP-KB-222-Bohdan-Kotov (github.com)](https://github.com/YarIkkkkkkkkkk/TP-KB-222-Bohdan-Kotov/blob/main/topic_05/calc.py)

Посилання на github: [TP-KB-222-Bohdan-Kotov/topic\_05/functions.py at main · YarIkkkkkkkkkk/TP-KB-222-Bohdan-Kotov (github.com)](https://github.com/YarIkkkkkkkkkk/TP-KB-222-Bohdan-Kotov/blob/main/topic_05/functions.py)

Посилання на github: [TP-KB-222-Bohdan-Kotov/topic\_05/operations.py at main · YarIkkkkkkkkkk/TP-KB-222-Bohdan-Kotov (github.com)](https://github.com/YarIkkkkkkkkkk/TP-KB-222-Bohdan-Kotov/blob/main/topic_05/operations.py)

Знімоки екрану з посилання на github:

Висновок

Під час виконання завдання я ознайомився з поняттям бібліотек, дізнався про бібліотеки random, requests. Навчився використовувати їх на практиці.

Звіт до Теми №6

Робота з файлами

Під час виконання практичного завдання до Теми №6 було надано варіанти рішення до наступних задач:

**Завдання**

1) Розробити механізм логування всіх дій, що виконує програма. Забезпечити зберігання інформації про введені данні, виконану операцію та результат виконання операції над даними.

2) Маючи не відсортований список, елементами якого є словники з двома параметрами (ім’я та оцінка) виконати сортування списку, використовуючи стандартну функцію sorted(). Другим параметром для функції sorted() має бути lambda функція, що повертає ім’я або оцінку із елемента словника.

Хід виконання завдання 1:

Створюємо три файли. У файлі functions прописуємо функції додавання, віднімання, ділення, множення. У файлі operations прописуємо функції запиту числа, запиту функції та логування. У файлі calc імпортуємо все з файлів functions та operations. Викликаємо функцію запиту чисел та функції. Перевіряємо яку саме функцію ввів користувач та викликаємо її.

Текст програми:

from functions import \*

from operations import \*

while True:

a,b,f=main()

match f:

case 'q': break

case '+':

print(plus(a,b))

case '-':

print(minus(a,b))

case '\*':

print(mnoj(a,b))

case '/':

print(dil(a,b))

case \_:

print('None')

from functions import \*

def main():

while True:

a = int(input("Перше число: "))

b = int(input("Друге число: "))

f = input("Операція (+, -, \*, /):")

if f == "+":

result = plus(a, b)

log\_message = f"numbers: a = {a}, b = {b}; operation = {f}; result = {result}\n"

log(log\_message)

elif f == "-":

result = minus(a, b)

log\_message = f"numbers: a = {a}, b = {b}; operation = {f}; result = {result}\n"

log(log\_message)

elif f == "\*":

result = mnoj(a, b)

log\_message = f"numbers: a = {a}, b = {b}; operation = {f}; result = {result}\n"

log(log\_message)

elif f == "/":

result=(dil(a,b))

log\_message = f"numbers: a = {a}, b = {b}; operation = {f}; result = {result}\n"

log(log\_message)

else:

result = "Невідома операція"

log\_message = f"numbers: a = {a}, b = {b}; operation = {f}; result = {result}\n"

log(log\_message)

return a,b,f

def log(log\_message):

with open("log.txt", "a") as file:

file.write(log\_message)

def plus(a,b):

return a+b

def minus(a,b):

return a-b

def mnoj(a,b):

return a\*b

def dil(a,b):

try:

dill=a/b

except ZeroDivisionError:

dill="Ділення на нуль"

else:

dill= a/b

return dill

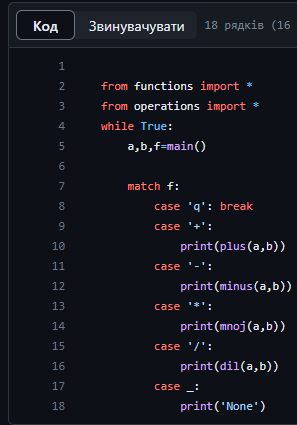
Посилання на github: [TP-KB-222-Bohdan-Kotov/topic\_06/calc.py at main · YarIkkkkkkkkkk/TP-KB-222-Bohdan-Kotov (github.com)](https://github.com/YarIkkkkkkkkkk/TP-KB-222-Bohdan-Kotov/blob/main/topic_06/calc.py)

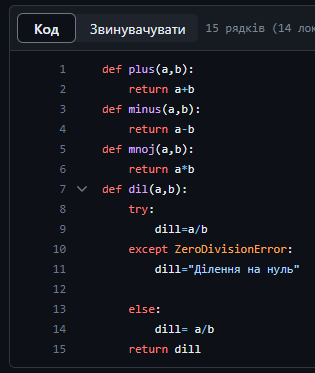
Посилання на github: [TP-KB-222-Bohdan-Kotov/topic\_06/functions.py at main · YarIkkkkkkkkkk/TP-KB-222-Bohdan-Kotov (github.com)](https://github.com/YarIkkkkkkkkkk/TP-KB-222-Bohdan-Kotov/blob/main/topic_06/functions.py)

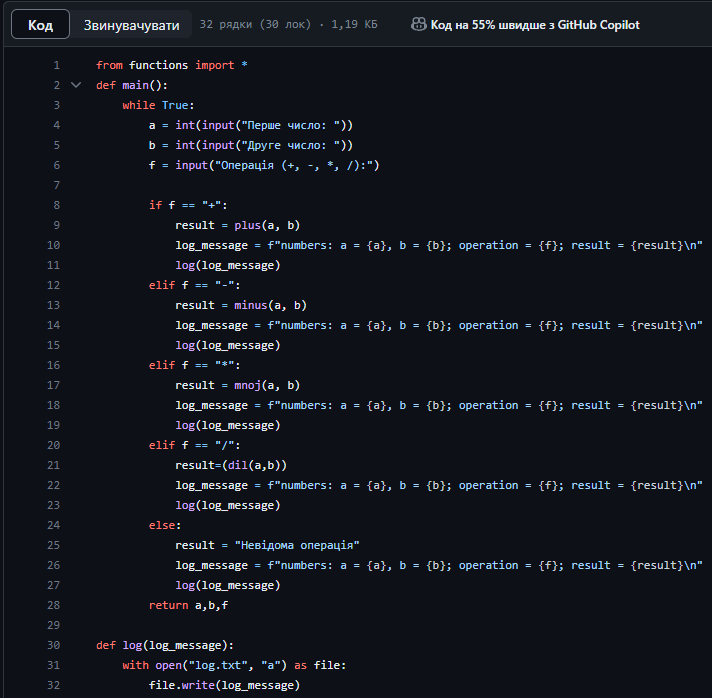
Посилання на github: [TP-KB-222-Bohdan-Kotov/topic\_06/operations.py at main · YarIkkkkkkkkkk/TP-KB-222-Bohdan-Kotov (github.com)](https://github.com/YarIkkkkkkkkkk/TP-KB-222-Bohdan-Kotov/blob/main/topic_06/operations.py)

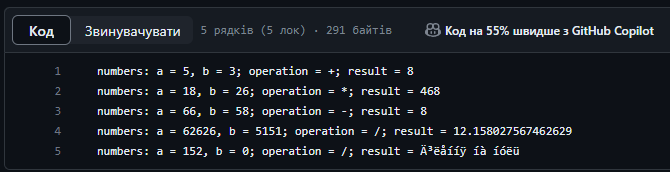
Посилання на github: [TP-KB-222-Bohdan-Kotov/topic\_06/log.txt at main · YarIkkkkkkkkkk/TP-KB-222-Bohdan-Kotov (github.com)](https://github.com/YarIkkkkkkkkkk/TP-KB-222-Bohdan-Kotov/blob/main/topic_06/log.txt)

Знімки екрану з посилання на github:









Хід виконання завдання 2:

Створюємо список словників який складається з імен та оцінок. Запитуємо ключ сортування. Прописуємо функцію сортування в якій перевіряємо за яким ключем сортувати та за допомогою sorted та lambda сортуємо список. Виводимо на екран значення імен та оцінок відсортованого списку.

Текст програми:

# Список словників із прикладами імен та оцінок

sp = [{'name': 'Anna', 'grade': 40},

{'name': 'Peter', 'grade': 35},

{'name': 'Michael', 'grade': 88},

{'name': 'Ivan', 'grade': 50},

{'name': 'Nadja', 'grade': 30},

{'name': 'Vitalii', 'grade': 55},

{'name': 'Alexander', 'grade': 83},

{'name': 'Boris', 'grade': 92},

{'name': 'Sergey', 'grade': 30},

{'name': 'Eugene', 'grade': 75},

{'name': 'Bohdan', 'grade': 92}]

parameter = input("Виберіть ключ для сортування : ")

def sort\_data(sp, parameter):

if parameter == 'name':

return sorted(sp, key=lambda x: x['name'])

elif parameter == 'grade':

return sorted(sp, key=lambda x: x['grade'])

else:

print("Неправильний ключ сортування.")

return sp

# Виведення відсортованого списку

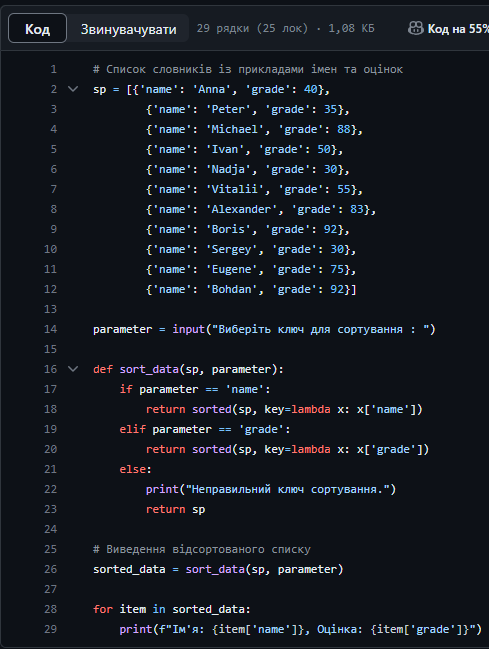
sorted\_data = sort\_data(sp, parameter)

for item in sorted\_data:

print(f"Ім'я: {item['name']}, Оцінка: {item['grade']}")

Посилання на github: [TP-KB-222-Bohdan-Kotov/topic\_06/task\_2.py at main · YarIkkkkkkkkkk/TP-KB-222-Bohdan-Kotov (github.com)](https://github.com/YarIkkkkkkkkkk/TP-KB-222-Bohdan-Kotov/blob/main/topic_06/task_2.py)

Знімок екрану з посилання на github:



Висновок

Під час виконання завдання я ознайомився з використанням файлів, дізнався про функції роботи з файлами. Навчився використовувати їх на практиці.

Звіт до Теми №7

Об’єктно-орієнтоване програмування

Під час виконання практичного завдання до Теми №7 було надано варіанти рішення до наступних задач:

**Завдання**

1)Ознайомитись з документацією що описує можливості використання класів у мові Python https://docs.python.org/3/tutorial/classes.html

2)Ознайомитись з існуючими за замовченням методами класу по типу \_\_init\_\_(self) \_\_str\_\_(self)\_\_ та надати приклади використання.

3)Розробити клас Student атрибутами якого э два параметра name та age. Створити список елементами якого є об'єкти класу Student. Написати цикл який виводить на екран елементи списку у відсортованому порядку. Для сортування використати стандартну функцію sorted. Функція sorted має використовувати lambda функцію для визначення ключа сортування.

4)Використовуючи принципи ООП переписати програму Калькулятор. Завдання має бути виконано використовуючи модульний підхід

Хід виконання завдання 2:

Створюємо клас Student. Цей клас містить метод \_\_init\_\_. При створенні нового об'єкта класу Student, цей метод встановлює атрибути name та mark для об'єкта. Метод \_\_str\_\_ повертає рядок, який містить ім'я та оцінку об'єкта.

Приклад програми:

class Student:

def \_\_init\_\_(self, name, mark):

self.name = name

self.mark = mark

def \_\_str\_\_(self):

return f"Person: {self.name}, mark: {self.mark}"

student = Student("Oleg", 99)

print(str(student))

Хід виконання завдання 3:

Створюємо клас Student з двома атрибутами name, age. Створюємо список з об'єктами класу Student. Запитуємо у користувача ключ за яким ми будемо сортувати список. Сортуємо список використовуючи lambda функцію. Створюємо цикл який виводить на екран елементи списку у відсортованому порядку.

Текст програми:

class Student:

def \_\_init\_\_(self, name, age):

self.name = name

self.age = age

students = [

Student("Віктор", 19),

Student("Григорій", 21),

Student("Богдан", 19),

Student("Степан", 25),

Student("Марія", 24),

Student("Ярослав", 28)

]

sort= input("Якщо бажаєте сортувати за віком введіть 'В',якщо сортувати за алфавітом введіть 'А' ")

if sort == 'А':

students = sorted(students, key=lambda x: x.name)

else:

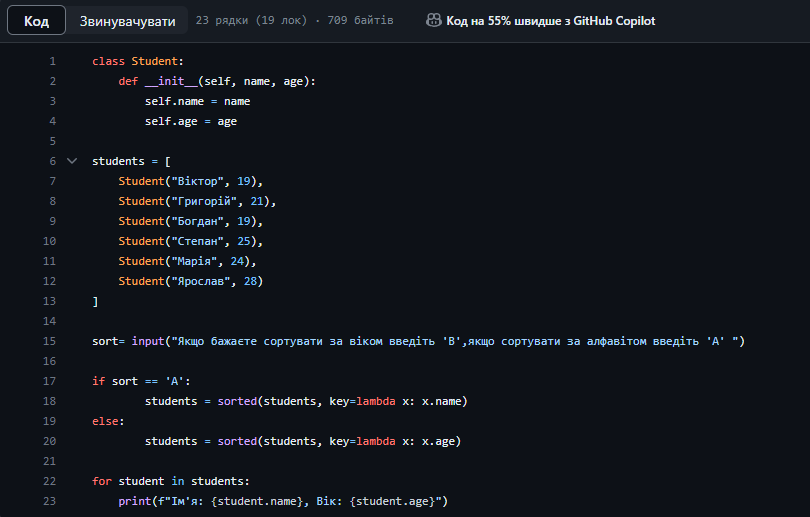
students = sorted(students, key=lambda x: x.age)

for student in students:

print(f"Ім'я: {student.name}, Вік: {student.age}")

Посилання на github: [TP-KB-222-Bohdan-Kotov/topic\_07/task\_3.py at main · YarIkkkkkkkkkk/TP-KB-222-Bohdan-Kotov (github.com)](https://github.com/YarIkkkkkkkkkk/TP-KB-222-Bohdan-Kotov/blob/main/topic_07/task_3.py)

Знімок екрану з посилання на github:



Хід виконання завдання 4:

Створюємо два файли. У файлі operations прописуємо функції запиту числа, запиту функції . У файлі calc створюємо клас Calculator, прописуємо в ньому операнди, функції додавання, віднімання, множення, ділення та логування.

Текст програми:

from operations import \*

class Calculator:

def \_\_init\_\_(self, log\_file):

self.\_operand1 = 0

self.\_operand2 = 0

self.log\_file = log\_file

@property

def operand1(self):

return self.\_operand1

@operand1.setter

def operand1(self, value):

self.\_operand1 = value

@property

def operand2(self):

return self.\_operand2

@operand2.setter

def operand2(self, value):

self.\_operand2 = value

def add(self):

result = self.operand1 + self.operand2

self.\_log("Addition", result)

return result

def subtract(self):

result = self.operand1 - self.operand2

self.\_log("Subtraction", result)

return result

def multiply(self):

result = self.operand1 \* self.operand2

self.\_log("Multiplication", result)

return result

def divide(self):

if self.operand2 == 0:

result = "Error: Division by zero"

self.\_log("Division", result)

return result

result = self.operand1 / self.operand2

self.\_log("Division", result)

return result

def \_log(self, operation, result):

with open(self.log\_file, "a") as log:

log.write(f"{operation}: operand1={self.operand1}, operand2={self.operand2}, result={result}\n")

if \_\_name\_\_ == "\_\_main\_\_":

main()

from calc import Calculator

def main():

log\_file = "topic\_07/log.txt"

calc = Calculator(log\_file)

while True:

choice = input("Введіть функцію: ")

operand = float(input("Введіть перше число: "))

calc.operand1 = operand

operand = float(input("Введіть: "))

calc.operand2 = operand

if choice == '+':

result = calc.add()

print("Result:", result)

elif choice == '-':

result = calc.subtract()

print("Result:", result)

elif choice == '\*':

result = calc.multiply()

print("Result:", result)

elif choice == '/':

result = calc.divide()

print("Result:", result)

else:

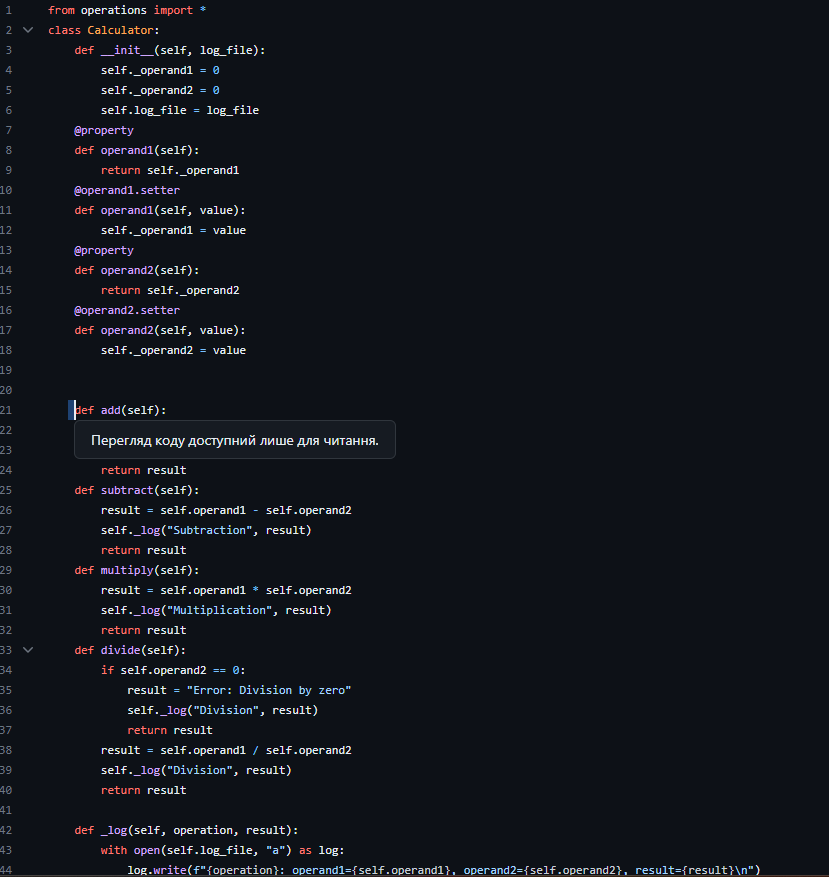
print("Invalid choice. Please try again.")

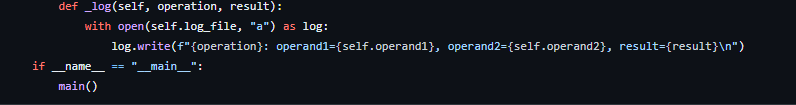
Посилання на github: [TP-KB-222-Bohdan-Kotov/topic\_07/calc.py at main · YarIkkkkkkkkkk/TP-KB-222-Bohdan-Kotov (github.com)](https://github.com/YarIkkkkkkkkkk/TP-KB-222-Bohdan-Kotov/blob/main/topic_07/calc.py)

Посилання на github: [TP-KB-222-Bohdan-Kotov/topic\_07/operations.py at main · YarIkkkkkkkkkk/TP-KB-222-Bohdan-Kotov (github.com)](https://github.com/YarIkkkkkkkkkk/TP-KB-222-Bohdan-Kotov/blob/main/topic_07/operations.py)

Посилання на github: [TP-KB-222-Bohdan-Kotov/topic\_07/log.txt at main · YarIkkkkkkkkkk/TP-KB-222-Bohdan-Kotov (github.com)](https://github.com/YarIkkkkkkkkkk/TP-KB-222-Bohdan-Kotov/blob/main/topic_07/log.txt)

Знімки екрану з посилання на github:







Висновок

Під час виконання завдання я ознайомився з поняттям ООП, дізнався про класи. Навчився використовувати їх на практиці.