**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:

1. Відкрила ВСкод
2. Прочитала завдання
3. Виконала завдання
4. Затестила
5. Закинула в звіт

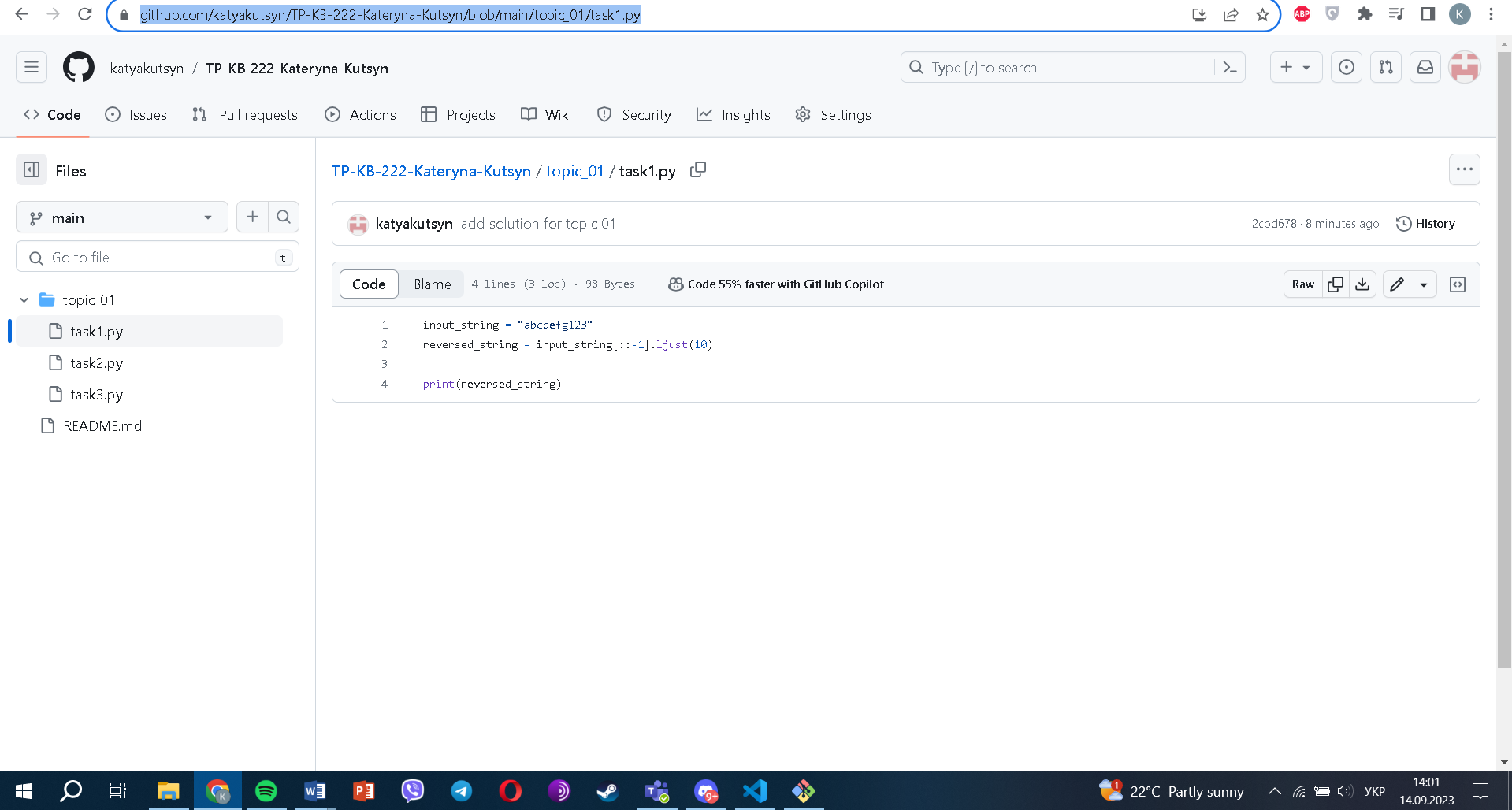
Текст програми:

|  |
| --- |
| input\_string = "abcdefg123"  reversed\_string = input\_string[::-1]  print(reversed\_string) |

Посилання на github:

<https://github.com/katyakutsyn/TP-KB-222-Kateryna-Kutsyn/blob/main/topic_01/task1.py>

Знімок екрану з посилання на github:



**Форматування рядка**

Виконайте певний тест для strip, capitalize, title, upper, lower, replace

Хід виконання завдання:

1. Відкрила ВСкод
2. Прочитала завдання
3. Виконала завдання
4. Затестила
5. Закинула в звіт

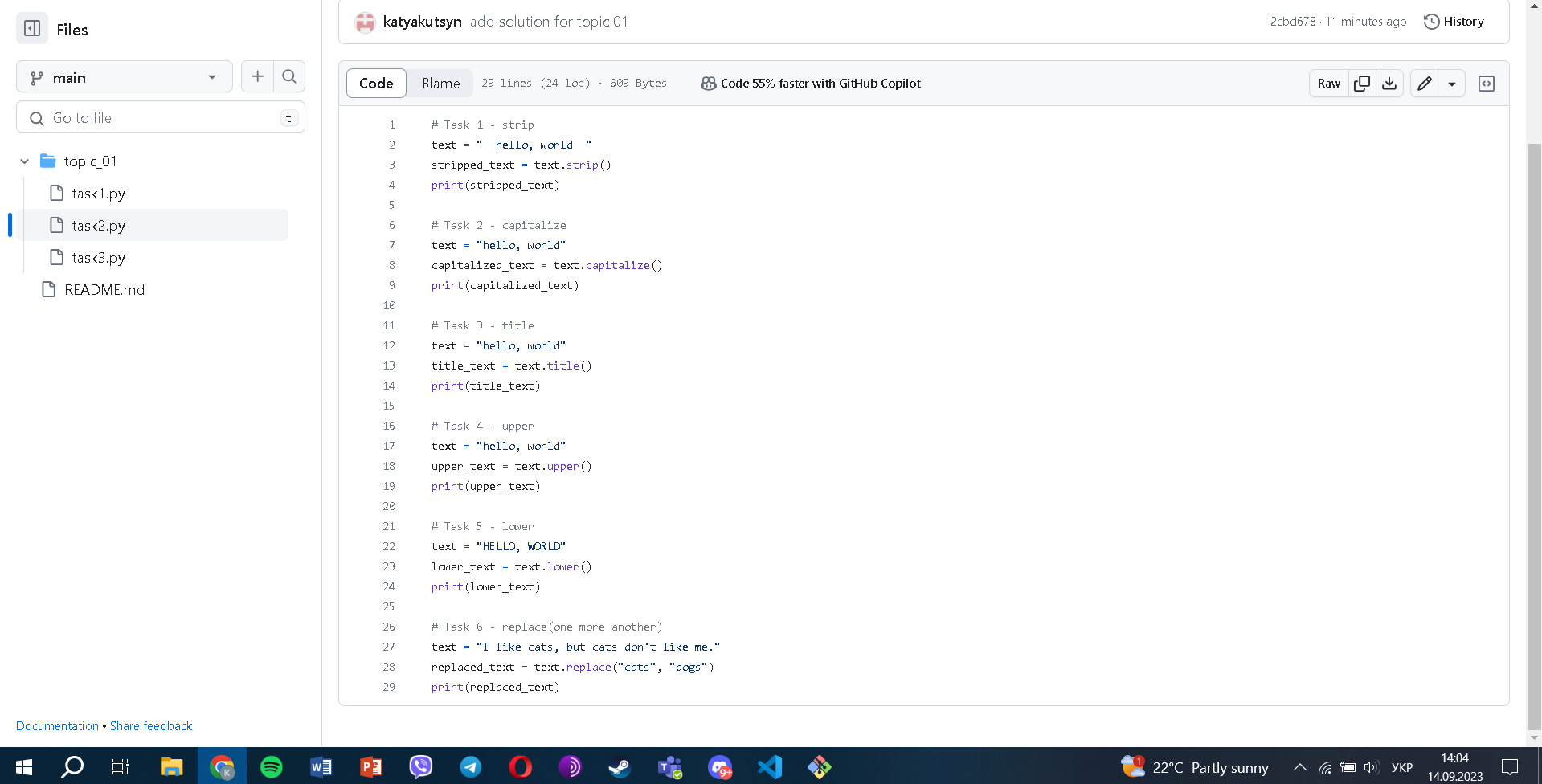
Текст програми:

|  |
| --- |
| # Task 1 - strip  text = " hello, world "  stripped\_text = text.strip()  print(stripped\_text)  # Task 2 - capitalize  text = "hello, world"  capitalized\_text = text.capitalize()  print(capitalized\_text)  # Task 3 - title  text = "hello, world"  title\_text = text.title()  print(title\_text)  # Task 4 - upper  text = "hello, world"  upper\_text = text.upper()  print(upper\_text)  # Task 5 - lower  text = "HELLO, WORLD"  lower\_text = text.lower()  print(lower\_text)  # Task 6 - replace(one more another)  text = "I like cats, but cats don't like me."  replaced\_text = text.replace("cats", "dogs")  print(replaced\_text) |

Посилання на github:

[https://github.com/katyakutsyn/TP-KB-222-Kateryna- Kutsyn/blob/main/topic\_01/task2.py](https://github.com/katyakutsyn/TP-KB-222-Kateryna-%20%20Kutsyn/blob/main/topic_01/task2.py)

Знімок екрану з посилання на github:



**Функція для знаходження дискримінанту**

Виконайте і напишіть функцію для знаходження дискримінанту

Хід виконання завдання:

1. Відкрила ВСкод
2. Прочитала завдання
3. Виконала завдання
4. Затестила
5. Закинула в звіт

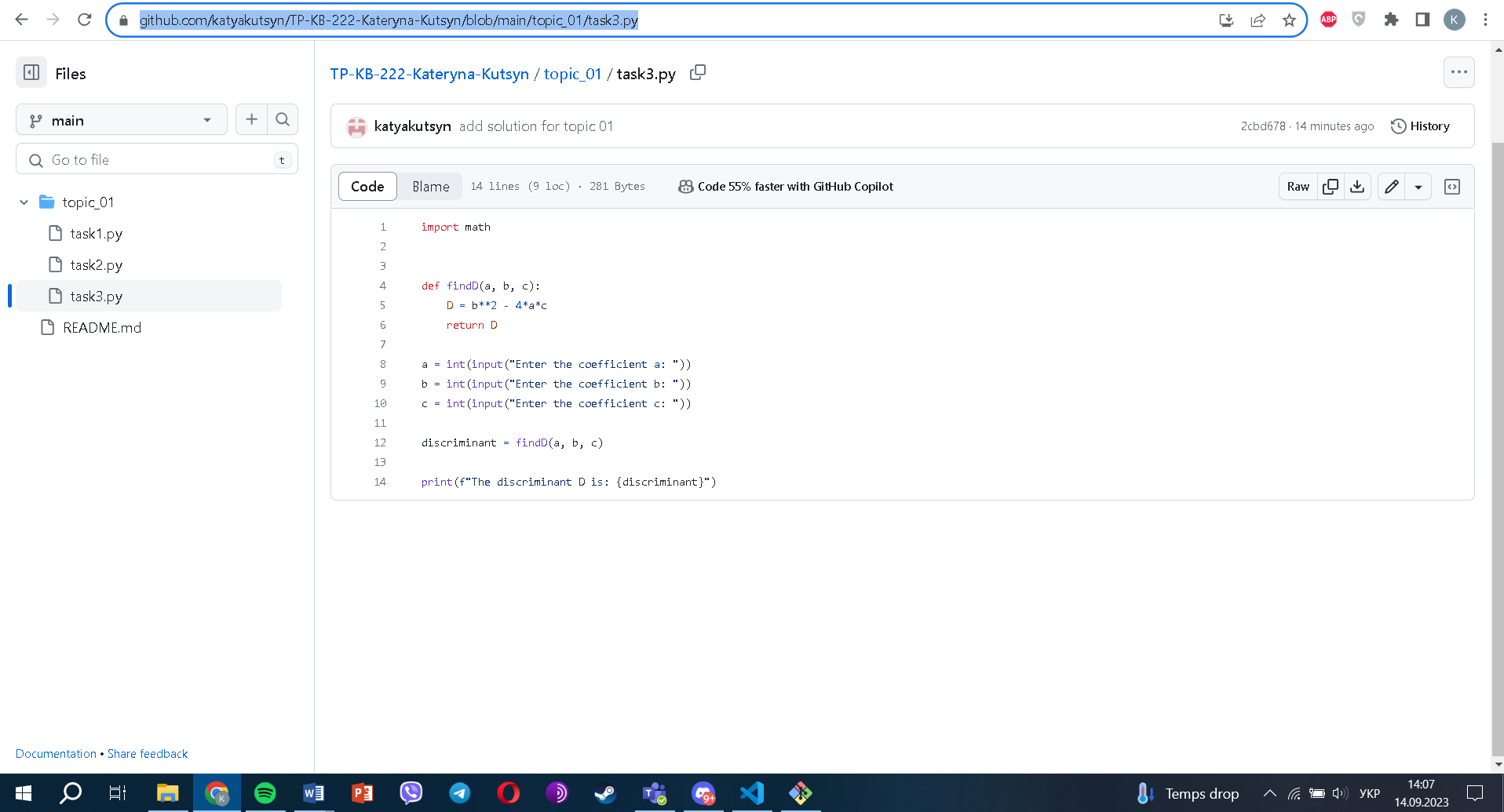
Текст програми:

|  |
| --- |
| import math  def findD(a, b, c):  D = b\*\*2 - 4\*a\*c  return D  a = int(input("Enter the coefficient a: "))  b = int(input("Enter the coefficient b: "))  c = int(input("Enter the coefficient c: "))  discriminant = findD(a, b, c)  print(f"The discriminant D is: {discriminant}") |

Посилання на github:

<https://github.com/katyakutsyn/TP-KB-222-Kateryna-Kutsyn/blob/main/topic_01/task3.py>

Знімок екрану з посилання на github:



Звіт до Теми №2

Функції та змінні

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

**Знаходження Дискримінанту**

rewrite function for quadratic equation with using conditional **for D**

D < 0

D == 0

D > 0

Хід виконання завдання:

1. Відкрила ВСкод
2. Прочитала завдання
3. Виконала завдання
4. Затестила
5. Закинула в звіт

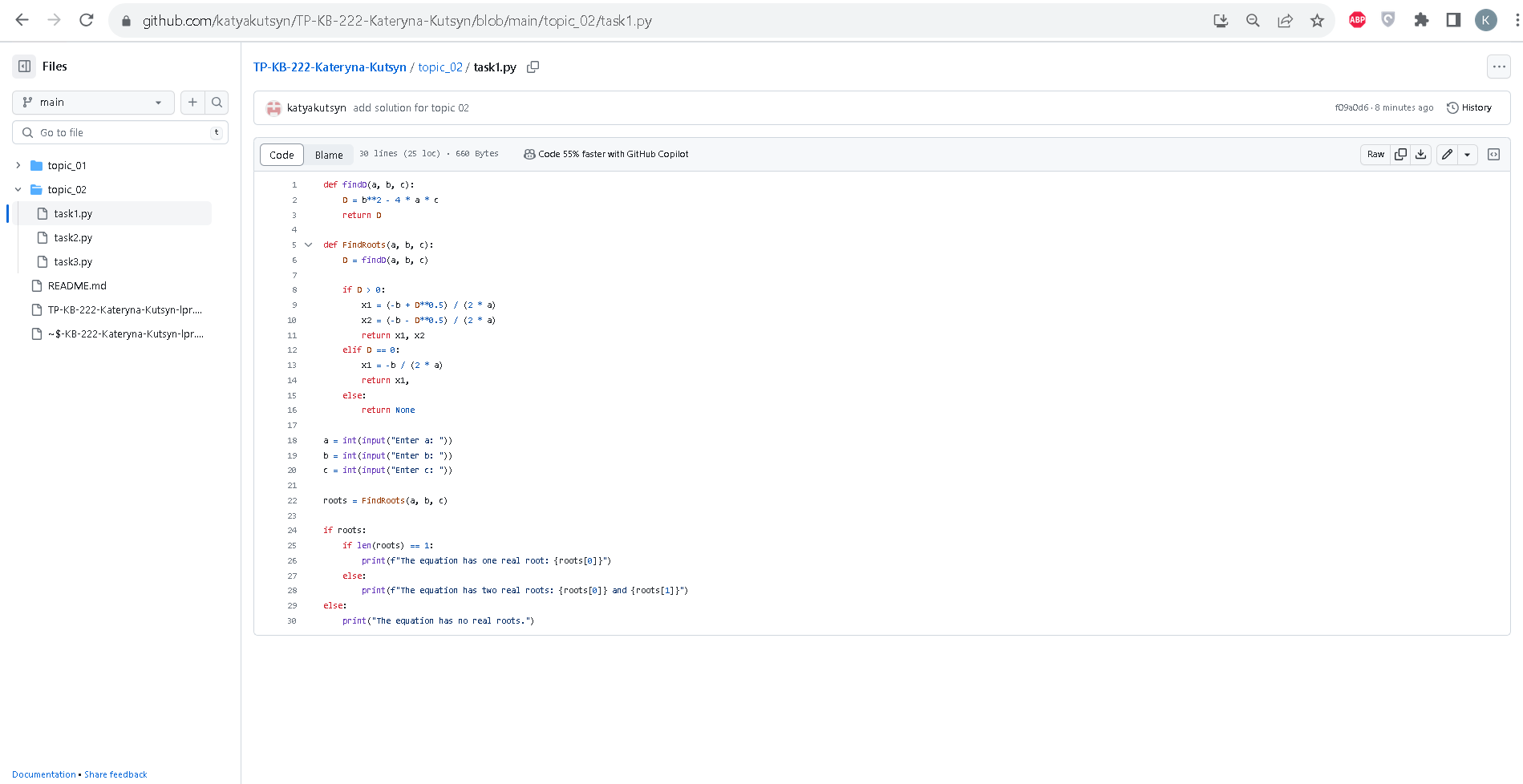
Текст програми:

|  |
| --- |
| def findD(a, b, c):  D = b\*\*2 - 4 \* a \* c  return D  def FindRoots(a, b, c):  D = findD(a, b, c)    if D > 0:  x1 = (-b + D\*\*0.5) / (2 \* a)  x2 = (-b - D\*\*0.5) / (2 \* a)  return x1, x2  elif D == 0:  x1 = -b / (2 \* a)  return x1,  else:  return None  a = int(input("Enter a: "))  b = int(input("Enter b: "))  c = int(input("Enter c: "))  roots = FindRoots(a, b, c)  if roots:  if len(roots) == 1:  print(f"The equation has one real root: {roots[0]}")  else:  print(f"The equation has two real roots: {roots[0]} and {roots[1]}")  else:  print("The equation has no real roots.") |

Посилання на github:

<https://github.com/katyakutsyn/TP-KB-222-Kateryna-Kutsyn/blob/main/topic_02/task1.py>

Знімок екрану з посилання на github:



**Створити калькулятор з if elif**

First using if elif. There are 2 values as input parameters and operation to do + - \* /

Хід виконання завдання:

1. Відкрила ВСкод
2. Прочитала завдання
3. Виконала завдання
4. Затестила
5. Закинула в звіт

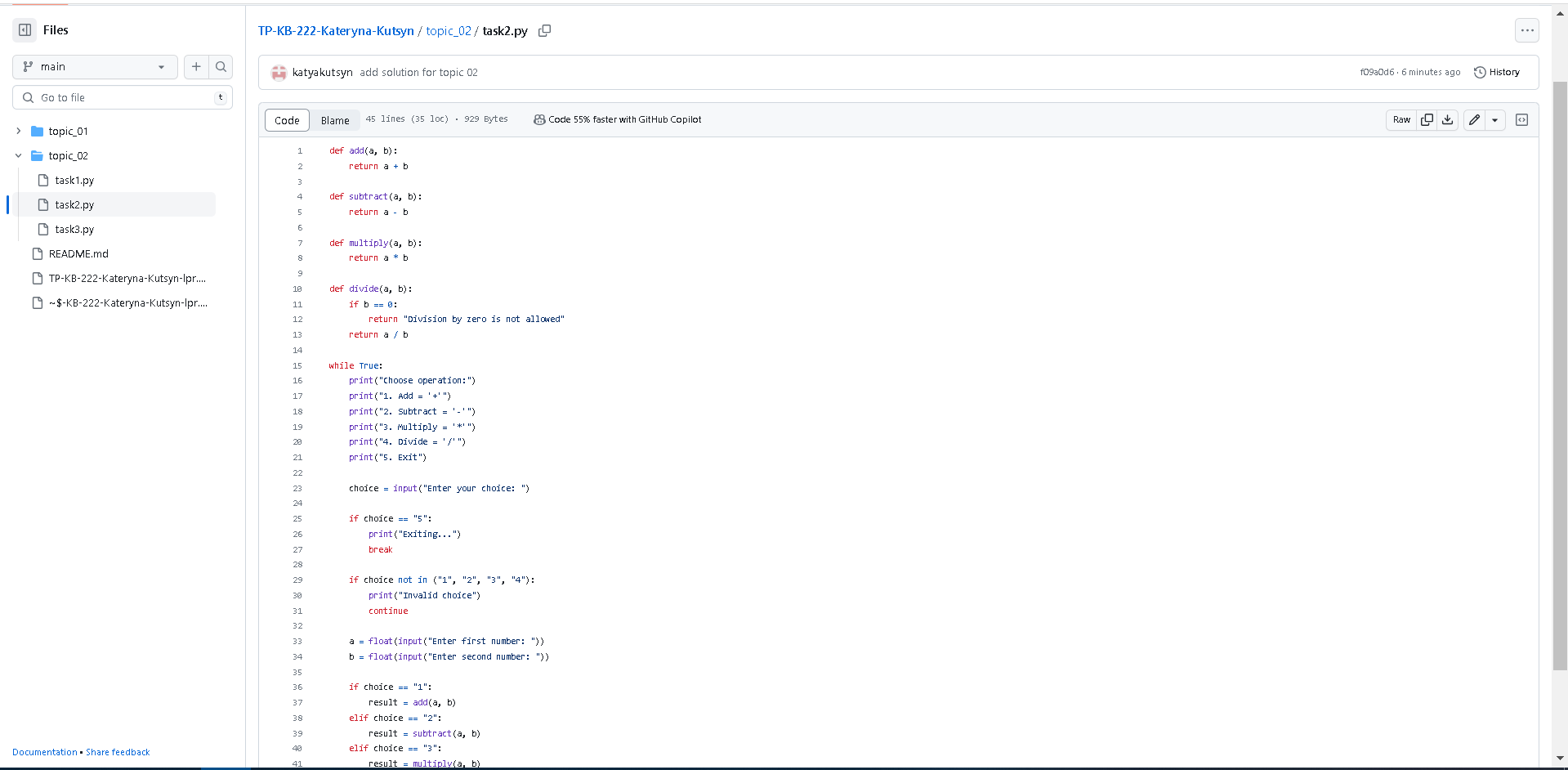
Текст програми:

|  |
| --- |
| def add(a, b):  return a + b  def subtract(a, b):  return a - b  def multiply(a, b):  return a \* b  def divide(a, b):  if b == 0:  return "Division by zero is not allowed"  return a / b  while True:  print("Choose operation:")  print("1. Add = '+'")  print("2. Subtract = '-'")  print("3. Multiply = '\*'")  print("4. Divide = '/'")  print("5. Exit")  choice = input("Enter your choice: ")  if choice == "5":  print("Exiting...")  break  if choice not in ("1", "2", "3", "4"):  print("Invalid choice")  continue  a = float(input("Enter first number: "))  b = float(input("Enter second number: "))  if choice == "1":  result = add(a, b)  elif choice == "2":  result = subtract(a, b)  elif choice == "3":  result = multiply(a, b)  elif choice == "4":  result = divide(a, b)  print("Result:", result) |

Посилання на github:

<https://github.com/katyakutsyn/TP-KB-222-Kateryna-Kutsyn/blob/main/topic_02/task2.py>

Знімок екрану з посилання на github:



**Створити калькулятор з** **match**

Second using match. There are 2 values as input parameters and operation to do + - \* /

Хід виконання завдання:

1. Відкрила ВСкод
2. Прочитала завдання
3. Виконала завдання
4. Затестила
5. Закинула в звіт

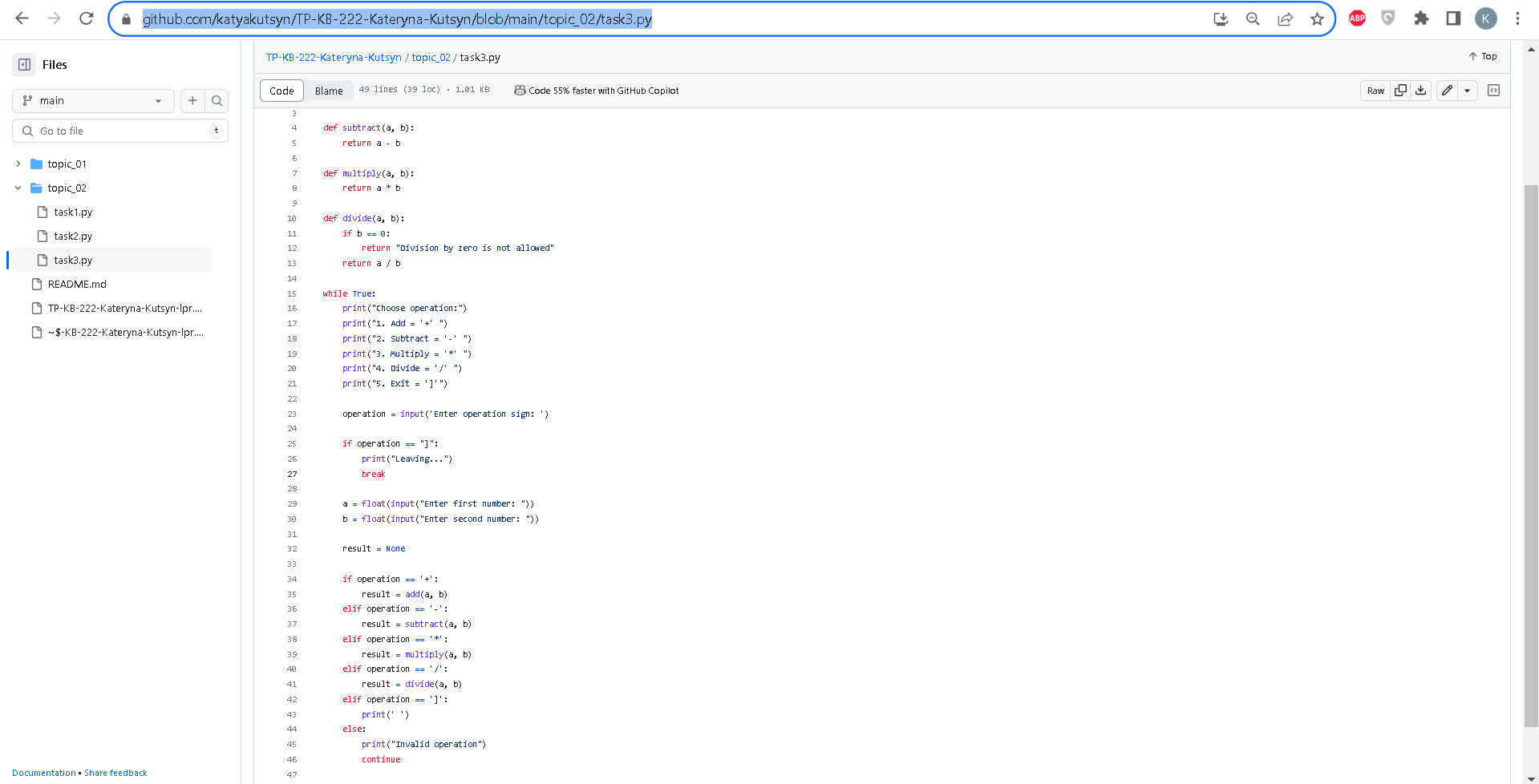
Текст програми:

|  |
| --- |
| def add(a, b):  return a + b  def subtract(a, b):  return a - b  def multiply(a, b):  return a \* b  def divide(a, b):  if b == 0:  return "Division by zero is not allowed"  return a / b  while True:  print("Choose operation:")  print("1. Add = '+' ")  print("2. Subtract = '-' ")  print("3. Multiply = '\*' ")  print("4. Divide = '/' ")  print("5. Exit = ']'")    operation = input('Enter operation sign: ')    if operation == "]":  print("Leaving...")  break    a = float(input("Enter first number: "))  b = float(input("Enter second number: "))  result = None  if operation == '+':  result = add(a, b)  elif operation == '-':  result = subtract(a, b)  elif operation == '\*':  result = multiply(a, b)  elif operation == '/':  result = divide(a, b)  elif operation == ']':  print(' ')  else:  print("Invalid operation")  continue  if result is not None:  print("Result:", result) |

Посилання на github:

<https://github.com/katyakutsyn/TP-KB-222-Kateryna-Kutsyn/blob/main/topic_02/task3.py>

Знімок екрану з посилання на github:



Звіт до Теми №3

Цикли

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

**Програма калькулятор**

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:

1. Відкрила ВСкод
2. Прочитала завдання
3. Виконала завдання
4. Затестила
5. Закинула в звіт

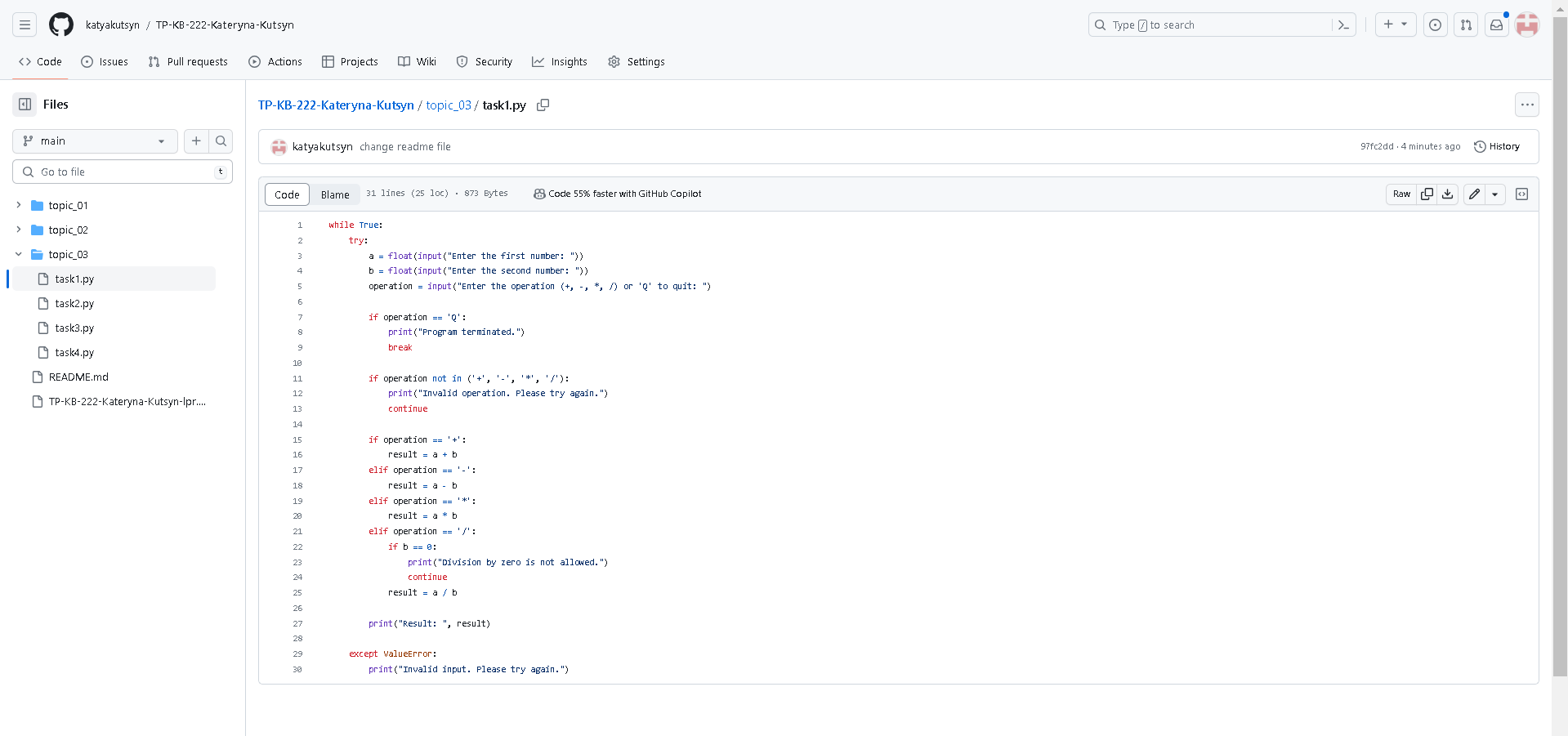
Текст програми:

|  |
| --- |
| while True:  try:  a = float(input("Enter the first number: "))  b = float(input("Enter the second number: "))  operation = input("Enter the operation (+, -, \*, /) or 'Q' to quit: ")  if operation == 'Q':  print("Program terminated.")  break  if operation not in ('+', '-', '\*', '/'):  print("Invalid operation. Please try again.")  continue  if operation == '+':  result = a + b  elif operation == '-':  result = a - b  elif operation == '\*':  result = a \* b  elif operation == '/':  if b == 0:  print("Division by zero is not allowed.")  continue  result = a / b  print("Result: ", result)  except ValueError:  print("Invalid input. Please try again.") |

Посилання на github:

<https://github.com/katyakutsyn/TP-KB-222-Kateryna-Kutsyn/blob/main/topic_03/task1.py>

Знімок екрану з посилання на github:



**Тестування функцій списків**

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

Хід виконання завдання:

1. Відкрила ВСкод
2. Прочитала завдання
3. Виконала завдання
4. Затестила
5. Закинула в звіт

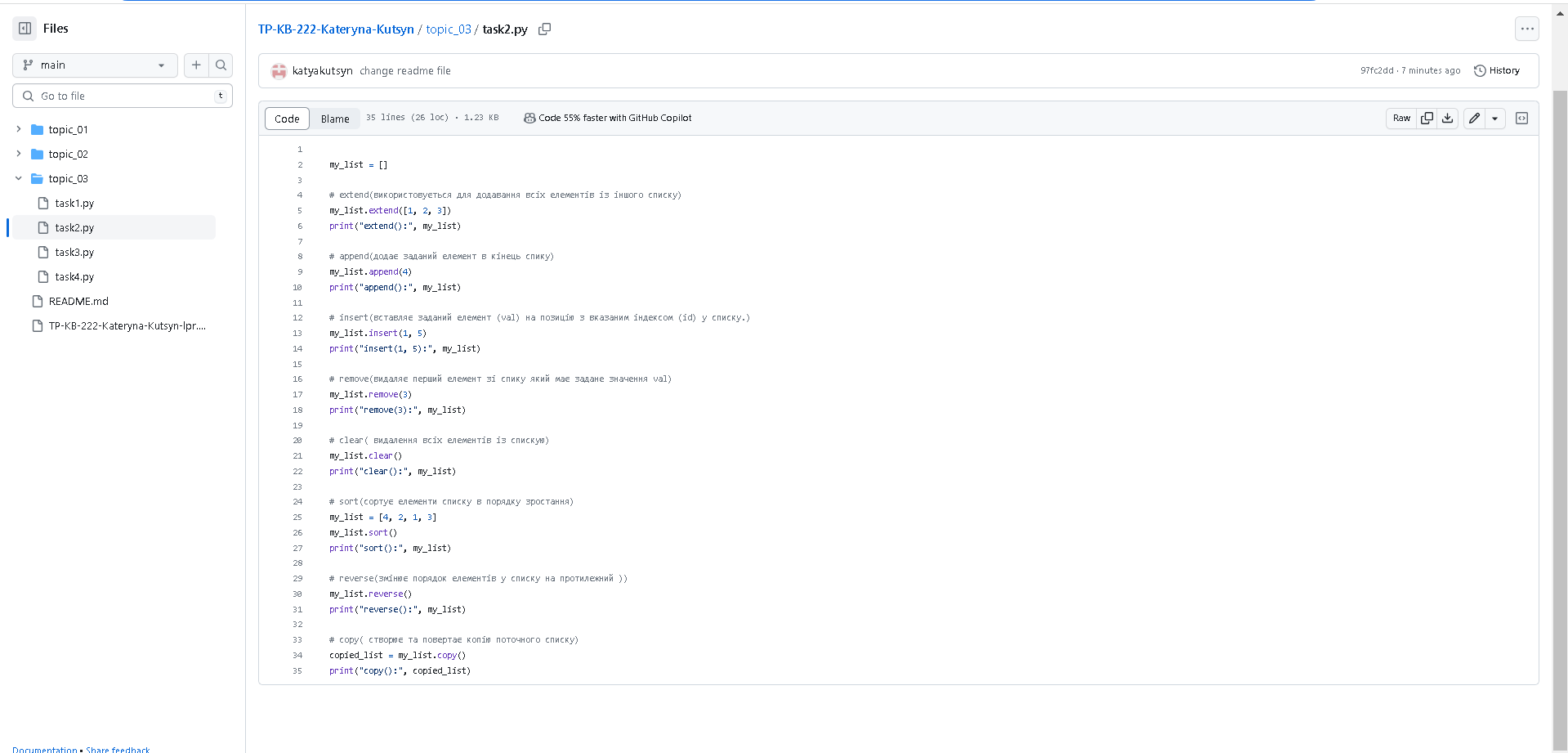
Текст програми:

|  |
| --- |
| my\_list = []  # extend(використовується для додавання всіх елементів із іншого списку)  my\_list.extend([1, 2, 3])  print("extend():", my\_list)  # append(додає заданий елемент в кінець спику)  my\_list.append(4)  print("append():", my\_list)  # insert(вставляє заданий елемент (val) на позицію з вказаним індексом (id) у списку.)  my\_list.insert(1, 5)  print("insert(1, 5):", my\_list)  # remove(видаляє перший елемент зі спику який має задане значення val)  my\_list.remove(3)  print("remove(3):", my\_list)  # clear( видалення всіх елементів із спискую)  my\_list.clear()  print("clear():", my\_list)  # sort(сортує елементи списку в порядку зростання)  my\_list = [4, 2, 1, 3]  my\_list.sort()  print("sort():", my\_list)  # reverse(змінює порядок елементів у списку на протилежний ))  my\_list.reverse()  print("reverse():", my\_list)  # copy( створює та повертає копію поточного списку)  copied\_list = my\_list.copy()  print("copy():", copied\_list) |

Посилання на github:

<https://github.com/katyakutsyn/TP-KB-222-Kateryna-Kutsyn/blob/main/topic_03/task2.py>

Знімок екрану з посилання на github:



**Тестування функцій словників**

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

Хід виконання завдання:

1. Відкрила ВСкод
2. Прочитала завдання
3. Виконала завдання
4. Затестила
5. Закинула в звіт

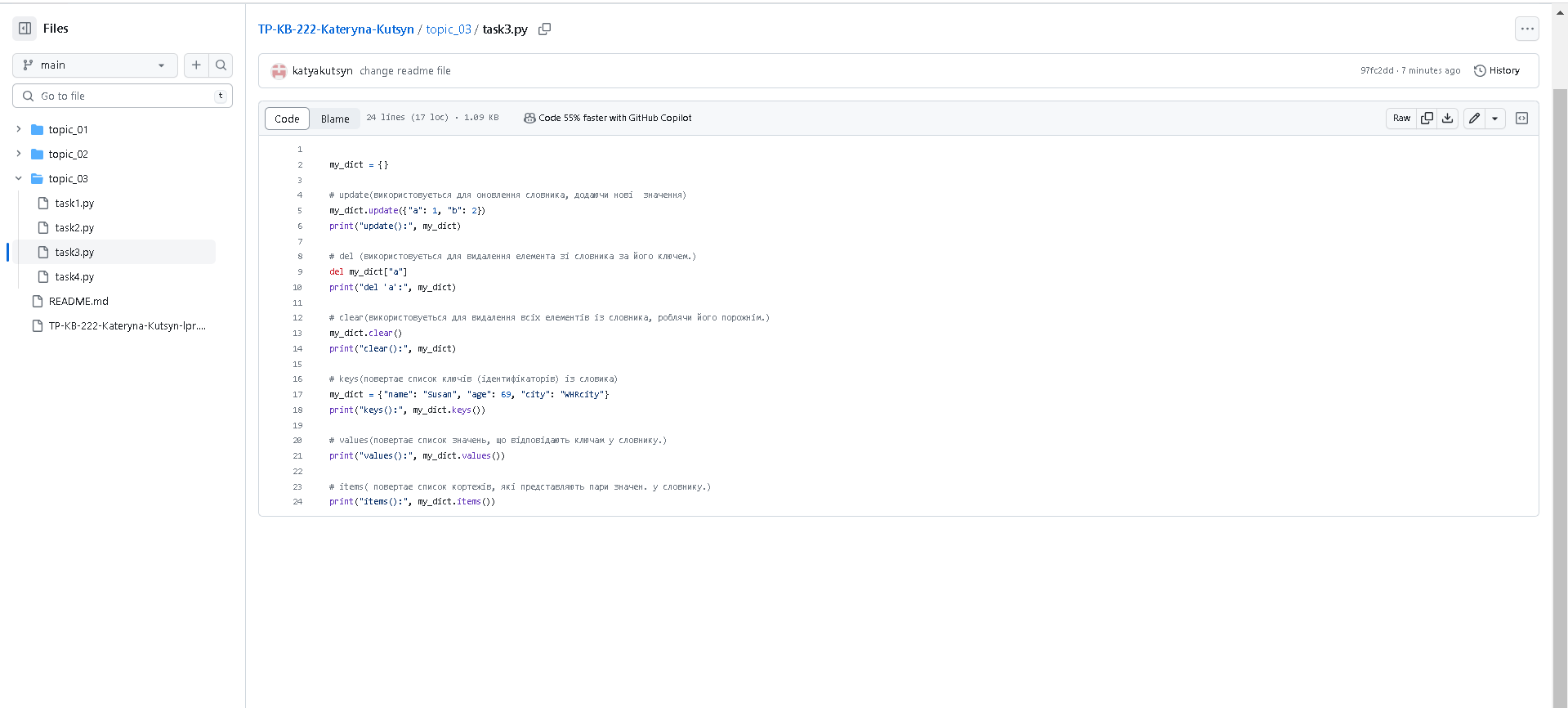
Текст програми:

|  |
| --- |
| my\_dict = {}  # update(використовується для оновлення словника, додаючи нові значення)  my\_dict.update({"a": 1, "b": 2})  print("update():", my\_dict)  # del (використовується для видалення елемента зі словника за його ключем.)  del my\_dict["a"]  print("del 'a':", my\_dict)  # clear(використовується для видалення всіх елементів із словника, роблячи його порожнім.)  my\_dict.clear()  print("clear():", my\_dict)  # keys(повертає список ключів (ідентифікаторів) із словика)  my\_dict = {"name": "Susan", "age": 69, "city": "WHRcity"}  print("keys():", my\_dict.keys())  # values(повертає список значень, що відповідають ключам у словнику.)  print("values():", my\_dict.values())  # items( повертає список кортежів, які представляють пари значен. у словнику.)  print("items():", my\_dict.items()) |

Посилання на github:

<https://github.com/katyakutsyn/TP-KB-222-Kateryna-Kutsyn/blob/main/topic_03/task3.py>

Знімок екрану з посилання на github:



**Функцію пошуку позиції**

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

Хід виконання завдання:

1. Відкрила ВСкод
2. Прочитала завдання
3. Виконала завдання
4. Затестила
5. Закинула в звіт

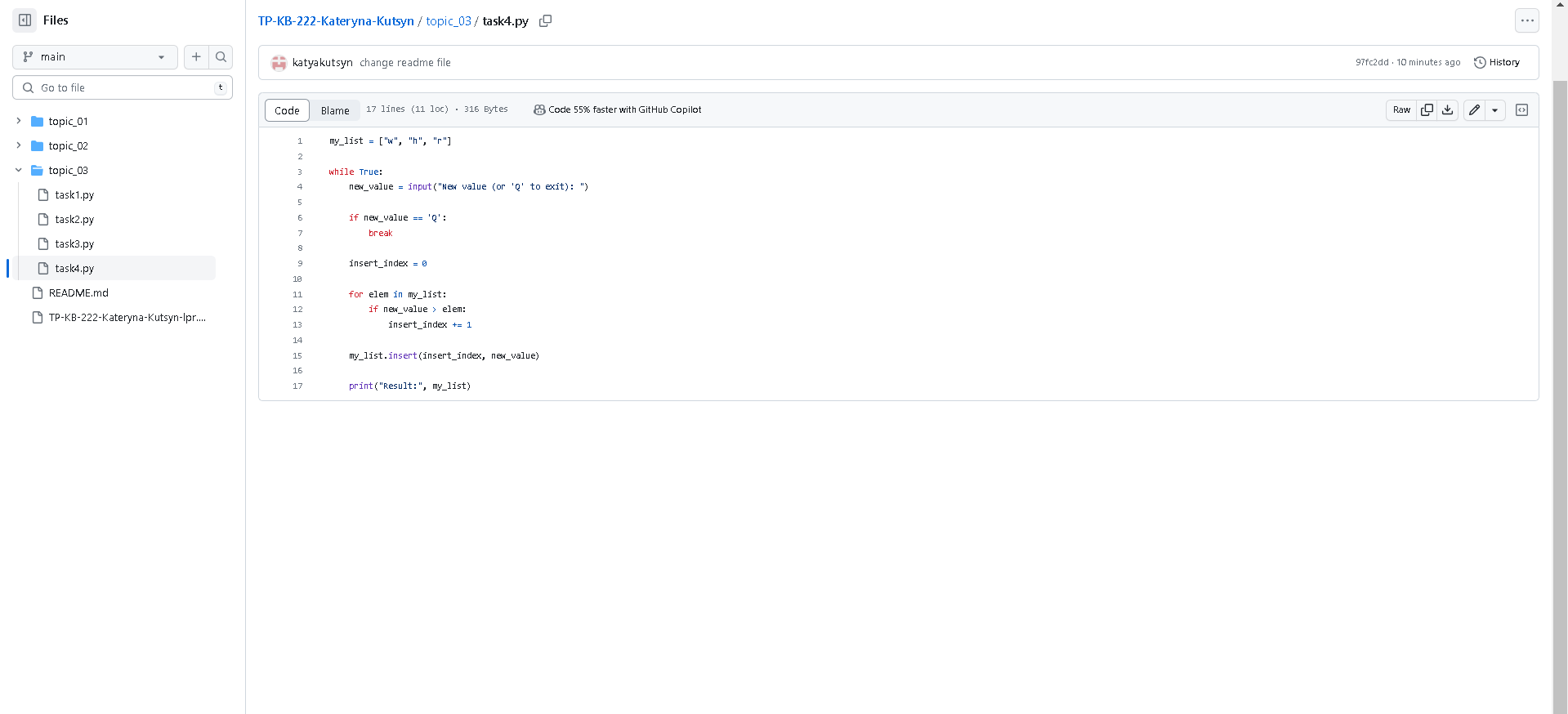
Текст програми:

|  |
| --- |
| my\_list = ["w", "h", "r"]  while True:  new\_value = input("New value (or 'Q' to exit): ")  if new\_value == 'Q':  break  insert\_index = 0  for elem in my\_list:  if new\_value > elem:  insert\_index += 1  my\_list.insert(insert\_index, new\_value)  print("Result:", my\_list) |

Посилання на github:

<https://github.com/katyakutsyn/TP-KB-222-Kateryna-Kutsyn/blob/main/topic_03/task4.py>

Знімок екрану з посилання на github:



Звіт до Теми №4

Виняткові ситуації

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

**getInt function**

Розширити програму калькулятор функцією запитів від користувача, що обробляє виняткові ситуації. getInt

Хід виконання завдання:

1. Відкрила ВСкод
2. Прочитала завдання
3. Виконала завдання
4. Затестила
5. Закинула в звіт

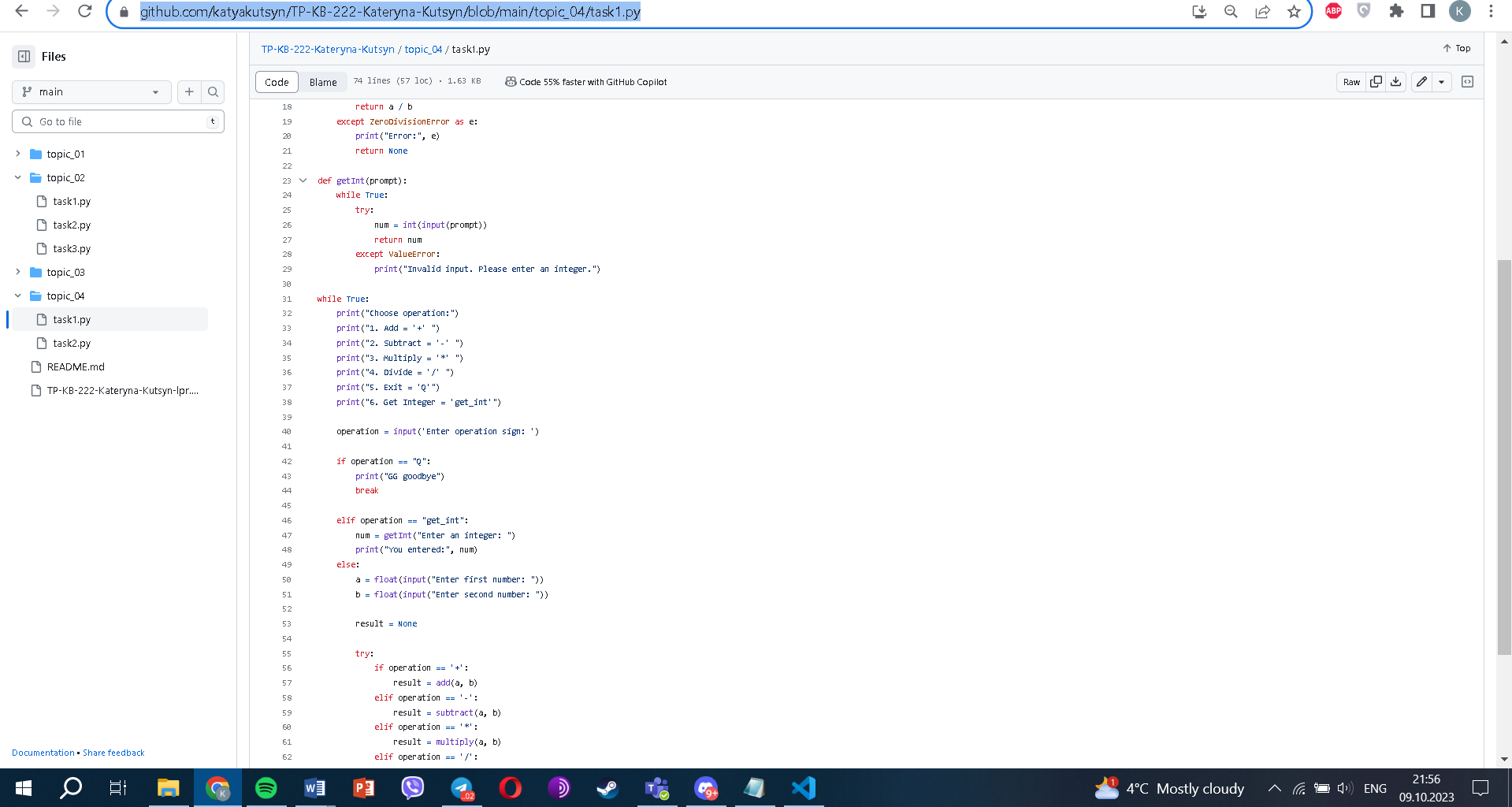
Текст програми:

|  |
| --- |
| import sys  def add(a, b):  return a + b  def subtract(a, b):  return a - b  def multiply(a, b):  return a \* b  def divide(a, b):  try:  if b == 0:  raise ZeroDivisionError("Division by zero is not allowed")  return a / b  except ZeroDivisionError as e:  print("Error:", e)  return None  def getInt(prompt):  while True:  try:  num = int(input(prompt))  return num  except ValueError:  print("Invalid input. Please enter an integer.")  while True:  print("Choose operation:")  print("1. Add = '+' ")  print("2. Subtract = '-' ")  print("3. Multiply = '\*' ")  print("4. Divide = '/' ")  print("5. Exit = 'Q'")  print("6. Get Integer = 'get\_int'")  operation = input('Enter operation sign: ')  if operation == "Q":  print("GG goodbye")  break  elif operation == "get\_int":  num = getInt("Enter an integer: ")  print("You entered:", num)  else:  a = float(input("Enter first number: "))  b = float(input("Enter second number: "))  result = None  try:  if operation == '+':  result = add(a, b)  elif operation == '-':  result = subtract(a, b)  elif operation == '\*':  result = multiply(a, b)  elif operation == '/':  result = divide(a, b)  else:  print("Invalid operation")  continue  if result is not None:  print("Result:", result)  except Exception as e:  print("Error:", e) |

Посилання на github:

<https://github.com/katyakutsyn/TP-KB-222-Kateryna-Kutsyn/blob/main/topic_04/task1.py>

Знімок екрану з посилання на github:



**ZeroDivisionError**

Розширити функцію ділення обробкою виняткової ситуації ділення но нуль. ZeroDivisionError

Хід виконання завдання:

1. Відкрила ВСкод
2. Прочитала завдання
3. Виконала завдання
4. Затестила
5. Закинула в звіт

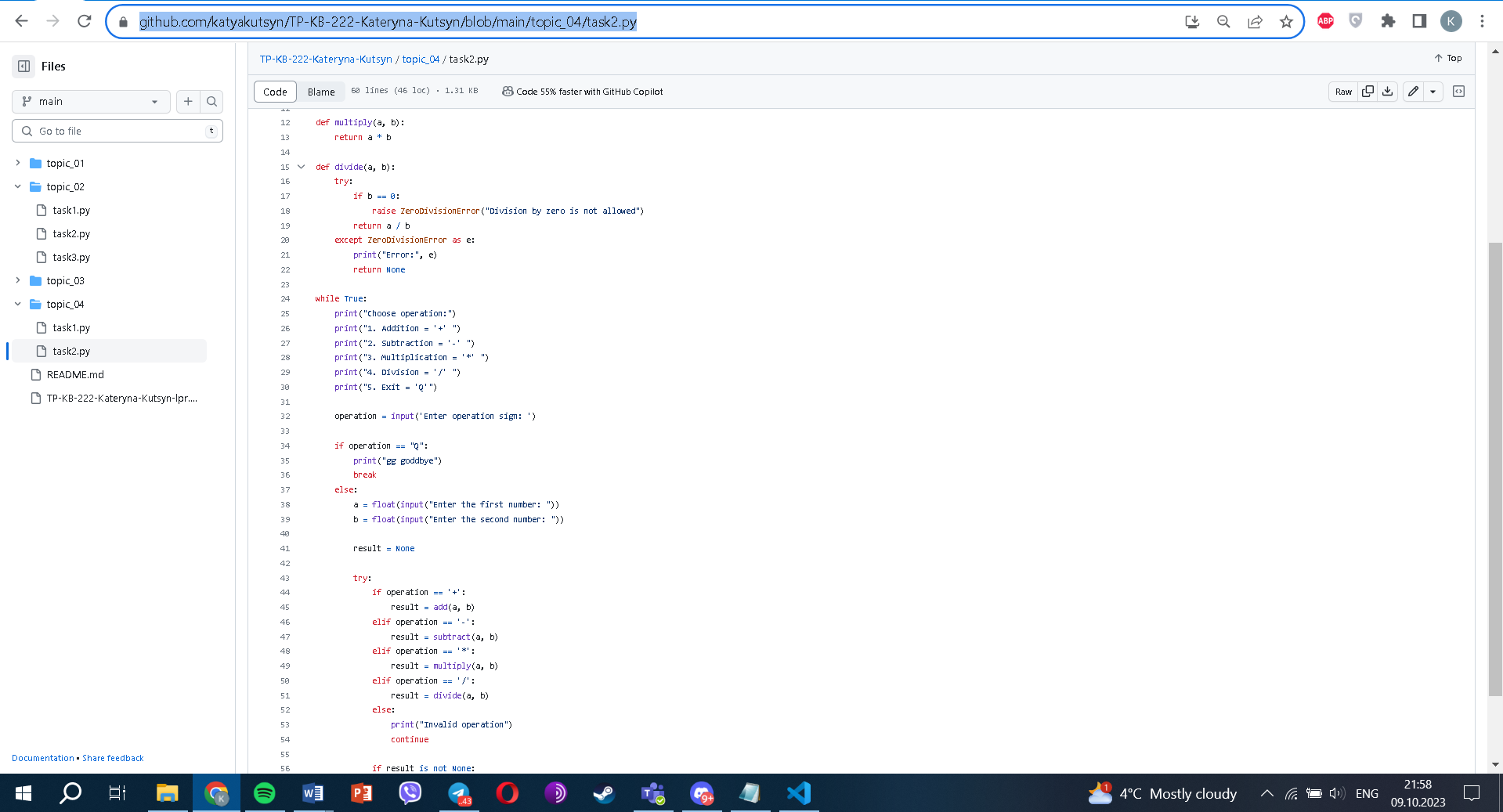
Текст програми:

|  |
| --- |
| import sys  def add(a, b):  return a + b  def subtract(a, b):  return a - b  def multiply(a, b):  return a \* b  def divide(a, b):  try:  if b == 0:  raise ZeroDivisionError("Division by zero is not allowed")  return a / b  except ZeroDivisionError as e:  print("Error:", e)  return None  while True:  print("Choose operation:")  print("1. Addition = '+' ")  print("2. Subtraction = '-' ")  print("3. Multiplication = '\*' ")  print("4. Division = '/' ")  print("5. Exit = 'Q'")    operation = input('Enter operation sign: ')  if operation == "Q":  print("gg goddbye")  break  else:  a = float(input("Enter the first number: "))  b = float(input("Enter the second number: "))  result = None  try:  if operation == '+':  result = add(a, b)  elif operation == '-':  result = subtract(a, b)  elif operation == '\*':  result = multiply(a, b)  elif operation == '/':  result = divide(a, b)  else:  print("Invalid operation")  continue  if result is not None:  print("Result:", result)  except Exception as e:  print("Error:", e) |

Посилання на github:

<https://github.com/katyakutsyn/TP-KB-222-Kateryna-Kutsyn/blob/main/topic_04/task2.py>

Знімок екрану з посилання на github:



Звіт до Теми №5

Використання бібліотек

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

**Створення гри «камень, ножиці та бумага»**

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.

Хід виконання завдання:

1. Відкрила ВСкод
2. Прочитала завдання
3. Виконала завдання
4. Затестила
5. Закинула в звіт

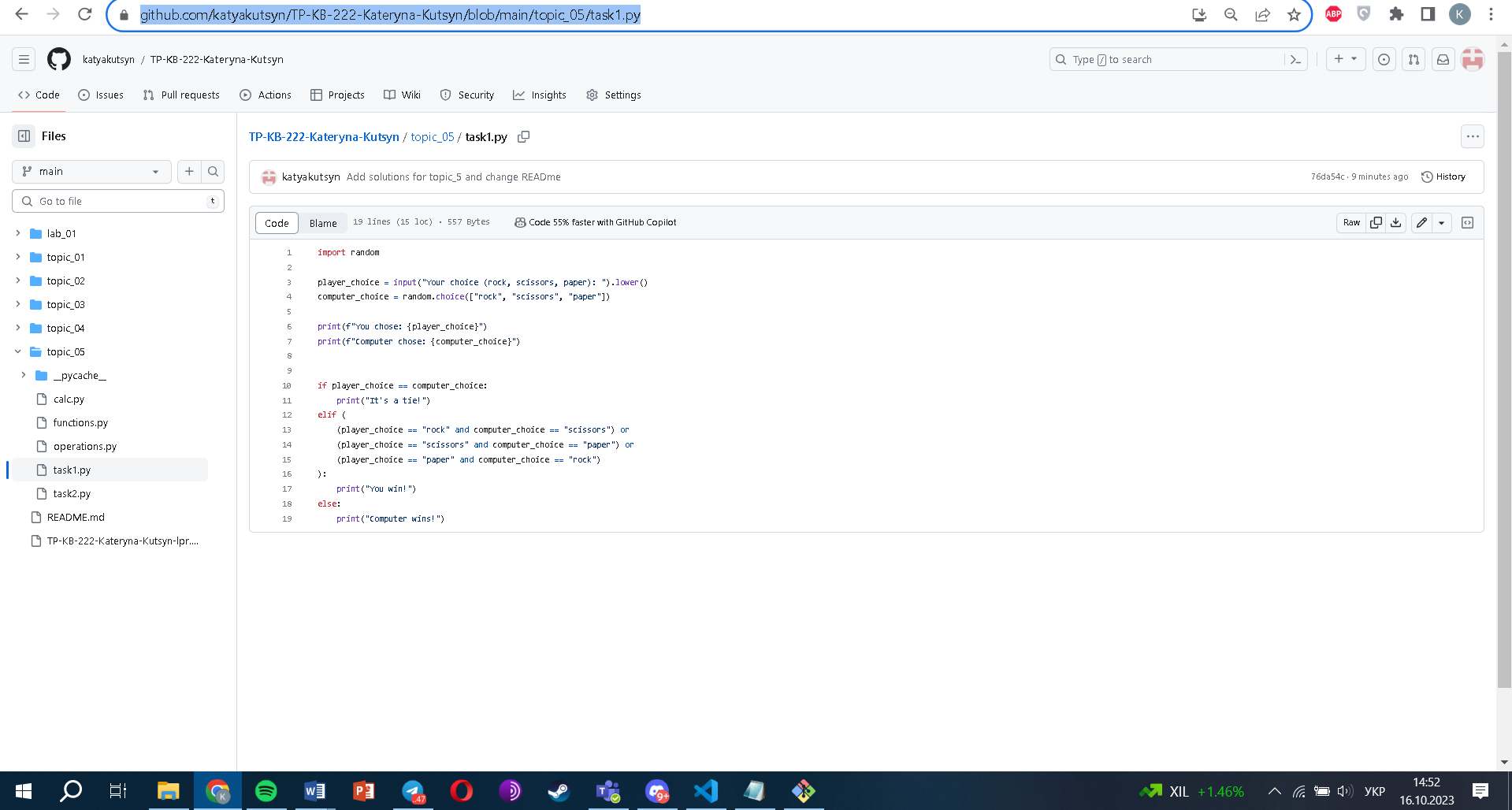
Текст програми:

|  |
| --- |
| import random  player\_choice = input("Your choice (rock, scissors, paper): ").lower()  computer\_choice = random.choice(["rock", "scissors", "paper"])  print(f"You chose: {player\_choice}")  print(f"Computer chose: {computer\_choice}")  if player\_choice == computer\_choice:  print("It's a tie!")  elif (  (player\_choice == "rock" and computer\_choice == "scissors") or  (player\_choice == "scissors" and computer\_choice == "paper") or  (player\_choice == "paper" and computer\_choice == "rock")  ):  print("You win!")  else:  print("Computer wins!") |

Посилання на github:

<https://github.com/katyakutsyn/TP-KB-222-Kateryna-Kutsyn/blob/main/topic_05/task1.py>

Знімок екрану з посилання на github:



**Конвертування іноземної валюти в українську гривню**

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.

Хід виконання завдання:

1. Відкрила ВСкод
2. Прочитала завдання
3. Виконала завдання
4. Затестила
5. Закинула в звіт

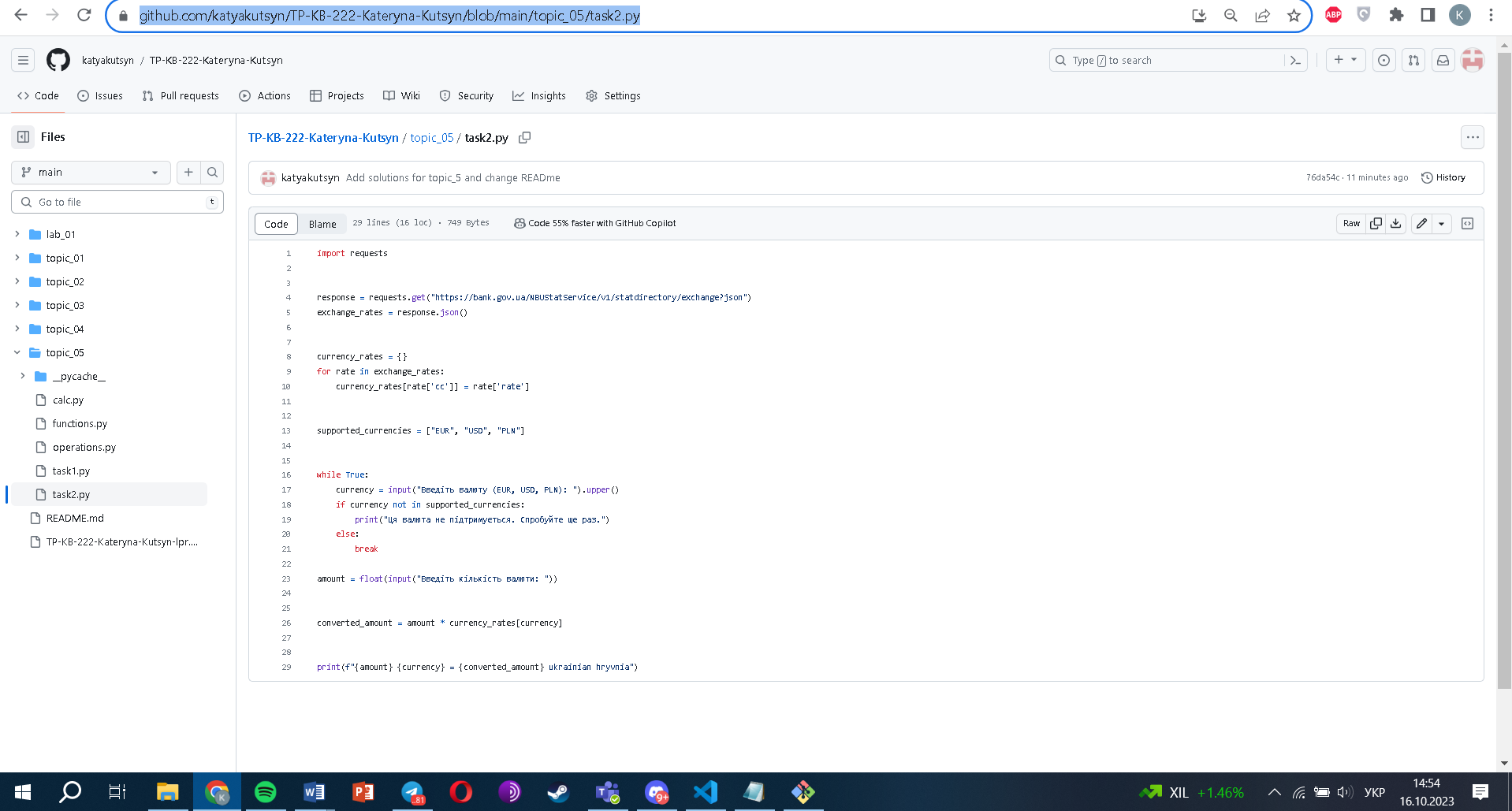
Текст програми:

|  |
| --- |
| import requests  response = requests.get("https://bank.gov.ua/NBUStatService/v1/statdirectory/exchange?json")  exchange\_rates = response.json()  currency\_rates = {}  for rate in exchange\_rates:  currency\_rates[rate['cc']] = rate['rate']  supported\_currencies = ["EUR", "USD", "PLN"]  while True:  currency = input("Введіть валюту (EUR, USD, PLN): ").upper()  if currency not in supported\_currencies:  print("Ця валюта не підтримується. Спробуйте ще раз.")  else:  break  amount = float(input("Введіть кількість валюти: "))  converted\_amount = amount \* currency\_rates[currency]  print(f"{amount} {currency} = {converted\_amount} ukrainian hryvnia") |

Посилання на github:

<https://github.com/katyakutsyn/TP-KB-222-Kateryna-Kutsyn/blob/main/topic_05/task2.py>

Знімок екрану з посилання на github:



**Використання модулів для програми калькулятор**

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. Відкрила ВСкод
2. Прочитала завдання
3. Виконала завдання
4. Затестила
5. Закинула в звіт

Текст програми:

|  |
| --- |
| def add(a, b):  return a + b  def sub(a, b):  return a - b  def mul(a, b):  return a \* b  def div(a, b):  if b != 0:  return a / b  else:  return "Ділення на нуль неможливе" |
| def get\_int\_or\_float\_value(prompt):  while True:  try:  value = float(input(prompt))  return value  except ValueError:  print("Введіть дійсне число.")  def get\_operation():  while True:  operation = input("Оберіть операцію (+, -, \*, /): ")  if operation in ['+', '-', '\*', '/']:  return operation  else:  print("Неправильна операція. Спробуйте ще раз.") |
| import functions  import operations  while True:  a = operations.get\_int\_or\_float\_value("Введіть перше число: ")  b = operations.get\_int\_or\_float\_value("Введіть друге число: ")  operation = operations.get\_operation()  if operation == '+':  result = functions.add(a, b)  elif operation == '-':  result = functions.sub(a, b)  elif operation == '\*':  result = functions.mul(a, b)  elif operation == '/':  result = functions.div(a, b)  print(f"Результат: {result}")  repeat = input("Бажаєте продовжити (yep/no)? ").lower()  if repeat != 'yep':  break |

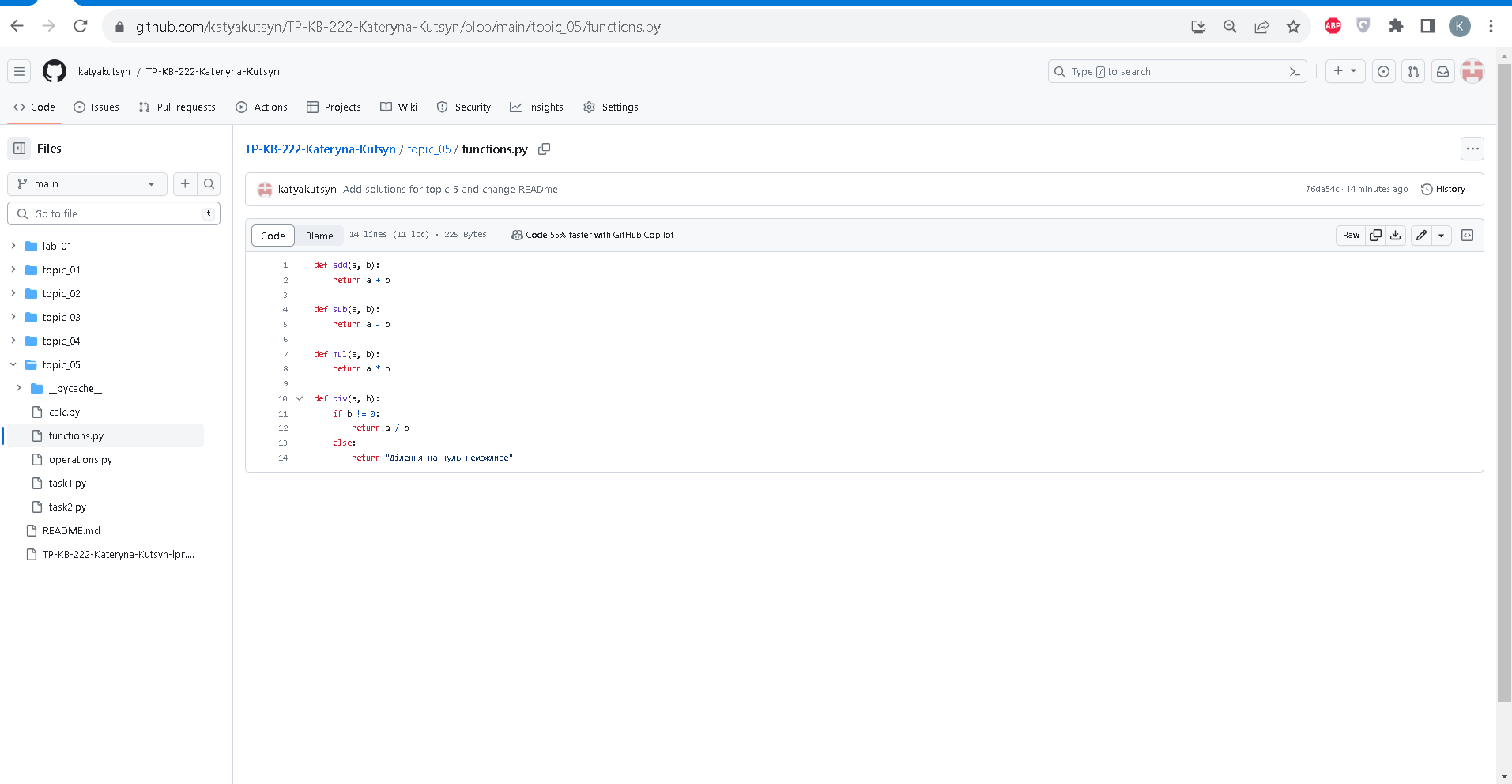
Посилання на github:

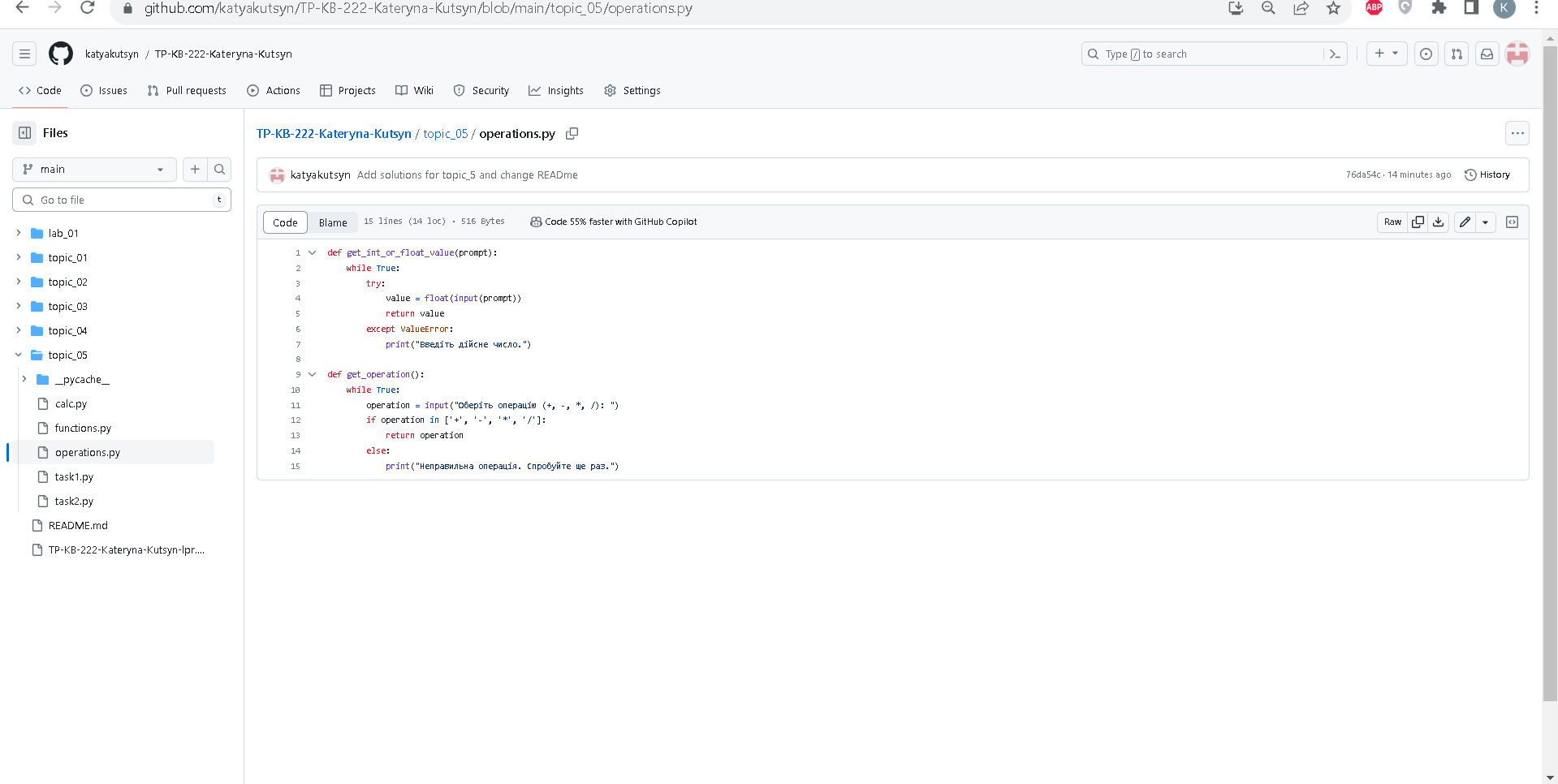
<https://github.com/katyakutsyn/TP-KB-222-Kateryna-Kutsyn/blob/main/topic_05/functions.py>

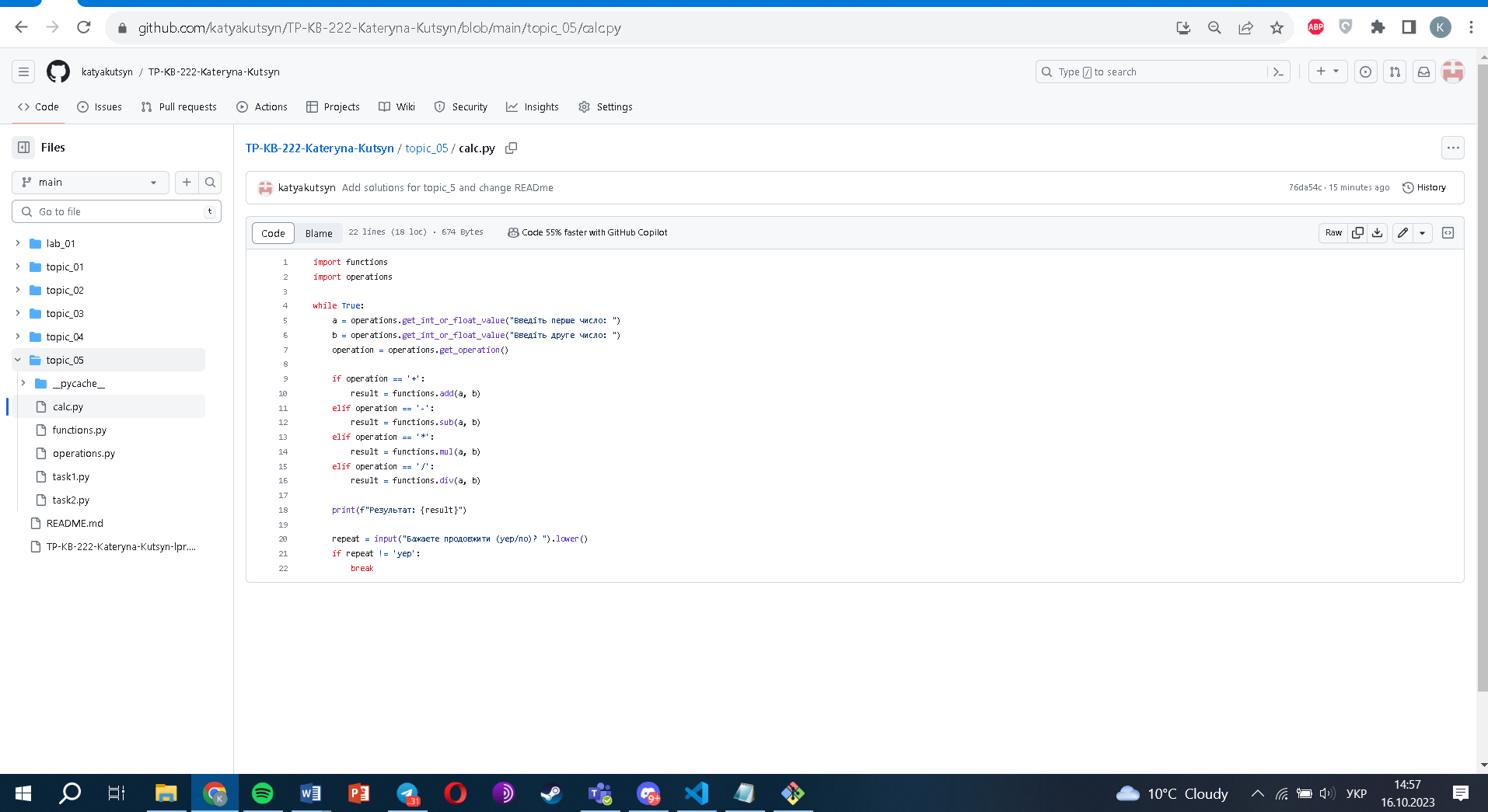
<https://github.com/katyakutsyn/TP-KB-222-Kateryna-Kutsyn/blob/main/topic_05/operations.py>

<https://github.com/katyakutsyn/TP-KB-222-Kateryna-Kutsyn/blob/main/topic_05/calc.py>

Знімок екрану з посилання на github:







Звіт до Теми №6

Робота з файлами

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

**Використання lambda функцій для функції сортування**

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. Відкрила ВСкод
2. Прочитала завдання
3. Виконала завдання
4. Затестила
5. Закинула в звіт

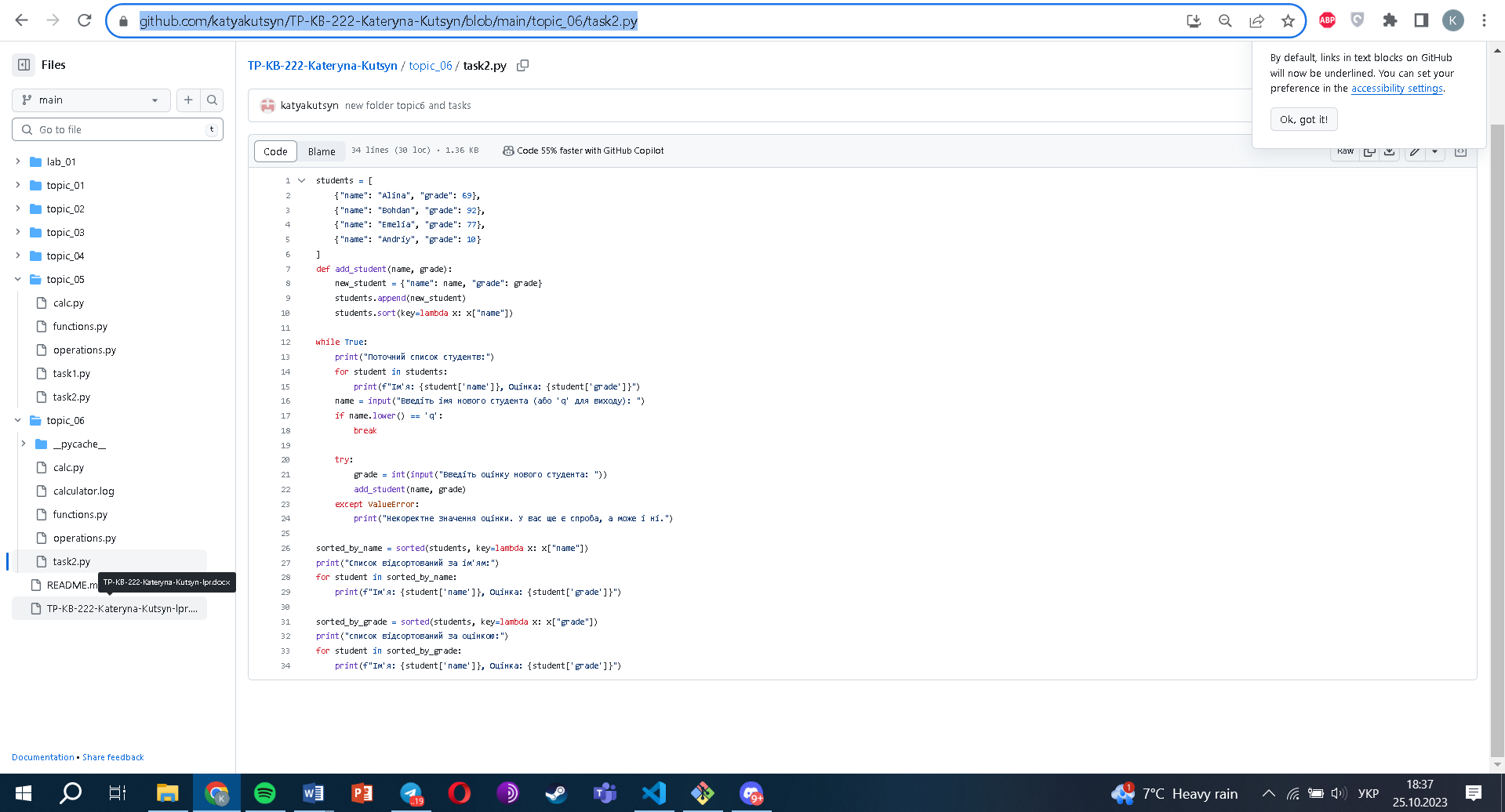
Текст програми:

|  |
| --- |
| students = [  {"name": "Alina", "grade": 69},  {"name": "Bohdan", "grade": 92},  {"name": "Emelia", "grade": 77},  {"name": "Andriy", "grade": 10}  ]  def add\_student(name, grade):  new\_student = {"name": name, "grade": grade}  students.append(new\_student)  students.sort(key=lambda x: x["name"])  while True:  print("Поточний список студентв:")  for student in students:  print(f"Ім'я: {student['name']}, Оцінка: {student['grade']}")  name = input("Введіть імя нового студента (або 'q' для виходу): ")  if name.lower() == 'q':  break  try:  grade = int(input("Введіть оцінку нового студента: "))  add\_student(name, grade)  except ValueError:  print("Некоректне значення оцінки. У вас ще є спроба, а може і ні.")  sorted\_by\_name = sorted(students, key=lambda x: x["name"])  print("Список відсортований за ім'ям:")  for student in sorted\_by\_name:  print(f"Ім'я: {student['name']}, Оцінка: {student['grade']}")  sorted\_by\_grade = sorted(students, key=lambda x: x["grade"])  print("список відсортований за оцінкою:")  for student in sorted\_by\_grade:  print(f"Ім'я: {student['name']}, Оцінка: {student['grade']}") |

Посилання на github:

<https://github.com/katyakutsyn/TP-KB-222-Kateryna-Kutsyn/blob/main/topic_06/task2.py>

Знімок екрану з посилання на github:



**Логування всіх дій в застосунку Калькулятор**

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

Хід виконання завдання:

1. Відкрила ВСкод
2. Прочитала завдання
3. Виконала завдання
4. Затестила
5. Закинула в звіт

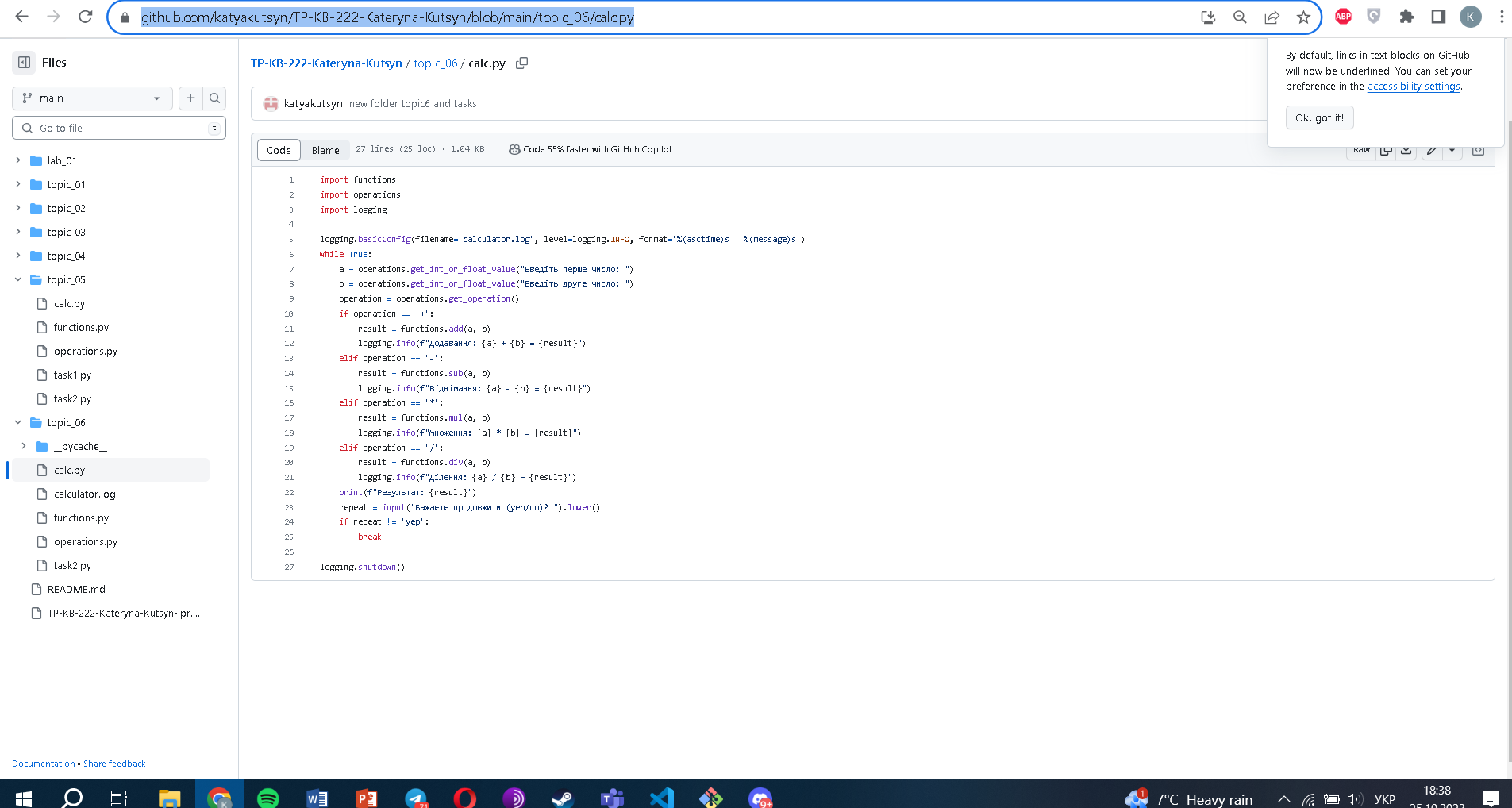
Текст програми:

|  |
| --- |
| import functions  import operations  import logging  logging.basicConfig(filename='calculator.log', level=logging.INFO, format='%(asctime)s - %(message)s')  while True:  a = operations.get\_int\_or\_float\_value("Введіть перше число: ")  b = operations.get\_int\_or\_float\_value("Введіть друге число: ")  operation = operations.get\_operation()  if operation == '+':  result = functions.add(a, b)  logging.info(f"Додавання: {a} + {b} = {result}")  elif operation == '-':  result = functions.sub(a, b)  logging.info(f"Віднімання: {a} - {b} = {result}")  elif operation == '\*':  result = functions.mul(a, b)  logging.info(f"Множення: {a} \* {b} = {result}")  elif operation == '/':  result = functions.div(a, b)  logging.info(f"Ділення: {a} / {b} = {result}")  print(f"Результат: {result}")  repeat = input("Бажаєте продовжити (yep/no)? ").lower()  if repeat != 'yep':  break  logging.shutdown() |

Посилання на github:

<https://github.com/katyakutsyn/TP-KB-222-Kateryna-Kutsyn/blob/main/topic_06/calc.py>

Знімок екрану з посилання на github:



Звіт до Теми №7

ООП

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

**Ознайомитись з документацією та з існуючими за замовченням методами класу**

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

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

Хід виконання завдання:

1. Відкрила ВСкод
2. Прочитала завдання
3. Виконала завдання
4. Затестила
5. Закинула в звіт

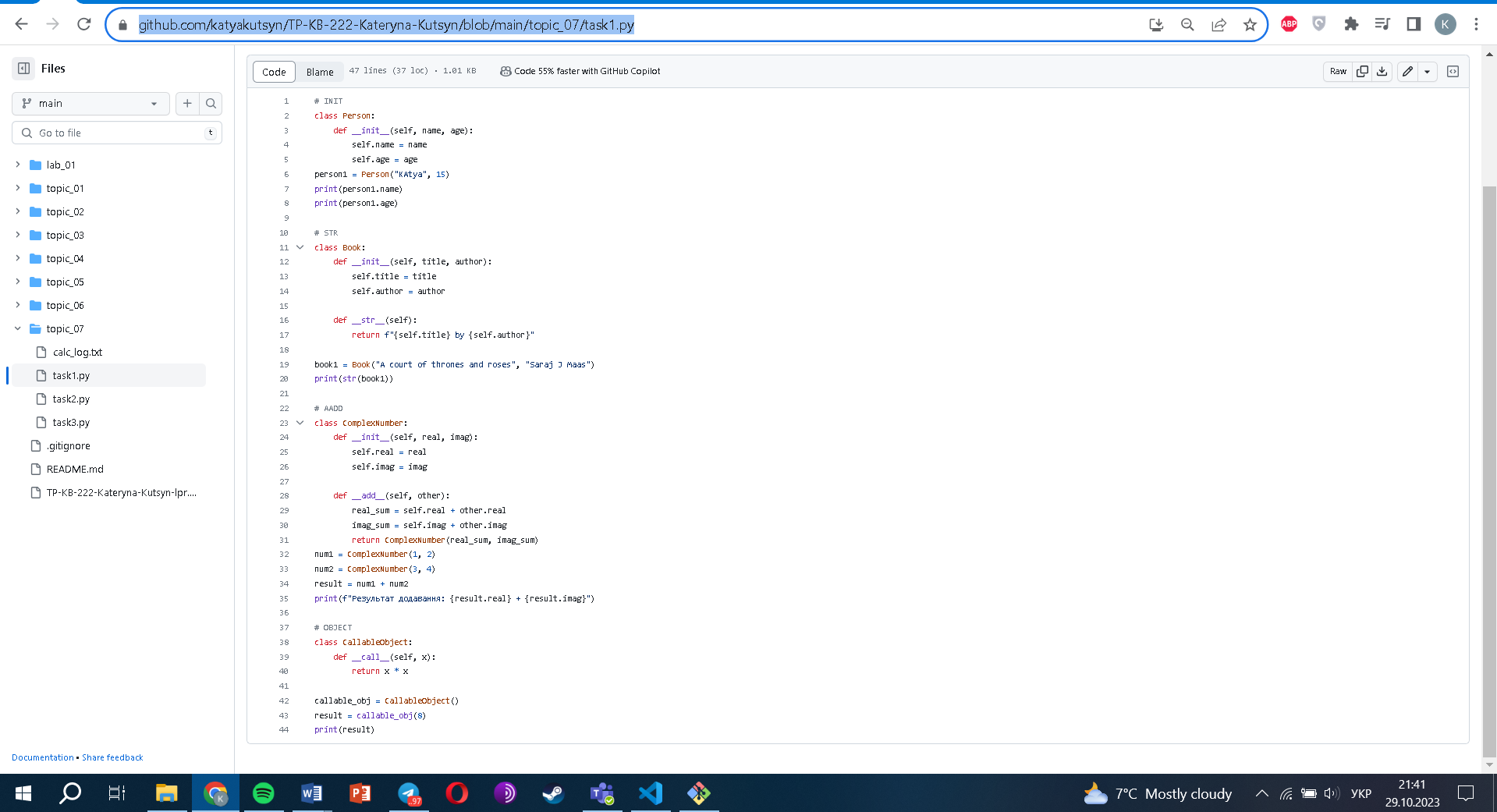
Текст програми:

|  |
| --- |
| # INIT  class Person:  def \_\_init\_\_(self, name, age):  self.name = name  self.age = age  person1 = Person("KAtya", 15)  print(person1.name)  print(person1.age)  # STR  class Book:  def \_\_init\_\_(self, title, author):  self.title = title  self.author = author  def \_\_str\_\_(self):  return f"{self.title} by {self.author}"  book1 = Book("A court of thrones and roses", "Saraj J Maas")  print(str(book1))  # AADD  class ComplexNumber:  def \_\_init\_\_(self, real, imag):  self.real = real  self.imag = imag  def \_\_add\_\_(self, other):  real\_sum = self.real + other.real  imag\_sum = self.imag + other.imag  return ComplexNumber(real\_sum, imag\_sum)  num1 = ComplexNumber(1, 2)  num2 = ComplexNumber(3, 4)  result = num1 + num2  print(f"Результат додавання: {result.real} + {result.imag}")  # OBJECT  class CallableObject:  def \_\_call\_\_(self, x):  return x \* x  callable\_obj = CallableObject()  result = callable\_obj(8)  print(result) |

Посилання на github:

<https://github.com/katyakutsyn/TP-KB-222-Kateryna-Kutsyn/blob/main/topic_07/task1.py>

Знімок екрану з посилання на github:



**Розробити клас Student атрибутами якого э два параметра name та age**

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.

Хід виконання завдання:

1. Відкрила ВСкод
2. Прочитала завдання
3. Виконала завдання
4. Затестила
5. Закинула в звіт

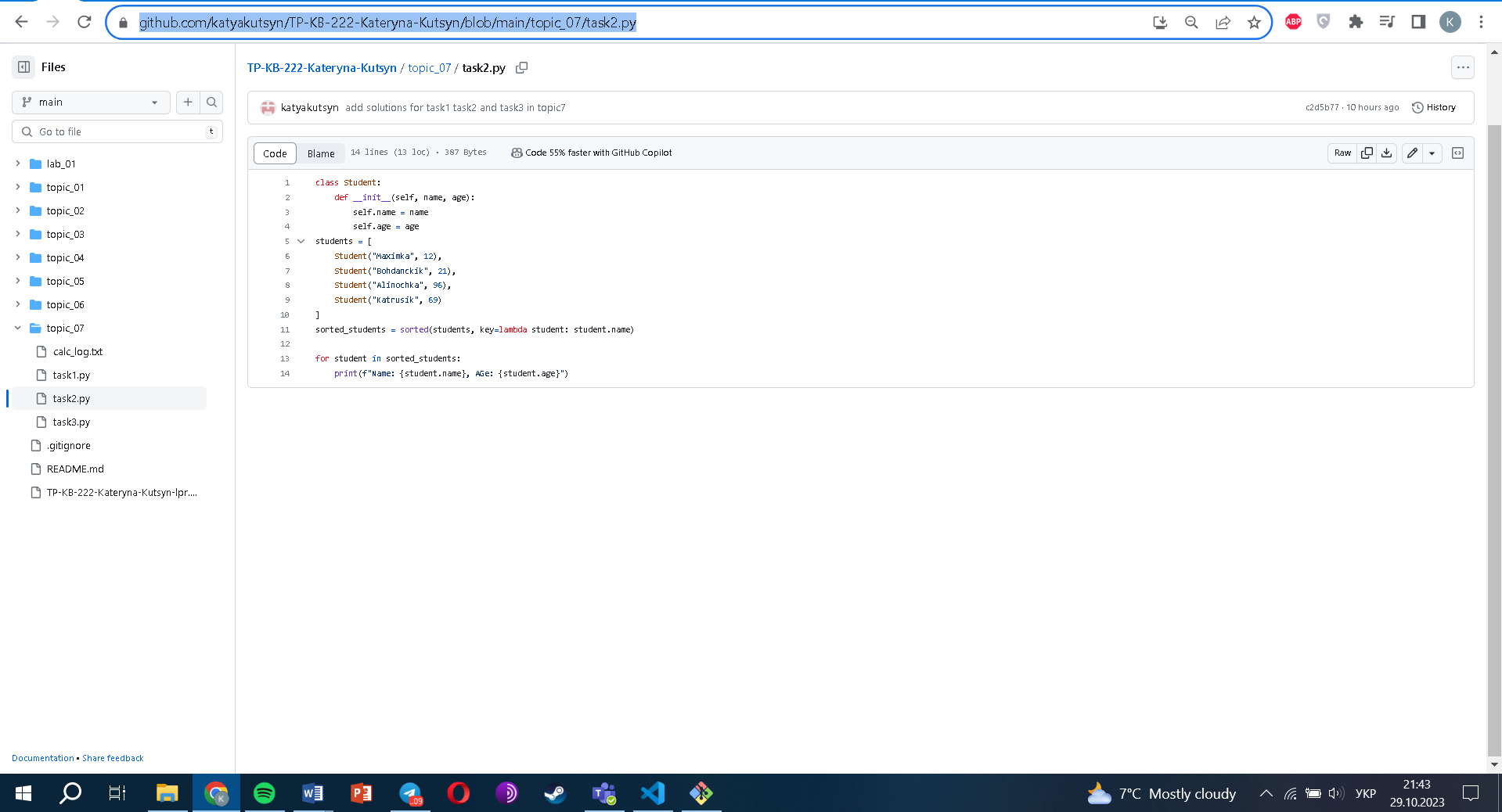
Текст програми:

|  |
| --- |
| class Student:  def \_\_init\_\_(self, name, age):  self.name = name  self.age = age  students = [  Student("Maximka", 12),  Student("Bohdanckik", 21),  Student("Alinochka", 96),  Student("Katrusik", 69)  ]  sorted\_students = sorted(students, key=lambda student: student.name)  for student in sorted\_students:  print(f"Name: {student.name}, AGe: {student.age}") |

Посилання на github:

<https://github.com/katyakutsyn/TP-KB-222-Kateryna-Kutsyn/blob/main/topic_07/task2.py>

Знімок екрану з посилання на github:



**Використовуючи принципи ООП переписати програму Калькулятор**

Використовуючи принципи ООП переписати програму Калькулятор. Завдання має бути виконано використовуючи модульний підхід. Також додати логування усіх дій в окремий текстовий файл, як в минулому завданні

Хід виконання завдання:

1. Відкрила ВСкод
2. Прочитала завдання
3. Виконала завдання
4. Затестила
5. Закинула в звіт

Текст програми:

|  |
| --- |
| 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")  def main():  log\_file = "calс\_log.txt"  calc = Calculator(log\_file)  while True:  print("Operations:")  print("1. Set Operand 1")  print("2. Set Operand 2")  print("3. Addition")  print("4. Subtraction")  print("5. Multiplication")  print("6. Division")  print("Q. Quit")  choice = input("Choose an operation (1/2/3/4/5/6/Q): ")  if choice == 'Q':  print("допобаченя")  break  if choice == '1':  operand = float(input("Enter Operand 1: "))  calc.operand1 = operand  elif choice == '2':  operand = float(input("Enter Operand 2: "))  calc.operand2 = operand  elif choice == '3':  result = calc.add()  print("Result:", result)  elif choice == '4':  result = calc.subtract()  print("Result:", result)  elif choice == '5':  result = calc.multiply()  print("Result:", result)  elif choice == '6':  result = calc.divide()  print("Result:", result)  else:  print("Invalid choice. Please try again.")  if \_\_name\_\_ == "\_\_main\_\_":  main() |

Посилання на github:

<https://github.com/katyakutsyn/TP-KB-222-Kateryna-Kutsyn/blob/main/topic_07/task3.py>

Знімок екрану з посилання на github:

