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

Звіт до Теми №1

Функції та змінні

Бодяк Олександр

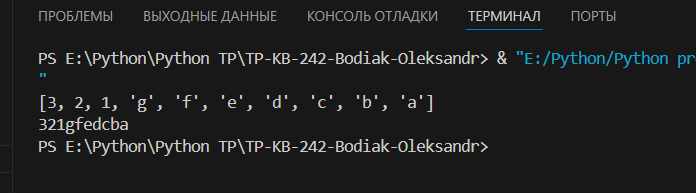
КБ - 242

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:



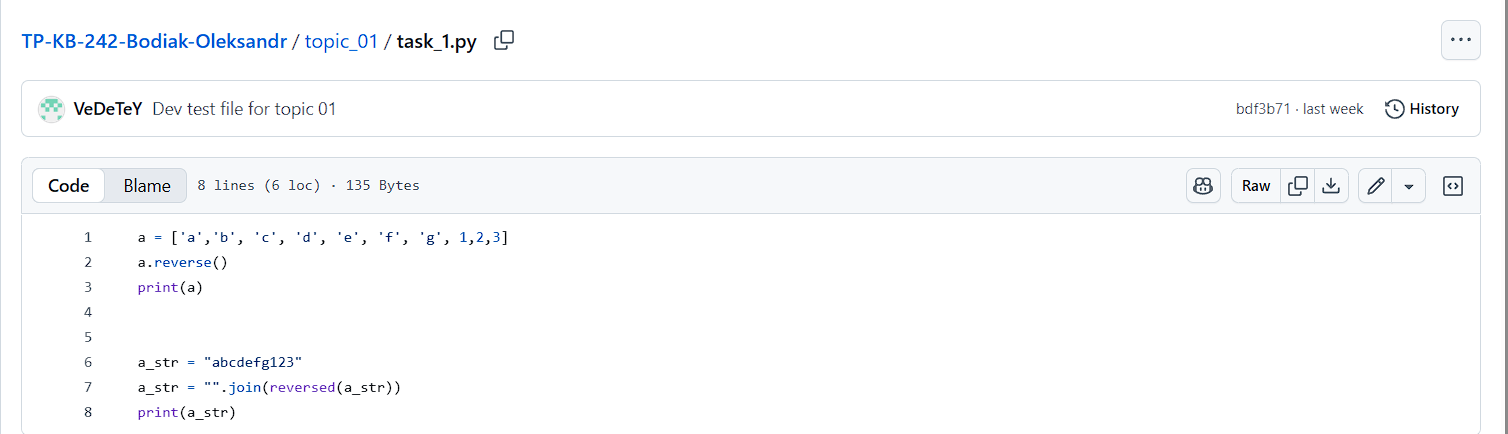
Текст програми:

|  |
| --- |
| a = ['a','b', 'c', 'd', 'e', 'f', 'g', 1,2,3]  a.reverse()  print(a)  a\_str = "abcdefg123"  a\_str = "".join(reversed(a\_str))  print(a\_str) |

Посилання на github:

<https://github.com/VeDeTeY/TP-KB-242-Bodiak-Oleksandr/blob/main/topic_01/task_1.py>

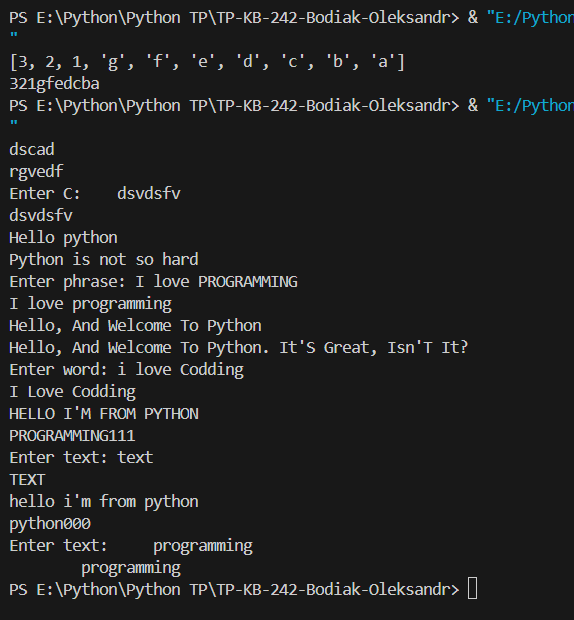
Знімок екрану з посилання на github:



**Дослідження методів класу :**

Виконати тестування функцій, що працюють з рядками: strip(), capitalize(), title(), upper(), lower().

Хід виконання завдання:



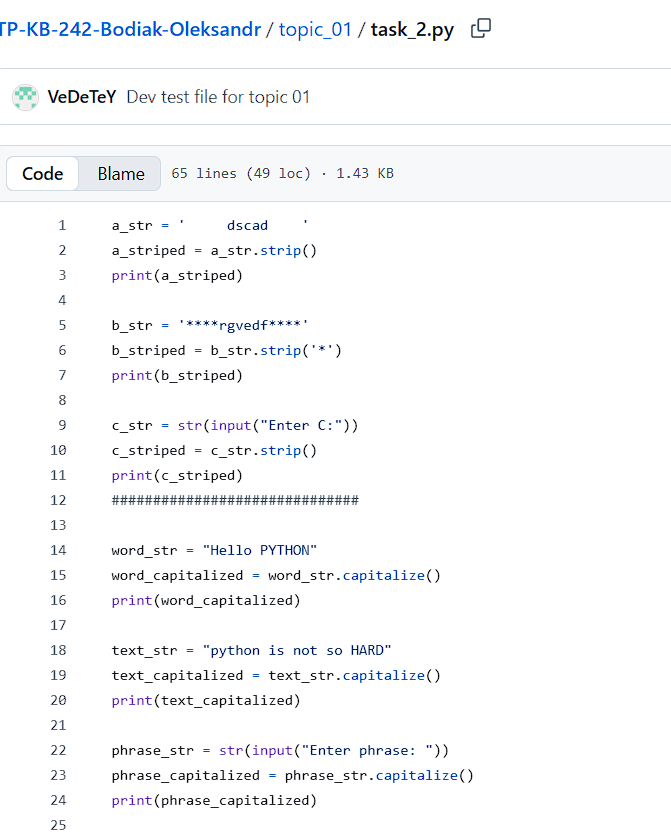
Текст програми:

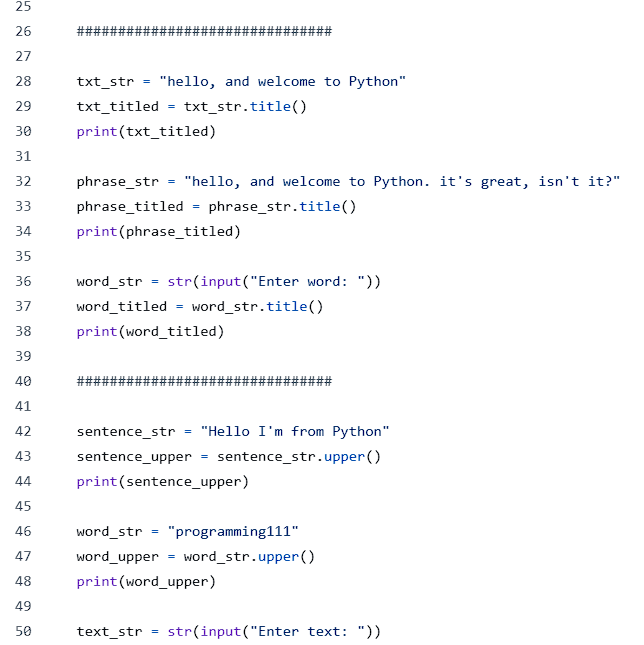
|  |
| --- |
| a\_str = '     dscad    '  a\_striped = a\_str.strip()  print(a\_striped)  b\_str = '\*\*\*\*rgvedf\*\*\*\*'  b\_striped = b\_str.strip('\*')  print(b\_striped)  c\_str = str(input("Enter C:"))  c\_striped = c\_str.strip()  print(c\_striped)  ##############################  word\_str = "Hello PYTHON"  word\_capitalized = word\_str.capitalize()  print(word\_capitalized)  text\_str = "python is not so HARD"  text\_capitalized = text\_str.capitalize()  print(text\_capitalized)  phrase\_str = str(input("Enter phrase: "))  phrase\_capitalized = phrase\_str.capitalize()  print(phrase\_capitalized)  ###############################  txt\_str = "hello, and welcome to Python"  txt\_titled = txt\_str.title()  print(txt\_titled)  phrase\_str = "hello, and welcome to Python. it's great, isn't it?"  phrase\_titled = phrase\_str.title()  print(phrase\_titled)  word\_str = str(input("Enter word: "))  word\_titled = word\_str.title()  print(word\_titled)  ###############################  sentence\_str = "Hello I'm from Python"  sentence\_upper = sentence\_str.upper()  print(sentence\_upper)  word\_str = "programming111"  word\_upper = word\_str.upper()  print(word\_upper)  text\_str = str(input("Enter text: "))  text\_upper = text\_str.upper()  print(text\_upper)  ###############################  text\_str = "Hello I'm from Python"  text\_lower = text\_str.lower()  print(text\_lower)  word\_str = "PYTHON000"  word\_lower = word\_str.lower()  print(word\_lower)  text\_str = str(input("Enter text: "))  text\_lower = text\_str.lower()  print(text\_lower) |

Посилання на github:

<https://github.com/VeDeTeY/TP-KB-242-Bodiak-Oleksandr/blob/main/topic_01/task_2.py>

Знімок екрану з посилання на github:

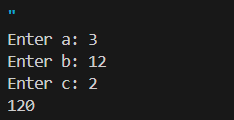






**Написати функцію пошуку дискримінанту:**

Хід виконання завдання:

****

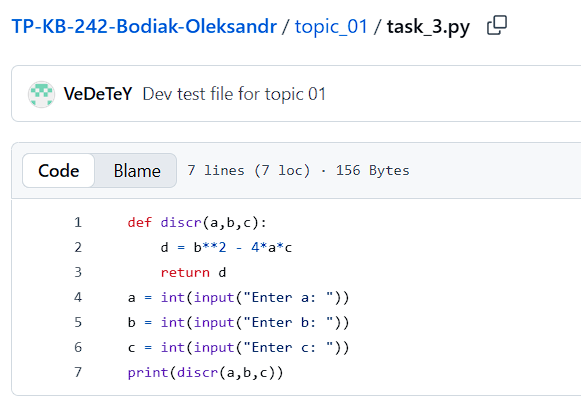
Текст програми:

|  |
| --- |
| def discr(a,b,c):      d = b\*\*2 - 4\*a\*c      return d  a = int(input("Enter a: "))  b = int(input("Enter b: "))  c = int(input("Enter c: "))  print(discr(a,b,c)) |

Посилання на github:

<https://github.com/VeDeTeY/TP-KB-242-Bodiak-Oleksandr/blob/main/topic_01/task_3.py>

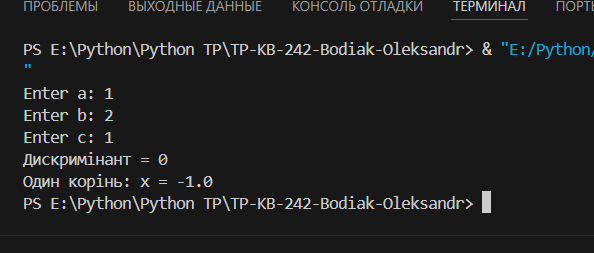
Знімок екрану з посилання на github:



Тема 2 :

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

Хід виконання завдання:



Текст програми :

|  |
| --- |
| import math  def discr(a, b, c):      d = b\*\*2 - 4\*a\*c      return d  # функція для пошуку коренів  def quadro(a, b, c):      d = discr(a, b, c) # виклик функції для обчислення дискримінанта      print(f"Дискримінант = {d}")        if d > 0:          x1 = (-b + math.sqrt(d)) / (2 \* a)          x2 = (-b - math.sqrt(d)) / (2 \* a)          return f"Два корені: x1 = {x1}, x2 = {x2}"      elif d == 0:          x = -b / (2 \* a)          return f"Один корінь: x = {x}"      else:          return "Коренів немає (дискримінант < 0)"  # введення даних  a = int(input("Enter a: "))  b = int(input("Enter b: "))  c = int(input("Enter c: "))  # виклик функції  print(quadro(a, b, c)) |

Посилання на github:

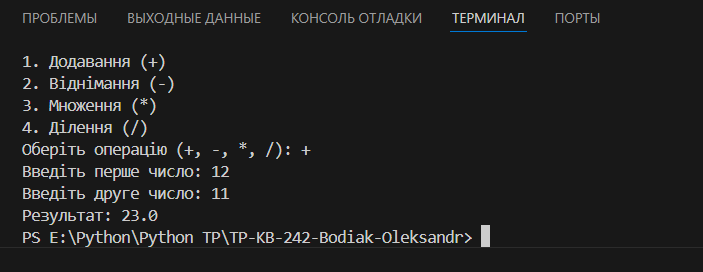
<https://github.com/VeDeTeY/TP-KB-242-Bodiak-Oleksandr/blob/main/topic_02/task_1.py>

Знімок екрану з посилання на github:



1. Написати програму калькулятор використовуючи **if else** конструкцію. Кожна операція має бути виконана в окремій функції.

Хід виконання завдання:



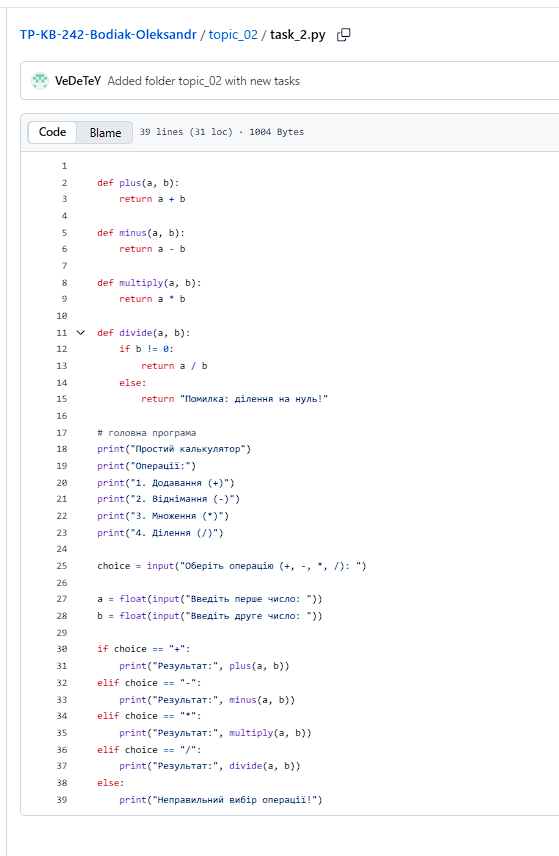
Текст програми :

|  |
| --- |
| def plus(a, b):      return a + b  def minus(a, b):      return a - b  def multiply(a, b):      return a \* b  def divide(a, b):      if b != 0:          return a / b      else:          return "Помилка: ділення на нуль!"  # головна програма  print("Простий калькулятор")  print("Операції:")  print("1. Додавання (+)")  print("2. Віднімання (-)")  print("3. Множення (\*)")  print("4. Ділення (/)")  choice = input("Оберіть операцію (+, -, \*, /): ")  a = float(input("Введіть перше число: "))  b = float(input("Введіть друге число: "))  if choice == "+":      print("Результат:", plus(a, b))  elif choice == "-":      print("Результат:", minus(a, b))  elif choice == "\*":      print("Результат:", multiply(a, b))  elif choice == "/":      print("Результат:", divide(a, b))  else:      print("Неправильний вибір операції!") |

Посилання на github:

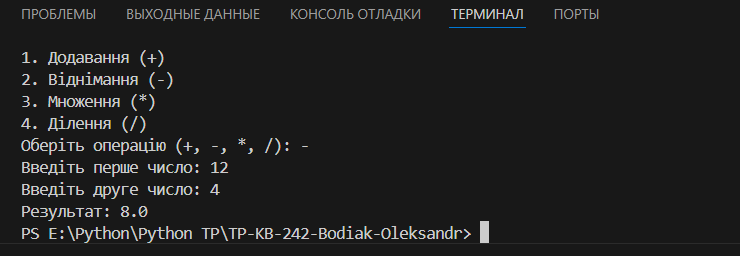
<https://github.com/VeDeTeY/TP-KB-242-Bodiak-Oleksandr/blob/main/topic_02/task_2.py>

Знімок екрану з посилання на github:



1. Написати програму калькулятор використовуючи **match** конструкцію. Кожна операція має бути виконана в окремій функції.

Хід виконання завдання:



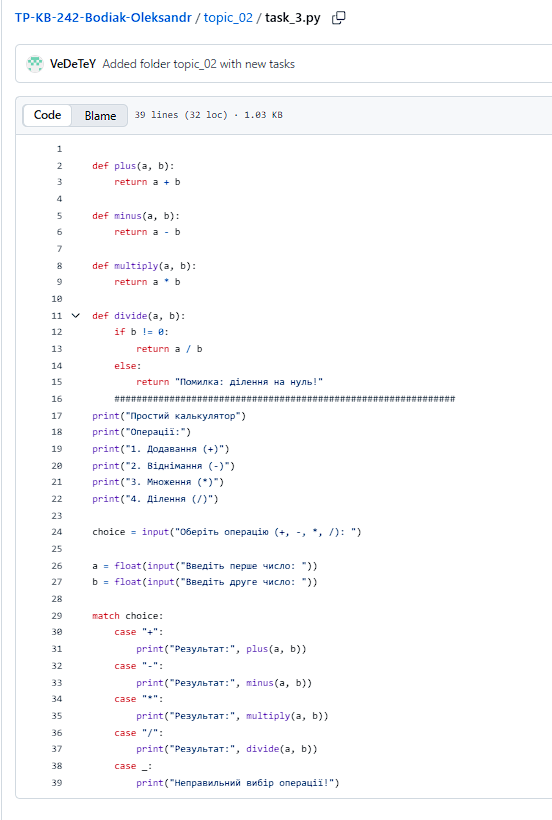
Текст програми :

|  |
| --- |
| def plus(a, b):      return a + b  def minus(a, b):      return a - b  def multiply(a, b):      return a \* b  def divide(a, b):      if b != 0:          return a / b      else:          return "Помилка: ділення на нуль!"      ##############################################################  print("Простий калькулятор")  print("Операції:")  print("1. Додавання (+)")  print("2. Віднімання (-)")  print("3. Множення (\*)")  print("4. Ділення (/)")  choice = input("Оберіть операцію (+, -, \*, /): ")  a = float(input("Введіть перше число: "))  b = float(input("Введіть друге число: "))  match choice:      case "+":          print("Результат:", plus(a, b))      case "-":          print("Результат:", minus(a, b))      case "\*":          print("Результат:", multiply(a, b))      case "/":          print("Результат:", divide(a, b))      case \_:          print("Неправильний вибір операції!") |

Посилання на github:

<https://github.com/VeDeTeY/TP-KB-242-Bodiak-Oleksandr/blob/main/topic_02/task_3.py>

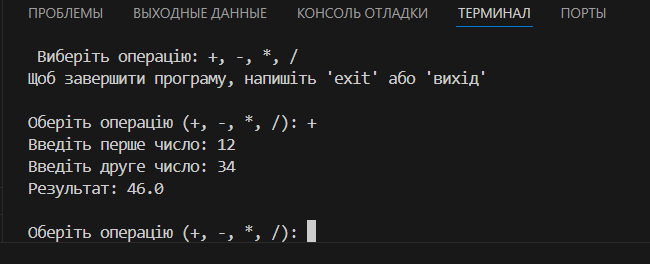
Знімок екрану з посилання на github:



**Тема 3:**

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:



Текст програми:

|  |
| --- |
| ef plus(a, b):      return a + b  def minus(a, b):      return a - b  def multiply(a, b):      return a \* b  def divide(a, b):      if b != 0:          return a / b      else:          return "Помилка: ділення на нуль!"  print("Калькулятор")  print(" Виберіть операцію: +, -, \*, /")  print("Щоб завершити програму, напишіть 'exit' або 'вихід'")  while True:      choice = input("\nОберіть операцію (+, -, \*, /): ")        if choice.lower() in ["exit", "вихід"]:          print("Роботу закінчено")          break        try:          a = float(input("Введіть перше число: "))          b = float(input("Введіть друге число: "))      except ValueError:          print("Помилка: потрібно вводити числа!")          continue        match choice:          case "+":              print("Результат:", plus(a, b))          case "-":              print("Результат:", minus(a, b))          case "\*":              print("Результат:", multiply(a, b))          case "/":              print("Результат:", divide(a, b))          case \_:              print("Неправильний вибір операції!") |

Посилання на github:

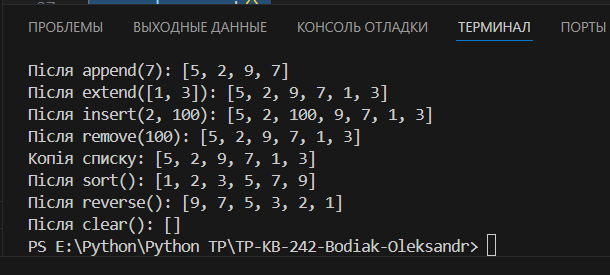
<https://github.com/VeDeTeY/TP-KB-242-Bodiak-Oleksandr/blob/main/topic_03/task_1.py>

Знімок екрану з посилання на github:



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

Хід виконання завдання:



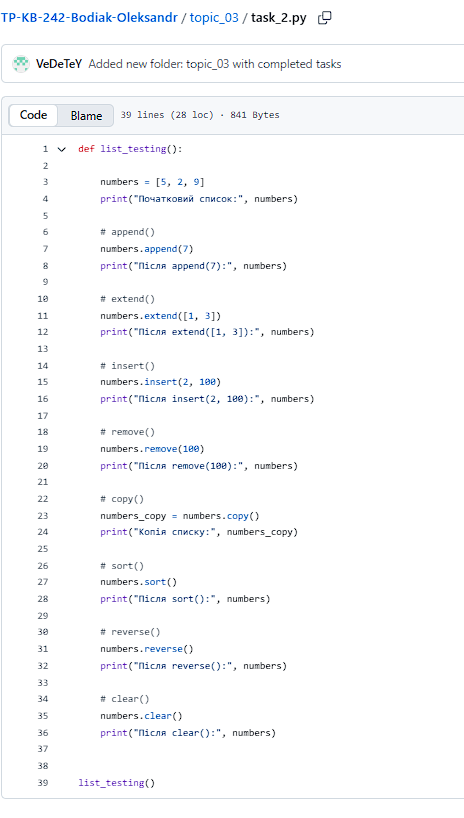
Текст програми :

|  |
| --- |
| numbers = [5, 2, 9]      print("Початковий список:", numbers)        # append()      numbers.append(7)      print("Після append(7):", numbers)        # extend()      numbers.extend([1, 3])      print("Після extend([1, 3]):", numbers)        # insert()      numbers.insert(2, 100)      print("Після insert(2, 100):", numbers)        # remove()      numbers.remove(100)      print("Після remove(100):", numbers)        # copy()      numbers\_copy = numbers.copy()      print("Копія списку:", numbers\_copy)        # sort()      numbers.sort()      print("Після sort():", numbers)        # reverse()      numbers.reverse()      print("Після reverse():", numbers)        # clear()      numbers.clear()      print("Після clear():", numbers)  list\_testing() |

Посилання на github:

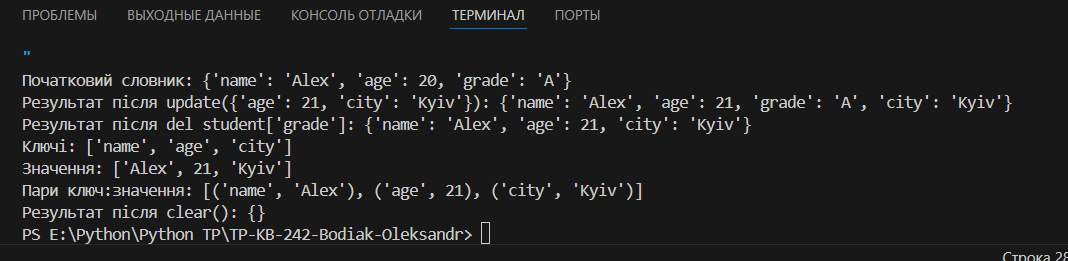
<https://github.com/VeDeTeY/TP-KB-242-Bodiak-Oleksandr/blob/main/topic_03/task_2.py>

Знімок екрану з посилання на github:



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

Хід виконання завдання:



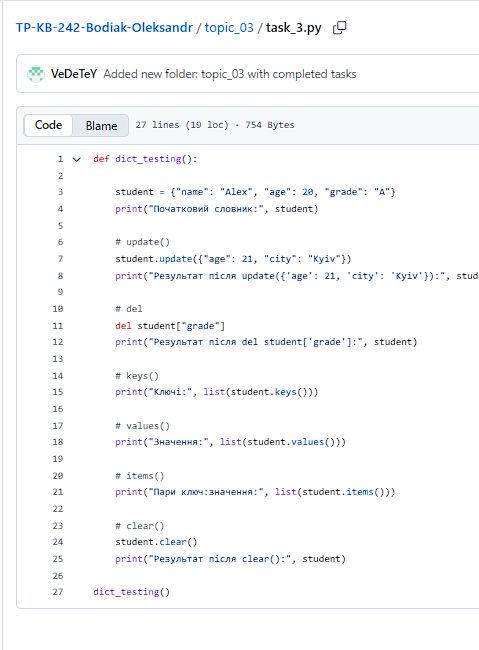
Текст програми :

|  |
| --- |
| def dict\_testing():      student = {"name": "Alex", "age": 20, "grade": "A"}      print("Початковий словник:", student)        # update()      student.update({"age": 21, "city": "Kyiv"})      print("Результат після update({'age': 21, 'city': 'Kyiv'}):", student)        # del      del student["grade"]      print("Результат після del student['grade']:", student)        # keys()      print("Ключі:", list(student.keys()))        # values()      print("Значення:", list(student.values()))        # items()      print("Пари ключ:значення:", list(student.items()))        # clear()      student.clear()      print("Результат після clear():", student)  dict\_testing() |

Посилання на github:

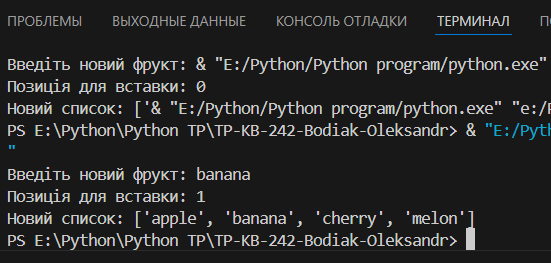
<https://github.com/VeDeTeY/TP-KB-242-Bodiak-Oleksandr/blob/main/topic_03/task_3.py>

Знімок екрану з посилання на github:



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

Хід виконання завдання:



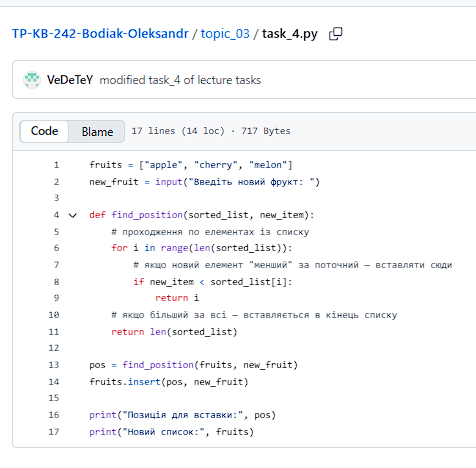
Текст програми:

|  |
| --- |
| fruits = ["apple", "cherry", "melon"]  new\_fruit = input("Введіть новий фрукт: ")  def find\_position(sorted\_list, new\_item):      # проходження по елементах із списку      for i in range(len(sorted\_list)):          # якщо новий елемент "менший" за поточний — вставляти сюди          if new\_item < sorted\_list[i]:              return i      # якщо більший за всі — вставляється в кінець списку      return len(sorted\_list)  pos = find\_position(fruits, new\_fruit)  fruits.insert(pos, new\_fruit)  print("Позиція для вставки:", pos)  print("Новий список:", fruits) |

Посилання на github:

<https://github.com/VeDeTeY/TP-KB-242-Bodiak-Oleksandr/blob/main/topic_03/task_4.py>

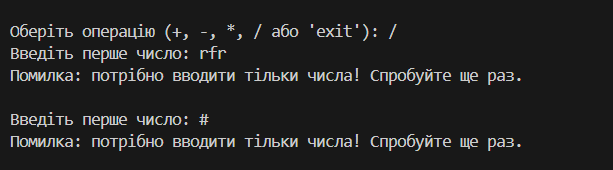
Знімок екрану з посилання на github:



**Тема 4:**

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

Хід виконання завдання:

****

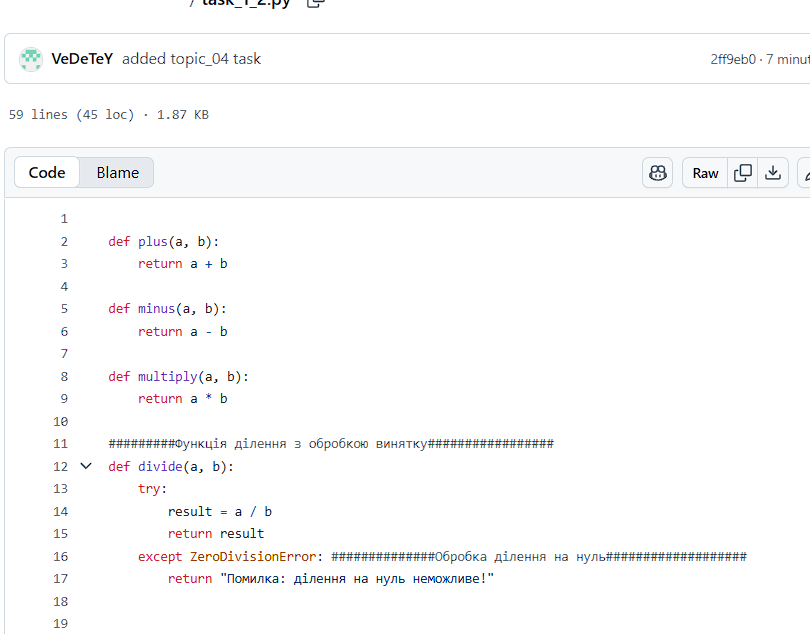
Текст програми:

|  |
| --- |
| def plus(a, b):      return a + b  def minus(a, b):      return a - b  def multiply(a, b):      return a \* b  #########Функція ділення з обробкою винятку#################  def divide(a, b):      try:          result = a / b          return result      except ZeroDivisionError: ##############Обробка ділення на нуль###################          return "Помилка: ділення на нуль неможливе!"  ##############Функція введення чисел із перевіркою###############  def get\_numbers():      while True:          try:              a = float(input("Введіть перше число: "))              b = float(input("Введіть друге число: "))              return a, b          except ValueError: ##############Обробка некоректного вводу#########              print("Помилка: потрібно вводити тільки числа! Спробуйте ще раз.\n")  #Основна програма  print("Простий калькулятор")  print("Операції: +, -, \*, /")  print("Для виходу введіть 'exit'\n")  while True:      choice = input("Оберіть операцію (+, -, \*, / або 'exit'): ")      if choice.lower() == "exit":          print("Роботу завершено. До побачення!")          break      if choice not in ["+", "-", "\*", "/"]:          print("Неправильний вибір операції! Спробуйте ще раз.\n")          continue      a, b = get\_numbers()      match choice:          case "+":              print("Результат:", plus(a, b))          case "-":              print("Результат:", minus(a, b))          case "\*":              print("Результат:", multiply(a, b))          case "/":              print("Результат:", divide(a, b))      print() |

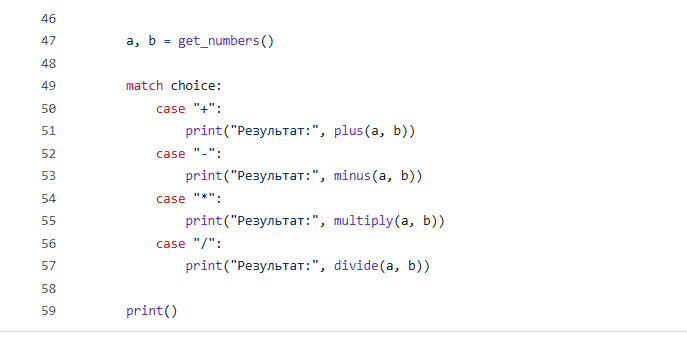
Посилання на github:

<https://github.com/VeDeTeY/TP-KB-242-Bodiak-Oleksandr/tree/main/topic_04>

Знімок екрану з посилання на github:

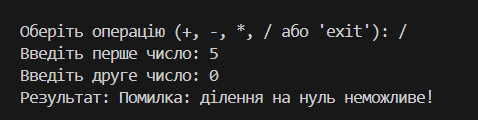
****

****

****

2) Розширити функцію ділення обробкою виняткової ситуації ділення но нуль

Хід виконання завдання:

****

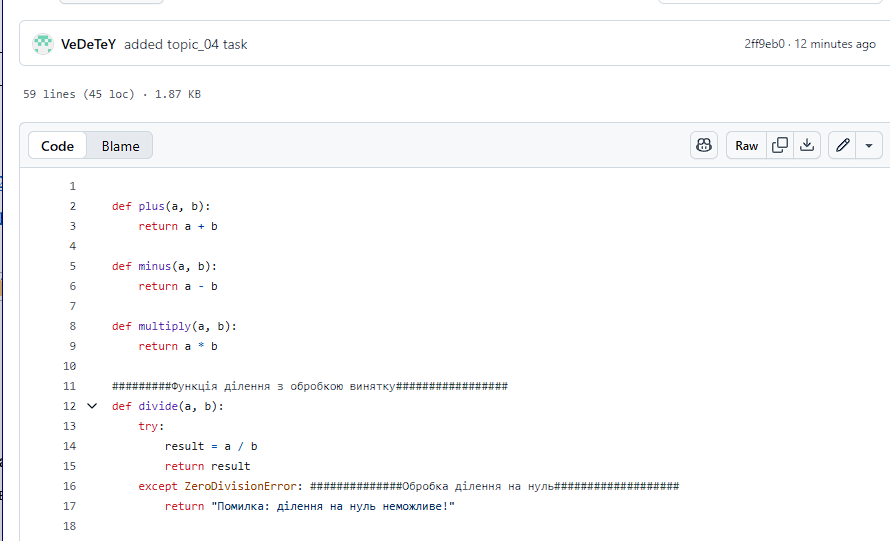
Текст програми:

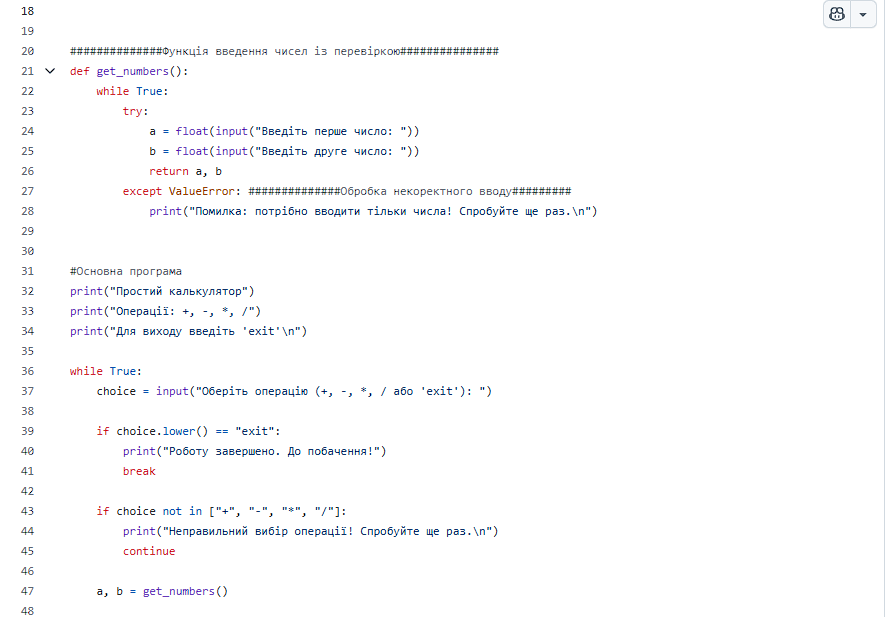
|  |
| --- |
| def plus(a, b):      return a + b  def minus(a, b):      return a - b  def multiply(a, b):      return a \* b  #########Функція ділення з обробкою винятку#################  def divide(a, b):      try:          result = a / b          return result      except ZeroDivisionError: ##############Обробка ділення на нуль###################          return "Помилка: ділення на нуль неможливе!"  ##############Функція введення чисел із перевіркою###############  def get\_numbers():      while True:          try:              a = float(input("Введіть перше число: "))              b = float(input("Введіть друге число: "))              return a, b          except ValueError: ##############Обробка некоректного вводу#########              print("Помилка: потрібно вводити тільки числа! Спробуйте ще раз.\n")  #Основна програма  print("Простий калькулятор")  print("Операції: +, -, \*, /")  print("Для виходу введіть 'exit'\n")  while True:      choice = input("Оберіть операцію (+, -, \*, / або 'exit'): ")      if choice.lower() == "exit":          print("Роботу завершено. До побачення!")          break      if choice not in ["+", "-", "\*", "/"]:          print("Неправильний вибір операції! Спробуйте ще раз.\n")          continue      a, b = get\_numbers()      match choice:          case "+":              print("Результат:", plus(a, b))          case "-":              print("Результат:", minus(a, b))          case "\*":              print("Результат:", multiply(a, b))          case "/":              print("Результат:", divide(a, b))      print() |

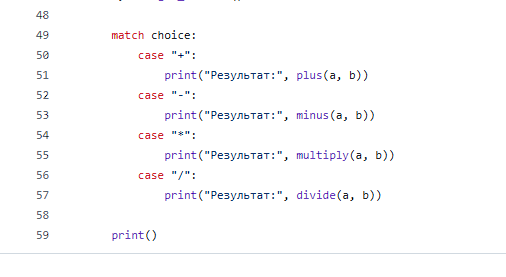
Посилання на github:

<https://github.com/VeDeTeY/TP-KB-242-Bodiak-Oleksandr/blob/main/topic_04/task_1_2.py>

Знімок екрану з посилання на github:







Висновок : виконав завдання які були надані у методичних вказівках. Під час виконання завдань дослідив та розібрав теми: створення нового репозиторію на github, написання функції пошуку коренів квадратного рівняння використовуючи функцію розрахунку дискримінанту, написання програми калькулятор використовуючи **if else** конструкцію, написання програми калькулятор використовуючи **match** конструкцію, написання програми калькулятор з постійними запитами на введення нових даних та операцій, написання програми тестування функцій списків таких як: extend(), append(), insert(id, val), remove(val), clear(), sort(), reverse(), copy(), написання програми тестування функцій словників таких як: update(), del(), clear(), keys(), values(), items(), маючи відсортований список, написання функції пошуку позиції для вставки нового елементу в список, розширення програми калькулятор функцією запитів даних для виконання операцій від користувача, що обробляє виняткові ситуації, розширення функції ділення обробкою виняткової ситуації ділення на нуль. У ході виконання цих завдань я розвинув свої знання та навички з Python та використав ці вміння у практичних ситуаціях.