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

Звіт до Теми №1

Функції та змінні

Під час виконання практичного завдання до Теми №1 було надано варіанті 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. Відкрив Visual Studio Code
2. Прочитав завдання
3. Написав код

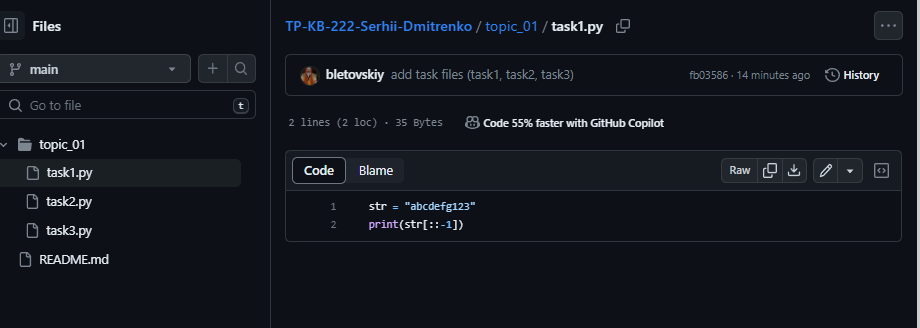
Текст програми:

|  |
| --- |
| str = "abcdefg123"  print(str[::-1]) |

Посилання на github:

<https://github.com/bletovskiy/TP-KB-222-Serhii-Dmitrenko/blob/main/topic_01/task1.py>

Знімок екрану з посилання на github:



**Форматування тексту**

Необхідно форматувати будь-який текст наступним чином:

1. Strip – прибирання пробілів
2. Capitalize – починати речення з великої букви
3. Title – кожне слово з великої букви
4. Upper, lower – змінювання реєстру на великі та маленькі букви

Хід виконання завдання:

1. Відкрив Visual Studio Code
2. Прочитав завдання
3. Написав код

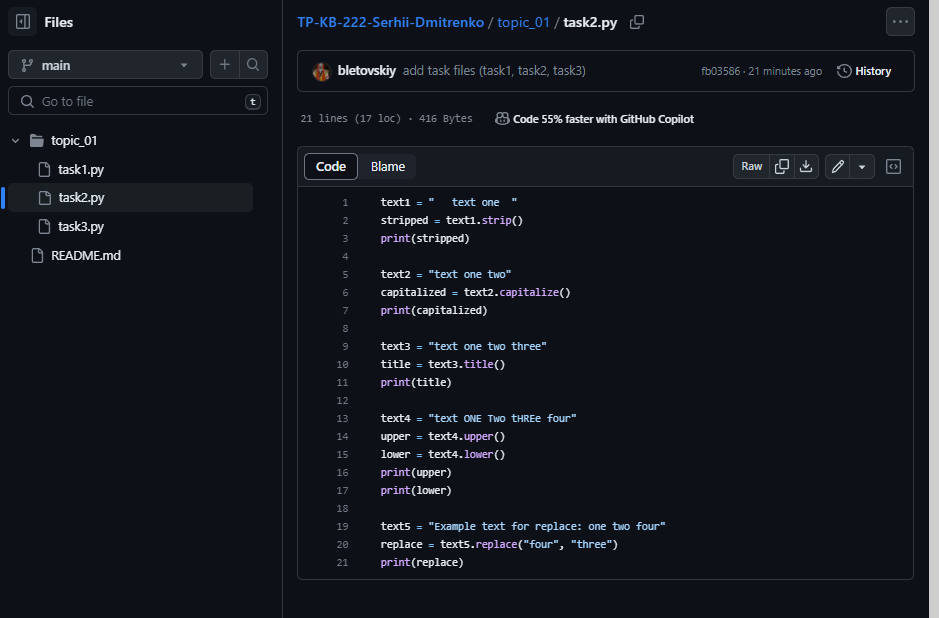
Текст програми:

|  |
| --- |
| text1 = " text one "  stripped = text1.strip()  print(stripped)  text2 = "text one two"  capitalized = text2.capitalize()  print(capitalized)  text3 = "text one two three"  title = text3.title()  print(title)  text4 = "text ONE Two tHREe four"  upper = text4.upper()  lower = text4.lower()  print(upper)  print(lower)  text5 = "Example text for replace: one two four"  replace = text5.replace("four", "three")  print(replace) |

Посилання на github:

<https://github.com/bletovskiy/TP-KB-222-Serhii-Dmitrenko/blob/main/topic_01/task2.py>

Знімок екрану з посилання на github:



**Дискримінант**

Необхідно написати функції яка вираховує функції

Хід виконання завдання:

1. Відкрив Visual Studio Code
2. Прочитав завдання
3. Написав код

Код програми:

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

Посилання на github:

<https://github.com/bletovskiy/TP-KB-222-Serhii-Dmitrenko/blob/main/topic_01/task3.py>

Знімок екрану з посилання на github:



Звіт до теми №2

Умовний перехід

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

**Дискримінант (З корнями)**

Необхідно написати код який знаходить дискримінант по формулі та знаходить його корені.

Хід виконання:

1. Прочитав завдання
2. Відкрив Visual Studio Code
3. Написав код

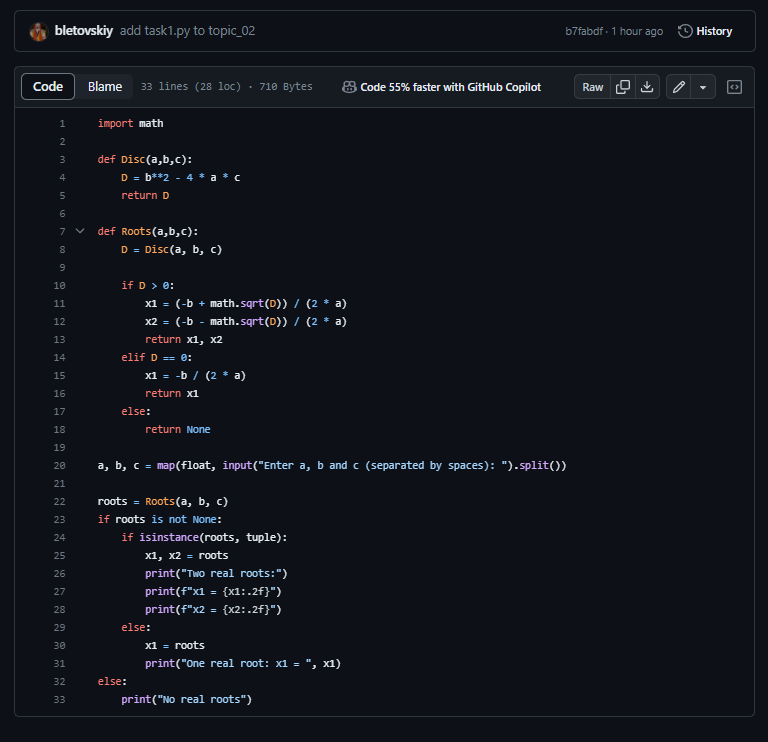
Код програми:

|  |
| --- |
| import math  def Disc(a,b,c):  D = b\*\*2 - 4 \* a \* c  return D  def Roots(a,b,c):  D = Disc(a, b, c)  if D > 0:  x1 = (-b + math.sqrt(D)) / (2 \* a)  x2 = (-b - math.sqrt(D)) / (2 \* a)  return x1, x2  elif D == 0:  x1 = -b / (2 \* a)  return x1  else:  return None    a, b, c = map(float, input("Enter a, b and c (separated by spaces): ").split())  roots = Roots(a, b, c)  if roots is not None:  if isinstance(roots, tuple):  x1, x2 = roots  print("Two real roots:")  print(f"x1 = {x1:.2f}")  print(f"x2 = {x2:.2f}")  else:  x1 = roots  print("One real root: x1 = ", x1)  else:  print("No real roots") |

Посилання на github:

<https://github.com/bletovskiy/TP-KB-222-Serhii-Dmitrenko/blob/main/topic_02/task1.py>

Знімок екрану з посилання на github:



**Калькулятор (з використанням if, else та elif)**

Потрібно написати калькулятор, який додає, віднімає, ділить та множить два числа. І зробити це за допомогою if, else та elif.

Хід виконання:

1. Прочитав завдання
2. Ознайомився з if, else та elif
3. Відкрив Visual Studio Code
4. Написав код

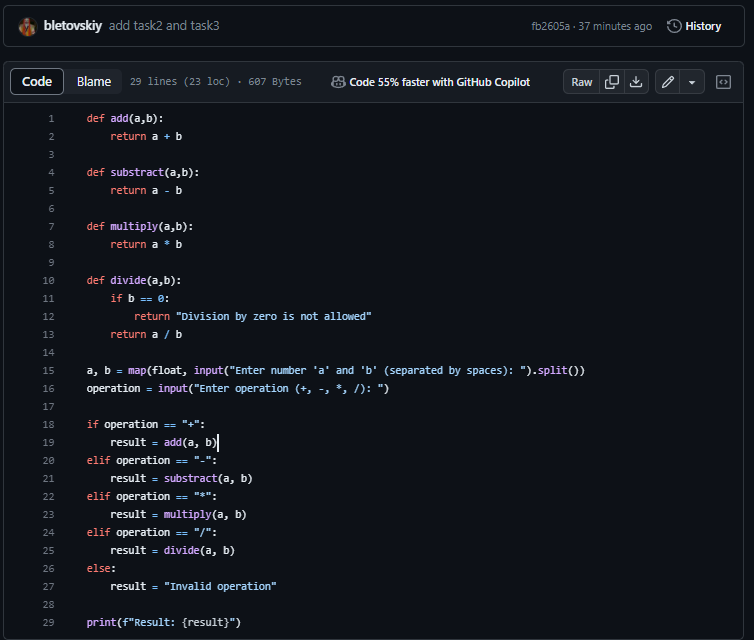
Код програми:

|  |
| --- |
| def add(a,b):  return a + b  def substract(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  a, b = map(float, input("Enter number 'a' and 'b' (separated by spaces): ").split())  operation = input("Enter operation (+, -, \*, /): ")  if operation == "+":  result = add(a, b)  elif operation == "-":  result = substract(a, b)  elif operation == "\*":  result = multiply(a, b)  elif operation == "/":  result = divide(a, b)  else:  result = "Invalid operation"  print(f"Result: {result}") |

Посилання на github:

<https://github.com/bletovskiy/TP-KB-222-Serhii-Dmitrenko/blob/main/topic_02/task2.py>

Знімок екрану з посилання на github:



Калькулятор (З використанням match)

Потрібно написати калькулятор але без if, else, elif, замість них потрібно використати match

Хід виконання:

1. Прочитав завдання
2. Ознайомився з match
3. Відкрив Visual Studio Code
4. Написав код

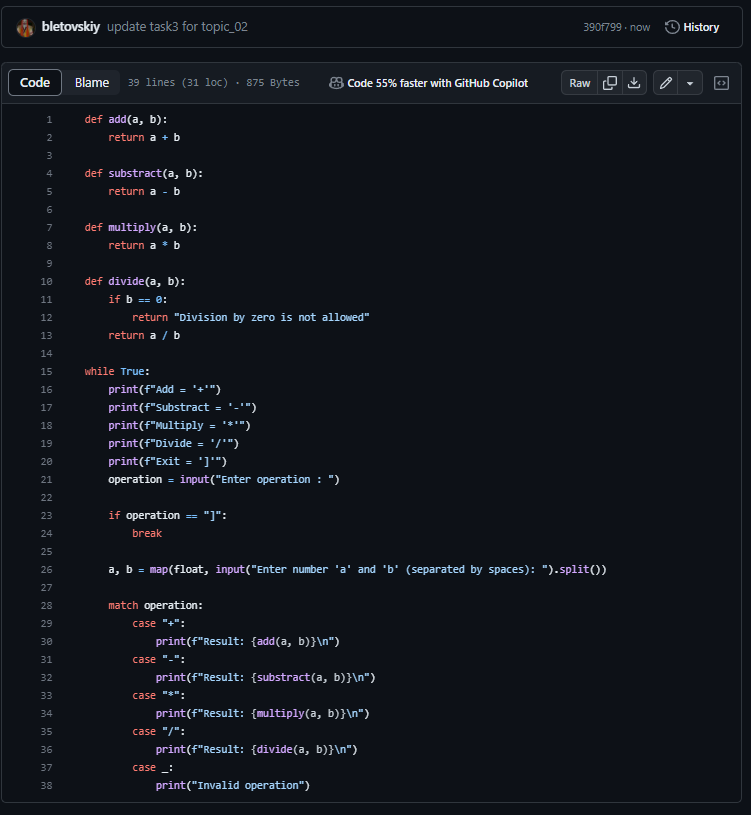
Код програми:

|  |
| --- |
| def add(a, b):  return a + b  def substract(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(f"Add = '+'")  print(f"Substract = '-'")  print(f"Multiply = '\*'")  print(f"Divide = '/'")  print(f"Exit = ']'")  operation = input("Enter operation : ")  if operation == "]":  break  a, b = map(float, input("Enter number 'a' and 'b' (separated by spaces): ").split())    match operation:  case "+":  print(f"Result: {add(a, b)}\n")  case "-":  print(f"Result: {substract(a, b)}\n")  case "\*":  print(f"Result: {multiply(a, b)}\n")  case "/":  print(f"Result: {divide(a, b)}\n")  case \_:  print("Invalid operation") |

Посилання на github:

<https://github.com/bletovskiy/TP-KB-222-Serhii-Dmitrenko/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 варіанті рішення до наступних задач:

Калькулятор (нескінченний)

Зробити калькулятор використовуючи цикл WHILE

Хід роботи:

1. Ознайомився з циклом WHILE
2. Відкрив Visual Studio Code
3. Написав код

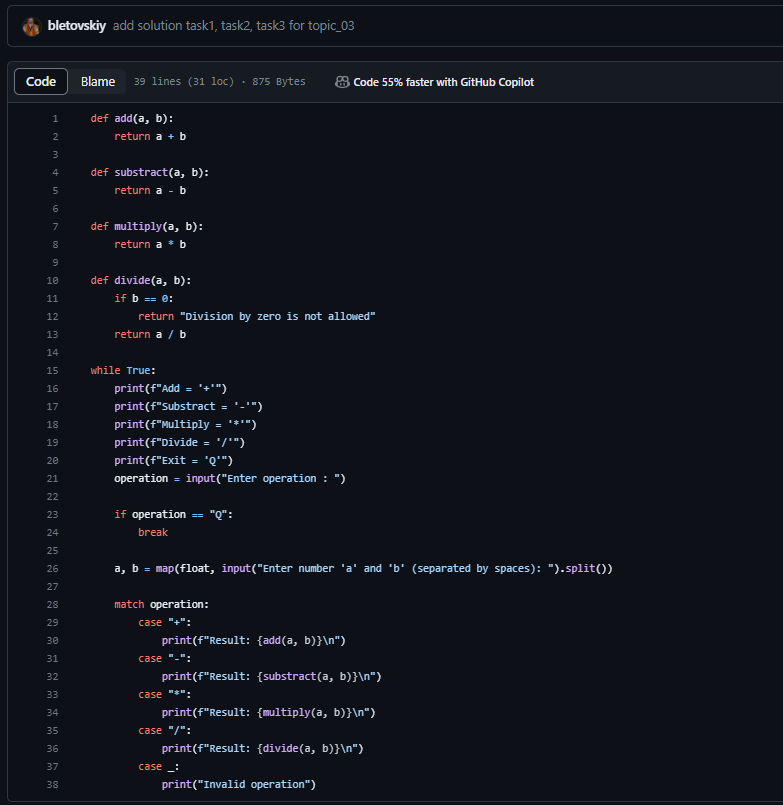
Код програми:

|  |
| --- |
| def add(a, b):  return a + b  def substract(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(f"Add = '+'")  print(f"Substract = '-'")  print(f"Multiply = '\*'")  print(f"Divide = '/'")  print(f"Exit = 'Q'")  operation = input("Enter operation : ")  if operation == "Q":  break  a, b = map(float, input("Enter number 'a' and 'b' (separated by spaces): ").split())    match operation:  case "+":  print(f"Result: {add(a, b)}\n")  case "-":  print(f"Result: {substract(a, b)}\n")  case "\*":  print(f"Result: {multiply(a, b)}\n")  case "/":  print(f"Result: {divide(a, b)}\n")  case \_:  print("Invalid operation") |

Посилання на github:

<https://github.com/bletovskiy/TP-KB-222-Serhii-Dmitrenko/blob/main/topic_03/task1.py>

Знімок екрану з посилання на github:



Ознайомлення з командами №1

Ознайомлення з командами – extend(), append(), insert(), remove(), clear(), sort(), reverse(), copy(),

Хід роботи:

1. Ознайомився з командами за допомогою інтернету
2. Написав приклади команд у Visual Studio Code

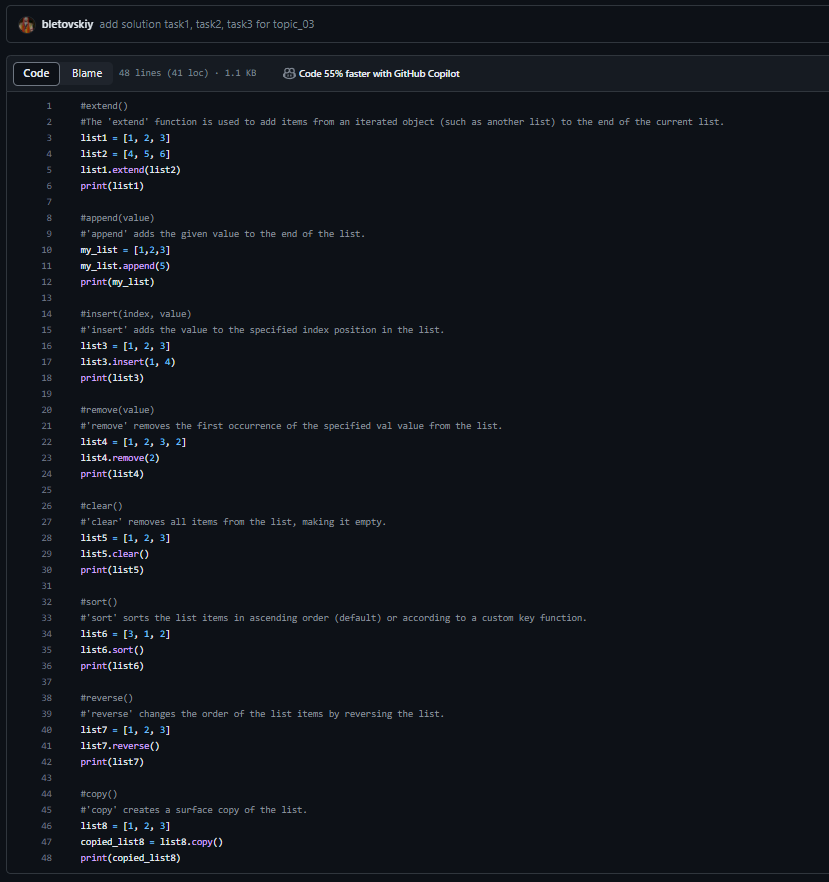
Код програми:

|  |
| --- |
| #extend()  #The 'extend' function is used to add items from an iterated object (such as another list) to the end of the current list.  list1 = [1, 2, 3]  list2 = [4, 5, 6]  list1.extend(list2)  print(list1)  #append(value)  #'append' adds the given value to the end of the list.  my\_list = [1,2,3]  my\_list.append(5)  print(my\_list)  #insert(index, value)  #'insert' adds the value to the specified index position in the list.  list3 = [1, 2, 3]  list3.insert(1, 4)  print(list3)  #remove(value)  #'remove' removes the first occurrence of the specified val value from the list.  list4 = [1, 2, 3, 2]  list4.remove(2)  print(list4)  #clear()  #'clear' removes all items from the list, making it empty.  list5 = [1, 2, 3]  list5.clear()  print(list5)  #sort()  #'sort' sorts the list items in ascending order (default) or according to a custom key function.  list6 = [3, 1, 2]  list6.sort()  print(list6)  #reverse()  #'reverse' changes the order of the list items by reversing the list.  list7 = [1, 2, 3]  list7.reverse()  print(list7)  #copy()  #'copy' creates a surface copy of the list.  list8 = [1, 2, 3]  copied\_list8 = list8.copy()  print(copied\_list8) |

Посилання на github:

<https://github.com/bletovskiy/TP-KB-222-Serhii-Dmitrenko/blob/main/topic_03/task2.py>

Знімок екрану з посилання на github:



Ознайомлення з командами №2

Ознайомлення з командами – update(), del, clear(), keys(), values(), items()

Хід роботи:

1. Ознайомився з командами за допомогою інтернету
2. Написав приклади команд у Visual Studio Code

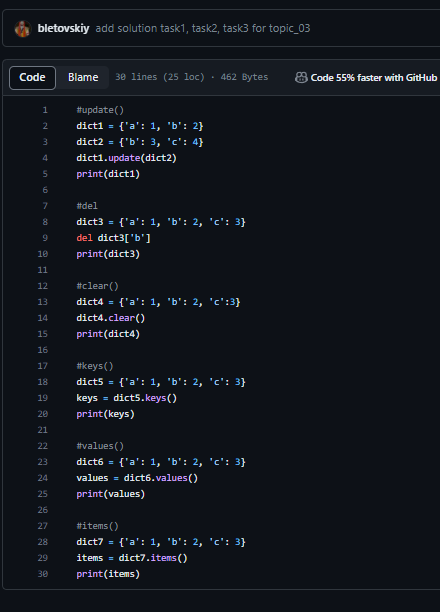
Код програми:

|  |
| --- |
| #update()  dict1 = {'a': 1, 'b': 2}  dict2 = {'b': 3, 'c': 4}  dict1.update(dict2)  print(dict1)  #del  dict3 = {'a': 1, 'b': 2, 'c': 3}  del dict3['b']  print(dict3)  #clear()  dict4 = {'a': 1, 'b': 2, 'c':3}  dict4.clear()  print(dict4)  #keys()  dict5 = {'a': 1, 'b': 2, 'c': 3}  keys = dict5.keys()  print(keys)  #values()  dict6 = {'a': 1, 'b': 2, 'c': 3}  values = dict6.values()  print(values)  #items()  dict7 = {'a': 1, 'b': 2, 'c': 3}  items = dict7.items()  print(items) |

Посилання на github:

<https://github.com/bletovskiy/TP-KB-222-Serhii-Dmtrenko/blob/main/topic_03/task3.py>

Знімок екрану з посилання на github:



Сортування

Написати програму яка дозволяє користувачеві вводити рядки та автоматично вставляти їх у впорядкований список за зростанням довжини рядків.

Хід роботи:

1. Прочитав завдання
2. Відкрив Visual Studio Code
3. Написав код

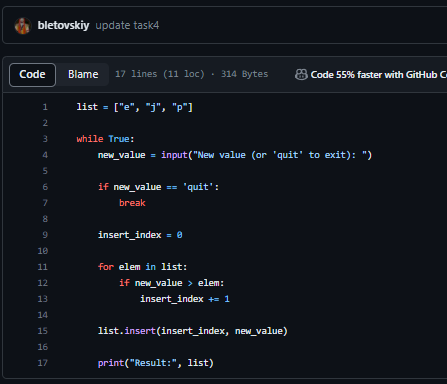
Код програми:

|  |
| --- |
| list = ["e", "j", "p"]  while True:  new\_value = input("New value (or 'quit' to exit): ")    if new\_value == 'quit':  break  insert\_index = 0  for elem in list:  if new\_value > elem:  insert\_index += 1  list.insert(insert\_index, new\_value)  print("Result:", list) |

Посилання на github:

<https://github.com/bletovskiy/TP-KB-222-Serhii-Dmitrenko/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 варіанті рішення до наступних задач:

Калькулятор

Завдання полягає в тому, щоб зробити функцію “getInt” та виняткові ситуації по типу “ZeroDevisionError” та спробувати дослідити і додати інші виняткові арифметичні та не арифметичні виняткові ситуації.

Хід роботи

1. Прочитав умови виконання завдання.
2. Дослідив сайт з винятковими ситуаціями та ознайомився з ними.
3. Відкрив Visual Studio Code.
4. Написав рішення.

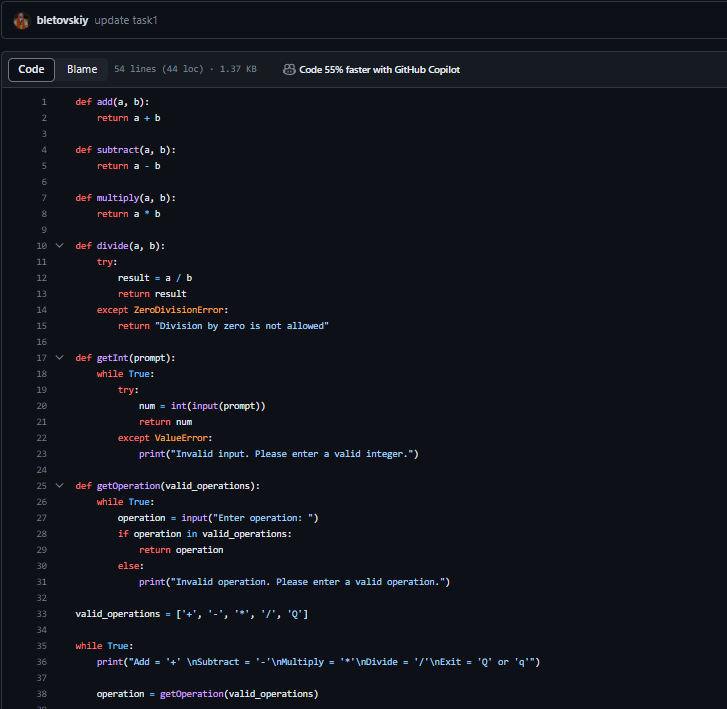
Код програми:

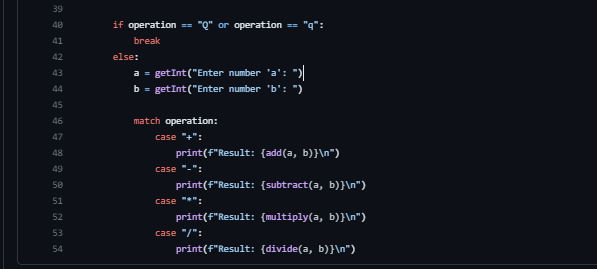
|  |
| --- |
| def add(a, b):  return a + b  def subtract(a, b):  return a - b  def multiply(a, b):  return a \* b  def divide(a, b):  try:  result = a / b  return result  except ZeroDivisionError:  return "Division by zero is not allowed"    def getInt(prompt):  while True:  try:  num = int(input(prompt))  return num  except ValueError:  print("Invalid input. Please enter a valid integer.")  def getOperation(valid\_operations):  while True:  operation = input("Enter operation: ")  if operation in valid\_operations:  return operation  else:  print("Invalid operation. Please enter a valid operation.")  valid\_operations = ['+', '-', '\*', '/', 'Q']  while True:  print("Add = '+' \nSubtract = '-'\nMultiply = '\*'\nDivide = '/'\nExit = 'Q' or 'q'")    operation = getOperation(valid\_operations)  if operation == "Q" or operation == "q":  break  else:  a = getInt("Enter number 'a': ")  b = getInt("Enter number 'b': ")    match operation:  case "+":  print(f"Result: {add(a, b)}\n")  case "-":  print(f"Result: {subtract(a, b)}\n")  case "\*":  print(f"Result: {multiply(a, b)}\n")  case "/":  print(f"Result: {divide(a, b)}\n") |

Посилання на github:

<https://github.com/bletovskiy/TP-KB-222-Serhii-Dmitrenko/blob/main/topic_04/task1.py>

Знімок екрану з посилання на github:





Звіт до теми №5

Бібліотеки

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

Гра “Камінь, ножиці, папір”

Написати гру “Камінь, ножиці, папір”

Хід роботи

1. Ознайомився з завданням
2. Відкрив Visual Studio Code
3. Написав код

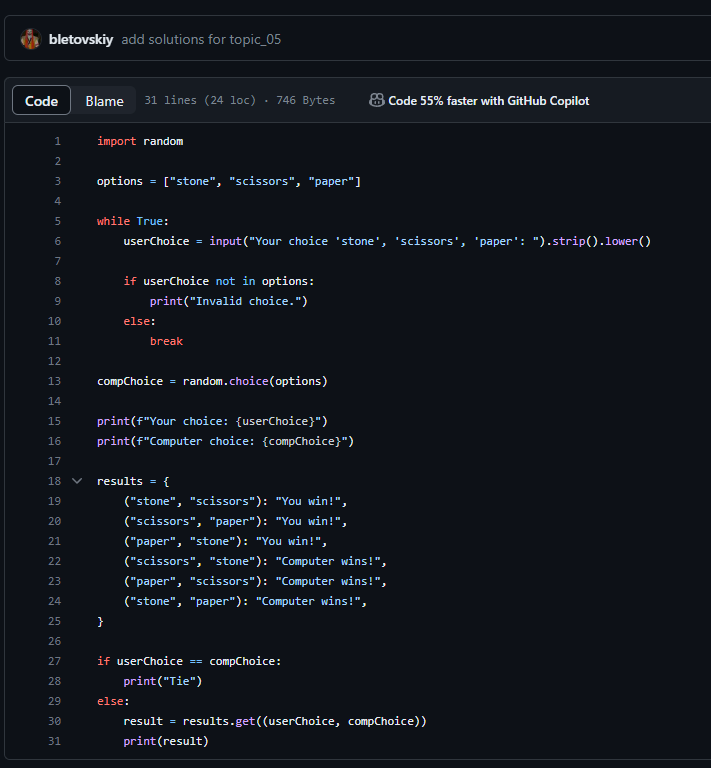
Код програми:

|  |
| --- |
| import random  options = ["stone", "scissors", "paper"]  while True:  userChoice = input("Your choice 'stone', 'scissors', 'paper': ").strip().lower()    if userChoice not in options:  print("Invalid choice.")  else:  break  compChoice = random.choice(options)  print(f"Your choice: {userChoice}")  print(f"Computer choice: {compChoice}")  results = {  ("stone", "scissors"): "You win!",  ("scissors", "paper"): "You win!",  ("paper", "stone"): "You win!",  ("scissors", "stone"): "Computer wins!",  ("paper", "scissors"): "Computer wins!",  ("stone", "paper"): "Computer wins!",  }  if userChoice == compChoice:  print("Tie")  else:  result = results.get((userChoice, compChoice))  print(result) |

Посилання на GitHub:

<https://github.com/bletovskiy/TP-KB-222-Serhii-Dmitrenko/blob/main/topic_05/task1.py>

Знімок екрану з посилання на GitHub:



Робота з API Національного банку України

Написати програму, яка буде давати вибір з трьох курсів валют (USD, EUR, PLN) та конвертувати їх у гривні (UAH)

Хід роботи

1. Ознайомився з інструкцією по API
2. Відкрив Visual Studio Code
3. Написав код

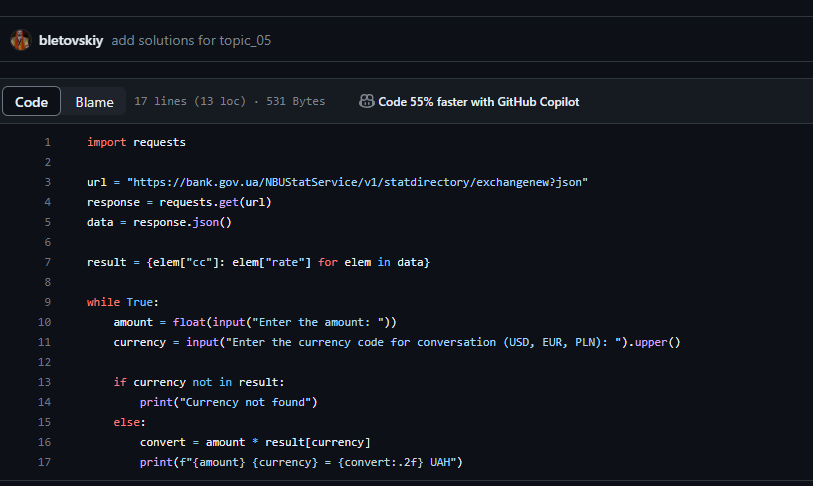
Код програми:

|  |
| --- |
| import requests  url = "https://bank.gov.ua/NBUStatService/v1/statdirectory/exchangenew?json"  response = requests.get(url)  data = response.json()  result = {elem["cc"]: elem["rate"] for elem in data}  while True:  amount = float(input("Enter the amount: "))  currency = input("Enter the currency code for conversation (USD, EUR, PLN): ").upper()  if currency not in result:  print("Currency not found")  else:  convert = amount \* result[currency]  print(f"{amount} {currency} = {convert:.2f} UAH") |

Посилання на GitHub:

<https://github.com/bletovskiy/TP-KB-222-Serhii-Dmitrenko/blob/main/topic_05/task2.py>

Знімок екрану з посилання на GitHub:



Калькулятор з відокремленими файлами для функцій

Потрібно створити файл functions.py та operations.py перенести усі функції до цих файлів і імпортувати їх до головного файл

Хід роботи

1. Ознайомився з функцією **import**
2. Відкрив Visual Studio Code
3. Написав код

Код програми(main):

|  |
| --- |
| from functions import \*  from operations import \*  while True:  print("Add = '+' \nSubtract = '-'\nMultiply = '\*'\nDivide = '/'\nExit = 'Q' or 'q'")    operation = getOperation(valid\_operations)  if operation == "Q" or operation == "q":  break  else:  a = getInt("Enter number 'a': ")  b = getInt("Enter number 'b': ")    match operation:  case "+":  print(f"Result: {add(a, b)}\n")  case "-":  print(f"Result: {subtract(a, b)}\n")  case "\*":  print(f"Result: {multiply(a, b)}\n")  case "/":  print(f"Result: {divide(a, b)}\n") |

Код програми(functions):

|  |
| --- |
| 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:  result = a / b  return result  except ZeroDivisionError:  return "Division by zero is not allowed" |

Код програми(operations):

|  |
| --- |
| def getInt(prompt):  while True:  try:  num = int(input(prompt))  return num  except ValueError:  print("Invalid input. Please enter a valid integer.")  def getOperation(valid\_operations):  while True:  operation = input("Enter operation: ")  if operation in valid\_operations:  return operation  else:  print("Invalid operation. Please enter a valid operation.")  valid\_operations = ['+', '-', '\*', '/', 'Q', 'q'] |

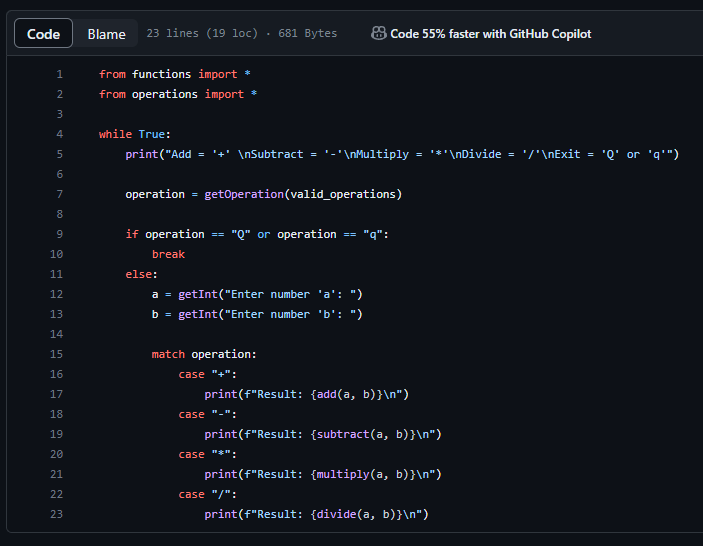
Посилання на GitHub:

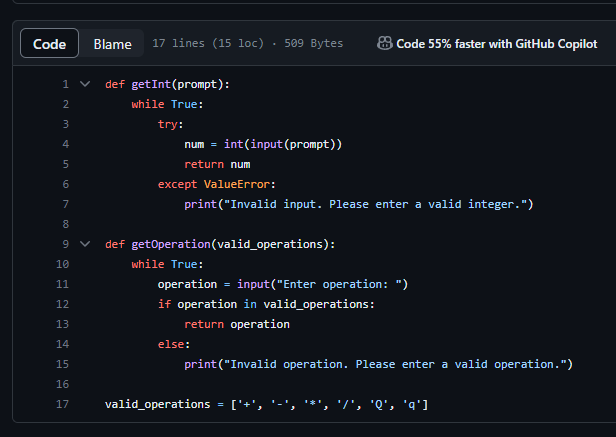
<https://github.com/bletovskiy/TP-KB-222-Serhii-Dmitrenko/blob/main/topic_05/task3.py>

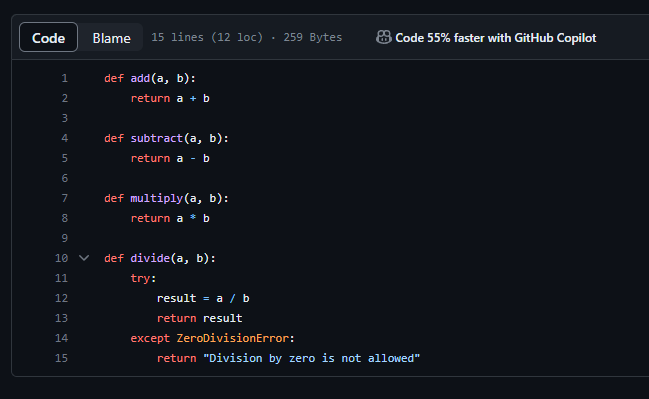
<https://github.com/bletovskiy/TP-KB-222-Serhii-Dmitrenko/blob/main/topic_05/functions.py>

<https://github.com/bletovskiy/TP-KB-222-Serhii-Dmitrenko/blob/main/topic_05/operations.py>

Знімок екрану з посилання на GitHub:







Звіт до Теми №6

Робота з файлами

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

Калькулятор  
Зробити калькулятор котрий буде робити лог файл з всіма операціями і помилками.

Хід роботи

1. Прочитав завдання
2. Відкрив Visual Studio Code
3. Написав код

Код програми(main):

|  |
| --- |
| from functions import \*  from operations import \*  while True:  print("Add = '+' \nSubtract = '-'\nMultiply = '\*'\nDivide = '/'\nExit = 'Q' or 'q'")    operation = getOperation(valid\_operations)  if operation == "Q" or operation == "q":  logToFile(operation, "Null", "Null", "Exit")  break  else:  a = getInt("Enter number 'a': ")  b = getInt("Enter number 'b': ")    match operation:  case "+":  result = add(a, b)  print(f"Result: {result}\n")  case "-":  result = subtract(a, b)  print(f"Result: {result}\n")  case "\*":  result = multiply(a, b)  print(f"Result: {result}\n")  case "/":  result = divide(a, b)  print(f"Result: {result}\n")    logToFile(operation, a, b, result) |

Код програми(functions):

|  |
| --- |
| import datetime  import os  logFile = "log.txt"  dir = os.path.dirname(os.path.abspath(\_\_file\_\_))  path = os.path.join(dir, "log.txt")  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:  result = a / b  return result  except ZeroDivisionError:    return "Division by zero is not allowed"    def logToFile(operation, a, b, result):  date = datetime.datetime.now()  formatDate = date.strftime("%Y-%m-%d %H:%M:%S")  logEntry = f"{formatDate}: Operation: {operation}, a: {a}, b: {b}, Result: {result}"  with open(path, "a") as file:  file.write(logEntry + "\n") |

Код програми(operations):

|  |
| --- |
| from functions import \*  def getInt(prompt):  while True:  try:  num = int(input(prompt))  return num  except ValueError:  print("Invalid input. Please enter a valid integer.")  logToFile("Invalid input", "", "", "Invalid input. Please enter a valid integer.")  def getOperation(valid\_operations):  while True:  operation = input("Enter operation: ")  if operation in valid\_operations:  return operation  else:  print("Invalid operation. Please enter a valid operation.")  logToFile("Invalid operation", "Null", "Null", "Invalid input")  valid\_operations = ['+', '-', '\*', '/', 'Q', 'q'] |

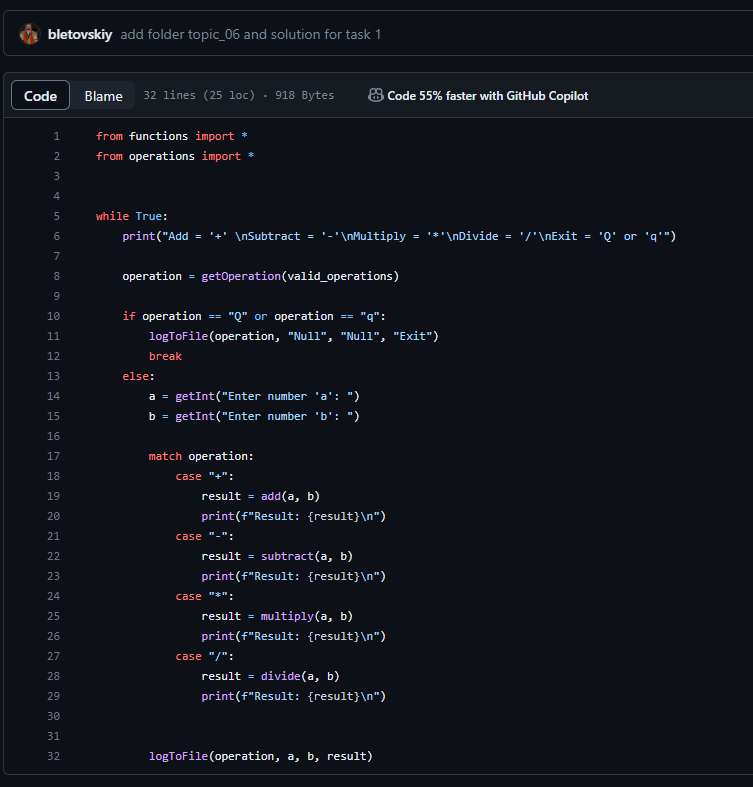
Посилання на GitHub:

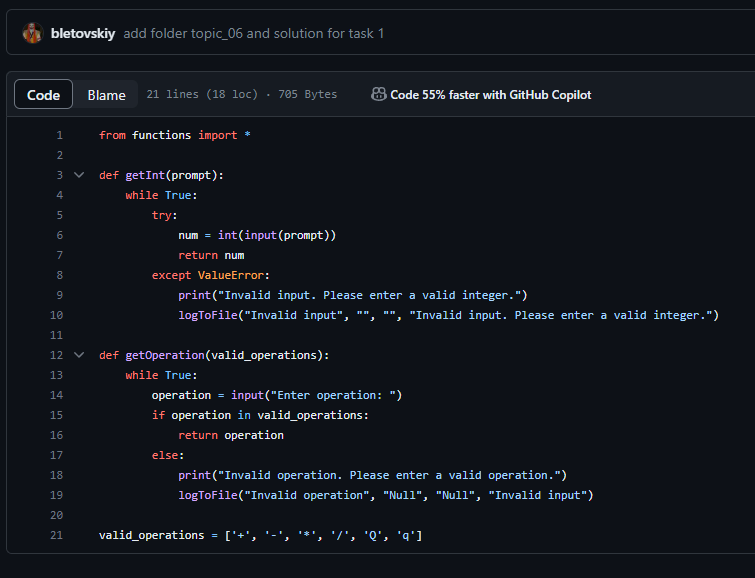
<https://github.com/bletovskiy/TP-KB-222-Serhii-Dmitrenko/blob/main/topic_06/functions.py>

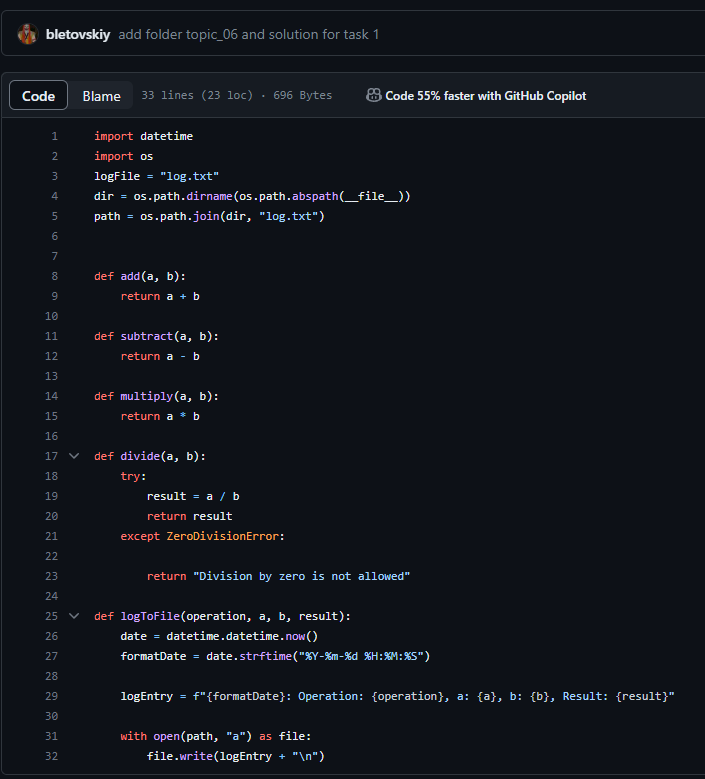
<https://github.com/bletovskiy/TP-KB-222-Serhii-Dmitrenko/blob/main/topic_06/operations.py>

<https://github.com/bletovskiy/TP-KB-222-Serhii-Dmitrenko/blob/main/topic_06/task1.py>

Знімок екрану з посилання на GitHub:







Робота з списком

Використовуючи файл student.txt відсортуйте список за ім’ям та оцінками

Хід роботи

1. Ознайомився з завданням
2. Відкрив Visual Studio Code
3. Написав код

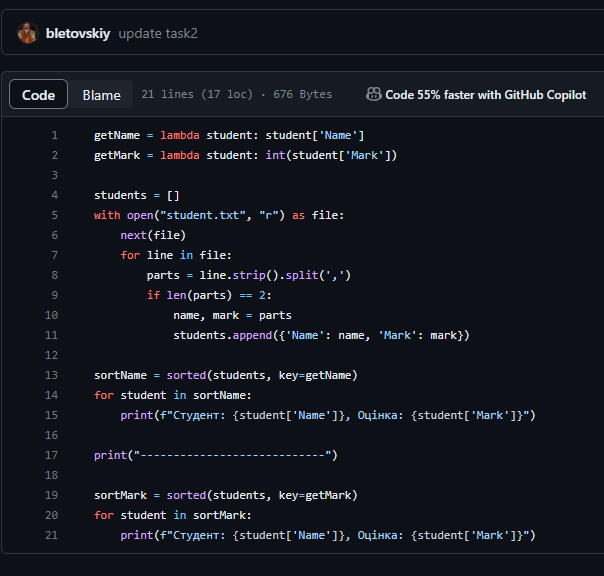
Код програми:

|  |
| --- |
| getName = lambda student: student['Name']  getMark = lambda student: int(student['Mark'])  students = []  with open("student.txt", "r") as file:  next(file)  for line in file:  parts = line.strip().split(',')  if len(parts) == 2:  name, mark = parts  students.append({'Name': name, 'Mark': mark})    sortName = sorted(students, key=getName)  for student in sortName:  print(f"Студент: {student['Name']}, Оцінка: {student['Mark']}")  print("----------------------------")  sortMark = sorted(students, key=getMark)  for student in sortMark:  print(f"Студент: {student['Name']}, Оцінка: {student['Mark']}") |

Посилання на GitHub:

<https://github.com/bletovskiy/TP-KB-222-Serhii-Dmitrenko/blob/main/topic_06/task2.py>

Знімок екрану з посилання на GitHub:



Звіт до Теми №7

Об’єктно-орієнтоване програмування

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

Ознайомлення з класами

Використовуючи посібники та документацію мови Python, ознайомитись з класами

Хід роботи

1. Ознайомився з завданням
2. Відкрив Visual Studio Code
3. Написав код

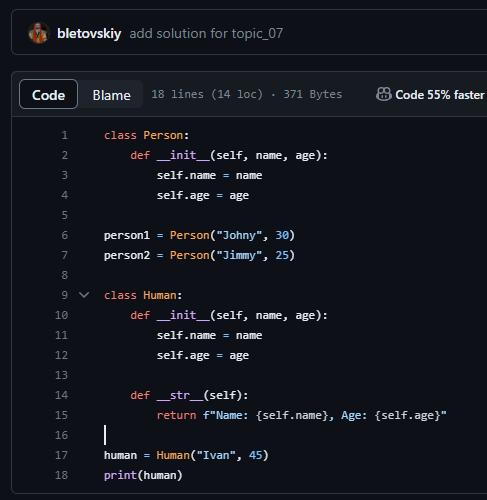
Код програми:

|  |
| --- |
| class Person:  def \_\_init\_\_(self, name, age):  self.name = name  self.age = age  person1 = Person("Johny", 30)  person2 = Person("Jimmy", 25)  class Human:  def \_\_init\_\_(self, name, age):  self.name = name  self.age = age  def \_\_str\_\_(self):  return f"Name: {self.name}, Age: {self.age}"  human = Human("Ivan", 45)  print(human) |

Посилання на GitHub:

<https://github.com/bletovskiy/TP-KB-222-Serhii-Dmitrenko/blob/main/topic_07/task1/main.py>

Знімок екрану з посилання на GitHub:



Сортування

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 роботи

1. Ознайомився з завданням
2. Відкрив Visual Studio Code
3. Написав код

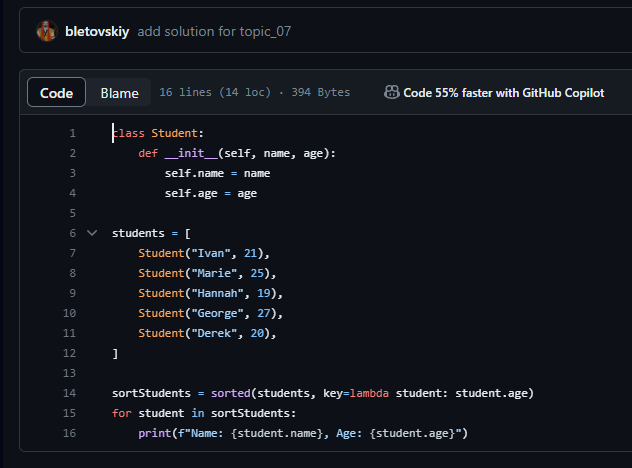
Код програми:

|  |
| --- |
| class Student:  def \_\_init\_\_(self, name, age):  self.name = name  self.age = age  students = [  Student("Ivan", 21),  Student("Marie", 25),  Student("Hannah", 19),  Student("George", 27),  Student("Derek", 20),  ]  sortStudents = sorted(students, key=lambda student: student.age)  for student in sortStudents:  print(f"Name: {student.name}, Age: {student.age}") |

Посилання на GitHub:

<https://github.com/bletovskiy/TP-KB-222-Serhii-Dmitrenko/blob/main/topic_07/task2/main.py>

Знімок екрану з посилання на GitHub:



Калькулятор

Використовуючи принципи ООП переписати програму Калькулятор.

Хід роботи

1. Ознайомився з завданням
2. Відкрив Visual Studio Code
3. Написав код

Код програми(main):

|  |
| --- |
| from operations import calculatorOperations  while True:  print("Add = '+' \nSubtract = '-'\nMultiply = '\*'\nDivide = '/'\nExit = 'Q' or 'q")    operation = input("Enter operation: ")    if operation in ["Q", "q"]:  break    calculator = calculatorOperations()  a = calculator.getInt("Enter number 'a': ")  b = calculator.getInt("Enter number 'b': ")    result = calculator.performOperation(operation, a, b)  print(f"Result: {result}\n")  calculator.logToFile(operation, a, b, result) |

Код програми(functions):

|  |
| --- |
| import datetime  import os  class calculatorFunctions:  def getInt(self, prompt):  while True:  try:  num = int(input(prompt))  return num  except ValueError:  print("Invalid input. Please enter a valid integer.")  self.logToFile("Invalid input", "", "", "Invalid input. Please enter a valid integer.")  def performOperation(self, operation, a, b):  if operation == "+":  return a + b  elif operation == "-":  return a - b  elif operation == "\*":  return a \* b  elif operation == "/":  try:  result = a / b  return result  except ZeroDivisionError:  return "Division by zero is not allowed"  def logToFile(self, operation, a, b, result):  logFile = "log.txt"  dir = os.path.dirname(os.path.abspath(\_\_file\_\_))  path = os.path.join(dir, "log.txt")  date = datetime.datetime.now()  formatDate = date.strftime("%Y-%m-%d %H:%M:%S")  logEntry = f"{formatDate}: Operation: {operation}, a: {a}, b: {b}, Result: {result}"  with open(path, "a") as file:  file.write(logEntry + "\n") |

Код програми(operations):

|  |
| --- |
| from functions import calculatorFunctions  class calculatorOperations:  def \_\_init\_\_(self):  self.valid\_operations = ['+', '-', '\*', '/', 'Q', 'q']  self.functions = calculatorFunctions()  def getInt(self, prompt):  return self.functions.getInt(prompt)  def performOperation(self, operation, a, b):  return self.functions.performOperation(operation, a, b)  def logToFile(self, operation, a, b, result):  self.functions.logToFile(operation, a, b, result) |

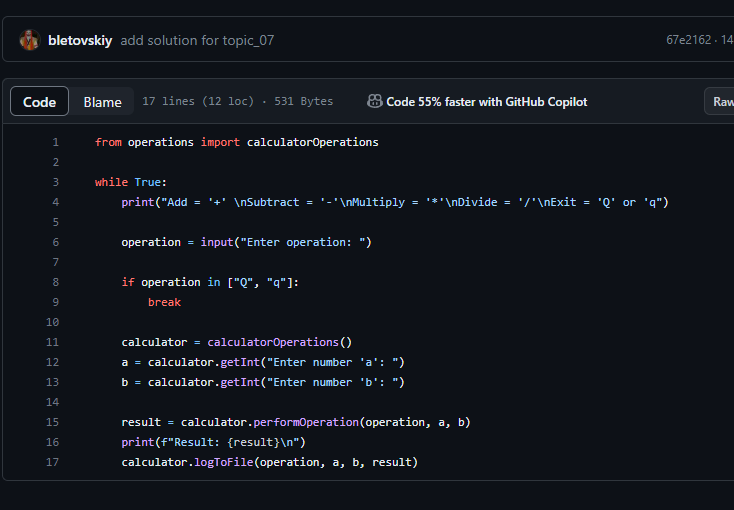
Посилання на GitHub:

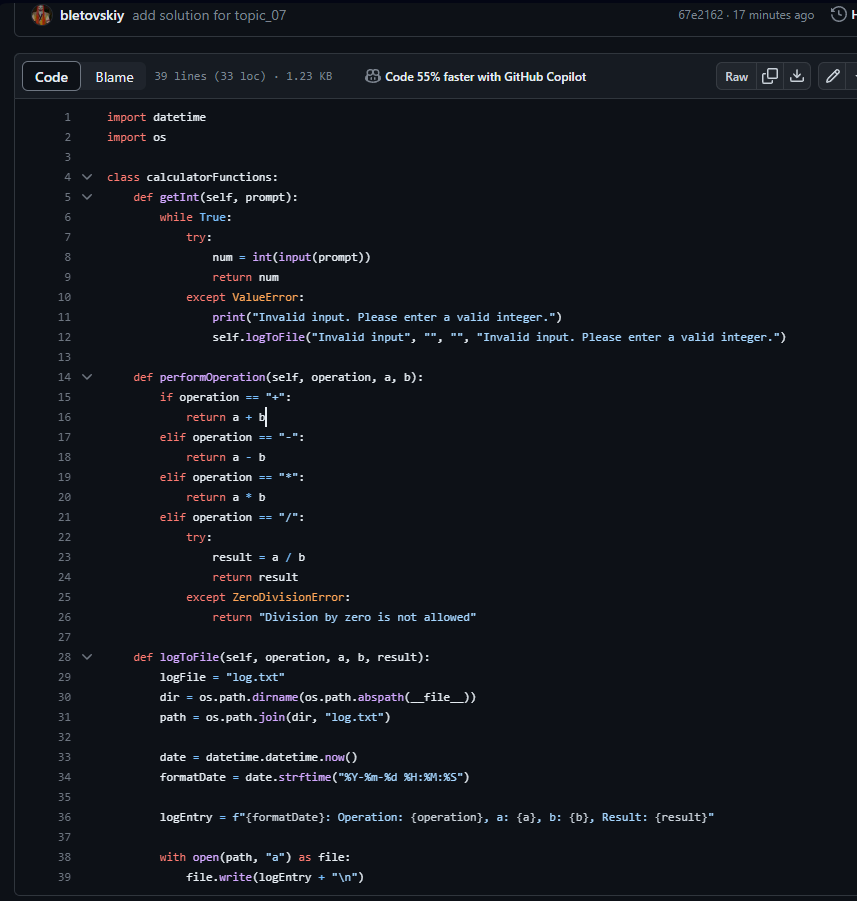
<https://github.com/bletovskiy/TP-KB-222-Serhii-Dmitrenko/blob/main/topic_07/task3/calculator.py>

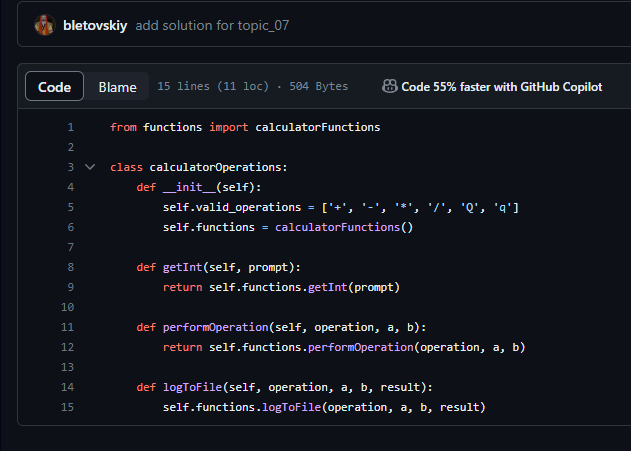
<https://github.com/bletovskiy/TP-KB-222-Serhii-Dmitrenko/blob/main/topic_07/task3/functions.py>

<https://github.com/bletovskiy/TP-KB-222-Serhii-Dmitrenko/blob/main/topic_07/task3/operations.py>

Знімок екрану з посилання на GitHub:







Висновок

Під час виконання практичних робіт, я з головою занурився у світ програмування на мові Python. За цей невеликий час вивчання цієї мови програмування, я зрозумів багато понять. Я познайомився з функціями, умовними переходами та циклами. Опанував використання бібліотек, роботу з файлами, та особливо принципи ООП.