**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: Повернення рядка в зворотному порядку, Тестування базових функцій для рядків, Написання функції пошуку Дискримінанту.

task1

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

Код програми:

str1="abcdefg123"

str2=""

for i in **range**(10):

    str2+=str1[9-i]

**print** (str2)

task2

Необхідно протестувати базові функції для рядків.

Код програми:

x = "QwErTy"

a = x.**strip**()

b = x.**capitalize**()

c = x.**title**()

d = x.**upper**()

e = x.**lower**()

**print** (a,"\n",b,"\n",c,"\n",d,"\n",e,"\n")

task3

Необхідно написати функцію пошуку дискримінанту.

Код програми:

import **math**

def **discr**(a,b,c):

    return b\*\*2 - (4\*a\*c)

a = **int** (**input**("a: "))

b = **int** (**input**("b: "))

c = **int** (**input**("c: "))

d = **discr**(a,b,c)

**print**("d= ", d)

Посилання на github: <https://github.com/kuzqz41/TP-KB-221-Kuzmenko-Bohdan/tree/main/topic_01>

Звіт до Теми №2

Функції перевірки

Під час виконання практичного завдання до Теми №2 було надано варіанти рішення до наступних задач: пошук коренів квадратного рівняння, створення калькуляторів за допомогою if, elif, else та match.

task1

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

Код програми:

import **math**

def **discr**(a,b,c):

    return b\*\*2 - (4\*a\*c)

def **roots**(a,b,c):

    d = **discr**(a,b,c)

    if d > 0:

        x1 = (-b + **math**.**sqrt**(d))/2\*a

        x2 = (-b - **math**.**sqrt**(d))/2\*a

    elif d == 0:

        x1 = (-b + **math**.**sqrt**(d))/2\*a

        x2 = "Кореня не існує"

    else:

        x1 = "Кореня не існує"

        x2 = "Кореня не існує"

    return x1, x2

**print** ("Квадратне рівняння ax^2 + bx + c = 0")

a = **int** (**input**("a: "))

b = **int** (**input**("b: "))

c = **int** (**input**("c: "))

d = **discr**(a,b,c);

**print** ("d= ",d)

x1, x2 = **roots**(a,b,c)

**print** ("x1= ",x1,"      x2= ",x2)

task2

Написати програму калькулятор використовуючи **if else** конструкцію

Код програми:

def **sum**(a,b):

    return a + b

def **minus**(a,b):

    return a - b

def **mult**(a,b):

    return a \* b

def **div**(a,b):

    return a / b

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

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

x = **str**(**input**("Введіть знак (+ або - або \* або /): "))

if x=="+":

**print** ("a + b = ", **sum**(a,b))

elif x=="-":

**print** ("a - b = ", **minus**(a,b))

elif x=="\*":

**print** ("a \* b = ", **mult**(a,b))

elif x=="/":

    if b==0:

**print** ("Ділити на нуль не можна")

    else:

**print** ("a / b = ", **div**(a,b))

else:

**print** ("Ви ввели неправильний знак")

task3

Написати програму калькулятор використовуючи **match** конструкцію

Код програми:

def **sum**(a,b):

    return a + b

def **minus**(a,b):

    return a - b

def **mult**(a,b):

    return a \* b

def **div**(a,b):

    return a / b

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

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

x = **str**(**input**("Введіть знак (+ або - або \* або /): "))

match x:

    case "+":

**print** ("a + b = ",**sum**(a,b))

    case "-":

**print** ("a - b = ",**minus**(a,b))

    case "\*":

**print** ("a \* b = ",**mult**(a,b))

    case "/":

        match b:

            case 0:

**print** ("Ділити на нуль не можна")

            case \_:

**print** ("a / b = ",**div**(a,b))

    case \_:

**print** ("Ви ввели неправильний знак")

Посилання на github: <https://github.com/kuzqz41/TP-KB-221-Kuzmenko-Bohdan/tree/main/topic_02>

Звіт до Теми №3

Цикли

Під час виконання практичного завдання до Теми №3 було надано варіанти рішення до наступних задач: Написання калькулятора з постійними запитами на введення даних, Тестування функцій для списків, Тестування функцій для словників, Написання функції пошуку позиції елементу у відсортованому списку.

task1

Необхідно 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

Код програми:

def **sum**(a,b):

    return a + b

def **minus**(a,b):

    return a - b

def **mult**(a,b):

    return a \* b

def **div**(a,b):

    return a / b

while 1:

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

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

    x = **str**(**input**("Введіть знак (+ або - або \* або /) (exit щоб вийти): "))

    match x:

        case "+":

**print** ("a + b = ",**sum**(a,b))

        case "-":

**print** ("a - b = ",**minus**(a,b))

        case "\*":

**print** ("a \* b = ",**mult**(a,b))

        case "/":

            match b:

                case 0:

**print** ("Ділити на нуль не можна")

                case \_:

**print** ("a / b = ",**div**(a,b))

        case "exit":

            break

        case \_:

**print** ("Ви ввели неправильний знак")

task2

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

Код програми:

a=[1,2,3]

b=[6,2,5]

c=[]

d=[]

a.**extend**(b)

**print** ("extend - ",a)

c.**append**(1)

c.**append**(9)

**print** ("append - ",c)

a.**insert**(2, "qwe")

**print** ("insert - ",a)

a.**remove** (3)

**print** ("remove - ",a)

a.**clear**()

**print** ("clear - ",a)

b.**sort**()

**print**("sort - ",b)

b.**reverse**()

**print**("reverse - ",b)

d = b.**copy**()

**print**("copy - ",d)

task3

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

Код програми:

a={'qwe':123}

b={'asd':456}

c={'qwe':123,'asd':456}

a.**update**(b)

**print** ("update - ",a)

del(b)

a.**clear**()

**print**("clear - ",a)

x=c.**keys**()

**print** ("keys - ",x)

x=c.**values**()

**print** ("values - ",x)

x=c.**items**()

**print**("items - ",x)

task4

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

Код програми:

def **PosSearch**(a,b):

    for i in **range**(**len**(a)):

        if a[i] == b:

            return i

a = ["aa", "bb", "cc", "dd", "ee", "ff", "gg", "hh", "ii", "jj", "kk", "ll", "mm", "nn", "oo", "pp", "qq", "rr", "ss", "tt", "uu", "vv", "ww", "xx", "yy", "zz"]

b = **str**(**input**("b: "))

a.**append**(b)

a.**sort**()

**print** (a)

i = **PosSearch** (a,b)

**print** ("Елемент ", b," знаходиться на ",i," позиції")

Посилання на github: <https://github.com/kuzqz41/TP-KB-221-Kuzmenko-Bohdan/tree/main/topic_03>

Звіт до Теми №4

Виняткові ситуації

Під час виконання практичного завдання до Теми №4 було надано варіанти рішення до наступних задач: Розширення програми калькулятора функцією запитів даних для виконання операцій від користувача, що обробляє виняткові ситуації; Розширення функції ділення обробкою виняткової ситуації ділення но нуль; Ознайомлення зі списком виняткових ситуацій.

task1

Код програми:

def **sum**(a,b):

    return a + b

def **minus**(a,b):

    return a - b

def **mult**(a,b):

    return a \* b

def **div**(a,b):

    return a / b

def **GetInt**(promt:**str**):

    while 1:

        try:

            return **int**(**input**(promt))

        except **ValueError**:

**print** ("Неправильні дані")

while 1:

    a = **GetInt**("Введіть перше число: ")

    b = **GetInt**("Введіть друге чисол: ")

    x = **str**(**input**("Введіть знак (+ або - або \* або /) (exit щоб вийти): "))

    match x:

        case "+":

**print** ("a + b = ",**sum**(a,b))

        case "-":

**print** ("a - b = ",**minus**(a,b))

        case "\*":

**print** ("a \* b = ",**mult**(a,b))

        case "/":

**print** ("a / b = ",**div**(a,b))

        case "exit":

            break

        case \_:

**print** ("Ви ввели неправильний знак")

task2

Код програми:

def **sum**(a,b):

    return a + b

def **minus**(a,b):

    return a - b

def **mult**(a,b):

    return a \* b

def **div**(a,b):

    return a / b

def **GetInt**(promt:**str**):

    while 1:

        try:

            return **int**(**input**(promt))

        except **ValueError**:

**print** ("Неправильні дані")

while 1:

    try:

        a = **GetInt**("Введіть перше число: ")

        b = **GetInt**("Введіть друге чисол: ")

        x = **str**(**input**("Введіть знак (+ або - або \* або /) (exit щоб вийти): "))

        match x:

            case "+":

**print** ("a + b = ",**sum**(a,b))

            case "-":

**print** ("a - b = ",**minus**(a,b))

            case "\*":

**print** ("a \* b = ",**mult**(a,b))

            case "/":

**print** ("a / b = ",**div**(a,b))

            case "exit":

                break

            case \_:

**print** ("Ви ввели неправильний знак")

    except **ZeroDivisionError**:

**print** ("Ділити на нуль не можна")

Посилання на github: <https://github.com/kuzqz41/TP-KB-221-Kuzmenko-Bohdan/tree/main/topic_04>

Звіт до Теми №5

Функції та змінні

Під час виконання практичного завдання до Теми №5 було надано варіанти рішення до наступних задач: Написання гри камінь, ножиці, папір; Написання програми конвертування іноземної валюти у гривню; Використання модулів для програми калькулятор.

task1

Необхідно створити гру: камінь, ножиці, папір.

Код програми:

import **random**

def **get\_str**(promt:**str**):

    while 1:

        x = **str**(**input**(promt))

        if (x != "stone" and  x != "scissor" and x != "paper" and x != "exit"):

**print** ("Wrong")

        else:

            return x

def **rand\_str**():

    return **random**.choice(["stone", "scissor", "paper"])

def **answ**(a,b):

    if (a == "stone" and b == "scissor") or (a == "paper" and b == "stone") or (a == "scissor" and b == "paper"):

        return "You win"

    elif a == b:

        return "Draw"

    else:

        return "Random wins"

while 1:

    choice = **get\_str**('Choise stone, scissor or paper (exit to leave): ')

    if choice == "exit":

        break

    rand\_choice = **rand\_str**()

**print** ("Random choice - ", rand\_choice)

**print** (**answ**(choice, rand\_choice))

task2

Необхідно написати програму конвертування іноземної валюти у гривню.

Код програми:

import **requests**

def **get\_str**(promt:**str**):

    while 1:

        x = **str**(**input**(promt))

        if (x != "EUR" and  x != "USD" and x != "PLN" and x != "exit"):

**print** ("Wrong")

        else:

            return x

def **get\_int**(promt:**str**):

    while 1:

        try:

            return **int**(**input**(promt))

        except **ValueError**:

**print** ("Wrong")

def **rate**(a):

    response = **requests**.**get**("https://bank.gov.ua/NBUStatService/v1/statdirectory/exchange?json")

    for elem in response.**json**():

        if elem['cc'] == a:

            return elem['rate']

def **result**(a, b):

    return a\*b

while 1:

    val = **get\_str**("Choise EUR, USD or PLN (exit to leave): ")

    if val == "exit":

        break

    amount = **get\_int**("Amount: ")

**print** ("Result ", amount, " ", val, " to UAH = ", **result**(**rate**(val), amount))

task3

Необхідно використовувати модулі для програми калькулятор

Код основного файлу calc.py:

from **operations** import **GetInt**

from **operations** import **GetOp**

from **functions** import **sum**

from **functions** import **minus**

from **functions** import **mult**

from **functions** import **div**

while 1:

    try:

        a = **GetInt**("Введіть перше число: ")

        b = **GetInt**("Введіть друге число: ")

        x = **GetOp**("Введіть знак (+ або - або \* або /) (exit щоб вийти): ")

        match x:

            case "+":

**print** ("a + b = ",**sum**(a,b))

            case "-":

**print** ("a - b = ",**minus**(a,b))

            case "\*":

**print** ("a \* b = ",**mult**(a,b))

            case "/":

**print** ("a / b = ",**div**(a,b))

            case "exit":

                break

    except **ZeroDivisionError**:

**print** ("Ділити на нуль не можна")

Код модуля functions.py

def **sum**(a,b):

    return a + b

def **minus**(a,b):

    return a - b

def **mult**(a,b):

    return a \* b

def **div**(a,b):

    try:

        return a / b

    except **ZeroDivisionError**:

        return "Ділити на нуль не можна"

Код модуля operations.py

def **GetInt**(promt:**str**):

    while 1:

        try:

            return **int**(**input**(promt))

        except **ValueError**:

**print** ("Неправильні дані")

def **GetOp**(promt:**str**):

    while 1:

        x = **str**(**input**(promt))

        if (x != "+" and  x != "-" and x != "\*" and x != "/" and x != "exit"):

**print** ("Ви ввели неправильний знак")

        else:

            return x

Посилання на github: <https://github.com/kuzqz41/TP-KB-221-Kuzmenko-Bohdan/tree/main/topic_05>

Звіт до Теми №6

Робота з файлами

Під час виконання практичного завдання до Теми №6 було надано варіанти рішення до наступних задач: Розробити механізм логування всіх дій, що виконує програма калькулятор; Виконати сортування невідсортованого списку, елементами якого є словники з двома параметрами.

task1

Розробити механізм логування всіх дій, що виконує калькулятор

Код основного файлу calc.py:

from **operations** import **GetInt**

from **operations** import **GetOp**

from **functions** import **sum**

from **functions** import **minus**

from **functions** import **mult**

from **functions** import **div**

from **functions** import **reswrite**

while 1:

    try:

        a = **GetInt**("Введіть перше число: ")

        b = **GetInt**("Введіть друге число: ")

        x = **GetOp**("Введіть знак (+ або - або \* або /) (exit щоб вийти): ")

        match x:

            case "+":

**print** ("a + b = ",**sum**(a,b))

**reswrite**(a,b,x,**sum**(a,b))

            case "-":

**print** ("a - b = ",**minus**(a,b))

**reswrite**(a,b,x,**minus**(a,b))

            case "\*":

**print** ("a \* b = ",**mult**(a,b))

**reswrite**(a,b,x,**mult**(a,b))

            case "/":

**print** ("a / b = ",**div**(a,b))

**reswrite**(a,b,x,**div**(a,b))

            case "exit":

                break

    except **ZeroDivisionError**:

**print** ("Ділити на нуль не можна")

Код модуля functions.py

def **sum**(a,b):

    return a + b

def **minus**(a,b):

    return a - b

def **mult**(a,b):

    return a \* b

def **div**(a,b):

    try:

        return a / b

    except **ZeroDivisionError**:

        return "Zero Division"

def **reswrite**(a,b,c,d):

    file = **open**("topic\_06\\_results.txt", "a")

    file.**write**(f"{a} {c} {b} = {d}\n")

    file.**close**

Код модуля operations.py

def **GetInt**(promt:**str**):

    while 1:

        try:

            return **int**(**input**(promt))

        except **ValueError**:

**print** ("Неправильні дані")

def **GetOp**(promt:**str**):

    while 1:

        x = **str**(**input**(promt))

        if (x != "+" and  x != "-" and x != "\*" and x != "/" and x != "exit"):

**print** ("Ви ввели неправильний знак")

        else:

            return x

task2

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

Код програми:

students = []

with **open**("topic\_06\\_students.txt") as file:

    for line **in** file:

        name, age = line.**rstrip**().**split**(",")

        student = {"name":name, "age": age}

        students.**append**(student)

for student in **sorted**(students, key=lambda student: student["name"]):

**print**(f"Name is {student['name']} age is {student['age']}") *# braces*

Посилання на github: <https://github.com/kuzqz41/TP-KB-221-Kuzmenko-Bohdan/tree/main/topic_06>

Звіт до Теми №7

Об’єктно-орієнтоване програмування

Під час виконання практичного завдання до Теми №7 було надано варіанти рішення до наступних задач: Ознайомитися з методами класу по типу \_\_init\_\_(self) \_\_str\_\_(self); Написати цикл який виводить на екран елементи списку у відсортованому порядку; Використовуючи принципи ООП переписати калькулятор

task1

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

Код програми:

class **Student**:

    def **\_\_init\_\_** (self, name, surname, date\_of\_birth):

        self.name = name

        self.surname = surname

        self. date\_of\_birth = date\_of\_birth

    def **\_\_str\_\_**(self):

        return (f"{self.name} {self.surname} {self.date\_of\_birth}\n")

s1 = **Student**("qwe", "rty", "01.01.2005")

s2 = **Student**("asd", "fgh", "12.12.2004")

**print** (s2, s1)

task2

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

Код програми:

class **Student**:

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

        self.name = name

        self.age = age

    def **\_\_repr\_\_**(self):

        return **repr**((self.name, self.age))

students = [

**Student**("Fink", 57),

**Student**("Ihor", 37),

**Student**("Jon", 44),

**Student**("Zak", 27),

]

**print** (**sorted**(students, key=lambda student: student.age))

task 3

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

Код основного файлу calc.py

from **operations** import \*

from **functions** import \*

class **Calculator**:

    def **main**():

        a = **Get**.**GetInt**("Введіть перше число: ")

        b = **Get**.**GetInt**("Введіть друге число: ")

        x = **Get**.**GetOp**("Введіть знак (+ або - або \* або /) (exit щоб вийти): ")

        r = **Functions**(a,b)

        match x:

            case "+":

                r = r.**sum**()

            case "-":

                r = r.**minus**()

            case "\*":

                r = r.**mult**()

            case "/":

                r = r.**div**()

            case "exit":

**exit**()

**Operations**.**res**(a,b,x,r)

while 1:

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

**Calculator**.**main**()

Код модуля functions.py

class **Functions**:

    def **\_\_init\_\_**(self, a, b):

        self.a = a

        self.b = b

    def **sum**(self):

        return self.a + self.b

    def **minus**(self):

        return self.a - self.b

    def **mult**(self):

        return self.a \* self.b

    def **div**(self):

        try:

            return self.a / self.b

        except **ZeroDivisionError**:

            return "Zero Division"

class **Result**:

    def **reswr**(a,b,c,d):

        file = **open**("topic\_07\\_results.txt", "a")

        file.**write**(f"{a} {c} {b} = {d}\n")

        file.**close**

Код модуля operations.py

from **functions** import \*

class **Get**:

    def **GetInt**(promt):

        while 1:

            try:

                return **int**(**input**(promt))

            except **ValueError**:

**print** ("Неправильні дані")

    def **GetOp**(promt):

        while 1:

            x = **str**(**input**(promt))

            if (x != "+" and  x != "-" and x != "\*" and x != "/" and x != "exit"):

**print** ("Ви ввели неправильний знак")

            else:

                return x

class **Operations**:

    def **res**(a,b,c,d):

**print** (f"{a} {c} {b} = {d}")

**Result**.**reswr**(a,b,c,d)

Посилання на github: <https://github.com/kuzqz41/TP-KB-221-Kuzmenko-Bohdan/tree/main/topic_07>