МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М. Є. Жуковського

«Харківський авіаційний інститут»

Кафедра систем управління літальними апаратами

Лабораторна робота № 1

з дисципліни «Об'єктно-орієнтоване програмування СУ»

Тема: "Розробка програм для математичних обчислень в Python"

ХАІ.301 . Авіоніка . група 3-92АВ(і). ЛР1

Виконав студент гр. \_\_\_\_\_\_3-92АВ(і)\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Сидоренко В.В\_\_\_

(підпис, дата) (П.І.Б.)

Перевірив

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ к.т.н., доц. О. В. Гавриленко

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ас.  В. О. Білозерський

(підпис, дата) (П.І.Б.)

2023

# МЕТА РОБОТИ

Ознайомитися з основами розробки інтерпретованих програм і реалізувати скрипт для введення / виведення даних і виконання найпростіших

математичних обчислень на мові програмування Python.

# ПОСТАНОВКА ЗАДАЧІ

Завдання 1. Вирішити завдання з цілочисельними змінними. Всі вхідні і

вихідні дані в задачах цієї групи є цілими числами. Всі числа, для яких вказано

кількість цифр (двозначне число, тризначне число та ін.), вважаються

додатними. Завдання представлено в табл.1.(Варіант 11)

Завдання 2. Обчислити математичне вираз зі змінними дійсного типу,

використовуючи стандартну бібліотеку math. Вирази представлено в табл.2.(Варіант 15)

Завдання 3. Вирішити завдання з логічними змінними. У всіх завданнях

даної групи потрібно вивести логічне значення True, якщо наведене

висловлювання для запропонованих вихідних даних є істинним, і значення

False в іншому випадку. Всі числа, для яких вказано кількість цифр (двозначне

число, тризначне число і т. Д.), Вважаються цілими додатними. Завдання

представлено в табл.3.(Варіант 31)

*Всі завдання повинні бути реалізовані в одному скрипті у вигляді окремих*

*функцій без параметрів зі строкою документації. Запустити функції з*

*командного вікна.*

*При введенні даних і обчисленнях необхідно передбачити перевірку даних і*

*обробку виняткових ситуацій.*

*Код повинен містити коментарі!*

# ВИКОНАННЯ РОБОТИ

Завдання 1. Integer11

Вирішення задачі:

Вхідні дані (ім’я, опис, тип, обмеження):

number (int): Вхідне тризначне число. Мінімальне значення -2 147 483 648, максимальне значення 2 147 483 647

Вихідні дані (ім’я, опис, тип):

digit\_sum (int): Сума цифр числа (number). Мінімальне значення -2 147 483 648, максимальне значення 2 147 483 647

digit\_product (int): Добуток цифр числа (number). Мінімальне значення -2 147 483 648, максимальне значення 2 147 483 647

Лістинг коду вирішення задачі наведено в дод. А (стор. 5). Екран роботи програми показаний на рис. Б.1.

Завдання 2. Var15

Вирішення задачі:

Вхідні дані (ім’я, опис, тип, обмеження):

x (float): Числове значення, для якого обчислюється математичний вираз.

Тип даних з плаваючою точкою і точністю від 6 до 9 цифр

Вихідні дані (ім’я, опис, тип):

y (float): Результат математичного виразу для заданого значення x.

Тип даних з плаваючою точкою і точністю від 6 до 9 цифр

Ця функція обчислює математичний вираз за заданою формулою для введеного значення x. Якщо виникає помилка під час обчислення (наприклад, ділення на нуль), то вона генерує виняток з повідомленням про помилку.

Лістинг коду вирішення задачі наведено в дод. А (стор. 6). Екран роботи програми показаний на рис. Б.2.

Завдання 3. Boolean31

Вирішення задачі:

Вхідні дані (ім’я, опис, тип, обмеження):

a (float): Довжина сторони а.

b (float): Довжина сторони b.

c (float): Довжина сторони c.

Тип даних з плаваючою точкою і точністю від 6 до 9 цифр.

Вихідні дані (ім’я, опис, тип):

bool: True, якщо трикутник рівнобедрений, False в іншому випадку.

bool представляє собою логічне значення

Лістинг коду вирішення задачі наведено в дод. А (стор. 7). Екран роботи програми показаний на рис. Б.3.

# ВИСНОВКИ

Було вивчено та закріплено на практиці використання вбудованих функцій та виключень у мові програмування Python. У результаті написаного коду були отримані навички введення, обробки та виведення даних у консолі. Вивчено принципи організації коду за допомогою функцій та обробки винятків для покращення його структури та забезпечення коректної роботи у різних ситуаціях.

ДОДАТОК А

Лістинг коду програми до задач 1-3>

Функція 1-ого завдання

<

def calculate\_sum\_and\_product(number):

"""

Обчислює суму та добуток цифр тризначного числа.

Аргумент функції:

number (int): Тризначне число.

Повертає:

tuple: Кортеж із сумою та добутком цифр.

"""

# Перевірка, чи введене число є тризначним

if 100 <= number <= 999:

# Розбиваємо число на цифри

firstnumber = number // 100

secondnumber = (number % 100) // 10

thirdnumber = number % 10

# Обчислити суму і добуток цифр

digit\_sum = firstnumber + secondnumber + thirdnumber

digit\_product = firstnumber \* secondnumber \* thirdnumber

return digit\_sum, digit\_product

# Зробити виняток, якщо число не є тризначним

else:

raise ValueError("Введіть тризначне число")

>

Функція 2-ого завдання

<

def calculate\_math\_expression(x):

"""

Обчислює математичний вираз.

Аргумент функції:

x (float): Числове значення.

Повертає:

float: Результат виразу.

"""

# Обчислення математичного виразу

try:

y = (math.log2(abs(x)) \* math.sin(math.radians(x + 35))\*\*2) / \

(3\*\*(x - 1) \* (2 \* math.pi + math.cos(x) \* 1/2)\*\*(1/3))

return y

# Перевірка на нуль у знаменнику

except ZeroDivisionError:

raise ValueError("Знаменник не може бути нулем")

# Уникнення інших помилок, наприклад введення невірних даних

except Exception as e:

raise ValueError(f"Помилка обчислення: {e}")

>

Функція 3-ого завдання

<

def check\_isosceles\_triangle(a, b, c):

"""

Перевіряє, чи є трикутник рівнобедреним.

Аргумент функції:

a (float): Довжина сторони a.

b (float): Довжина сторони b.

c (float): Довжина сторони c.

Повертає:

bool: True, якщо трикутник рівнобедрений, False в іншому випадку.

"""

return a == b or b == c or c == a

>

Загальний вигляд коду

<

import math

# -\*- coding: utf-8 -\*-

def calculate\_sum\_and\_product(number):

"""

Обчислює суму та добуток цифр тризначного числа.

Аргумент функції:

number (int): Тризначне число.

Повертає:

tuple: Кортеж із сумою та добутком цифр.

"""

# Перевірка, чи введене число є тризначним

if 100 <= number <= 999:

firstnumber = number // 100

secondnumber = (number % 100) // 10

thirdnumber = number % 10

# Обчислити суму і добуток цифр

digit\_sum = firstnumber + secondnumber + thirdnumber

digit\_product = firstnumber \* secondnumber \* thirdnumber

return digit\_sum, digit\_product

# Зробити виняток, якщо число не є тризначним

else:

raise ValueError("Введіть тризначне число")

def calculate\_math\_expression(x):

"""

Обчислює математичний вираз.

Аргумент функції:

x (float): Числове значення.

Повертає:

float: Результат виразу.

"""

# Обчислення математичного виразу

try:

y = (math.log2(abs(x)) \* math.sin(math.radians(x + 35))\*\*2) / \

(3\*\*(x - 1) \* (2 \* math.pi + math.cos(x) \* 1/2)\*\*(1/3))

return y

# Перевірка на нуль у знаменнику

except ZeroDivisionError:

raise ValueError("Знаменник не може бути нулем")

# Уникнення інших помилок, наприклад введення невірних даних

except Exception as e:

raise ValueError(f"Помилка обчислення: {e}")

def check\_isosceles\_triangle(a, b, c):

"""

Перевіряє, чи є трикутник рівнобедреним.

Аргумент функції:

a (float): Довжина сторони a.

b (float): Довжина сторони b.

c (float): Довжина сторони c.

Повертає:

bool: True, якщо трикутник рівнобедрений, False в іншому випадку.

"""

return a == b or b == c or c == a

def main():

#Відображення меню в консолі

print("1. Обчислити суму та добуток цифр")

print("2. Обчислити математичний вираз")

print("3. Перевірити, чи є трикутник рівнобедреним")

print("0. Вийти")

#Вибір завдання завдяки вводу відповідного номеру

choice = int(input("Оберіть завдання (0-3): "))

#Відповідні випадки у разі вводу цифр 0-3

if choice == 1:

try:

# Зчитуємо введене користувачем число

user\_input = int(input("Введіть тризначне число: "))

#Виклик функції до 1-ого завдання

result\_sum, result\_product = calculate\_sum\_and\_product(user\_input)

print(f"Сума цифр: {result\_sum}")

print(f"Добуток чисел: {result\_product}")

#Відповідні винятки

except ValueError as ve:

print(f"Помилка: {ve}")

except Exception as e:

print(f"Виникла помилка: {e}")

elif choice == 2:

try:

#Введення значення х для розрахунку виразу

x = float(input("Введіть x: "))

#Виклик функції другого завдання

result = calculate\_math\_expression(x)

print(f"Результат виразу: {result}")

#Відповідні винятки

except ValueError as ve:

print(f"Помилка: {ve}")

except Exception as e:

print(f"Виникла помилка: {e}")

elif choice == 3:

try:

#Введення сторін трикутника

a = float(input("Введіть довжину сторони a: "))

b = float(input("Введіть довжину сторони b: "))

c = float(input("Введіть довжину сторони c: "))

#Виклик функції до 3-ого завдання

result = check\_isosceles\_triangle(a, b, c)

print(f"Трикутник рівнобедрений: {result}")

#Відповідні винятки

except ValueError:

print("Будь ласка, введіть правильні числові значення для довжин сторін.")

except Exception as e:

print(f"Виникла помилка: {e}")

#Пункт меню для виходу з програми

elif choice == 0:

print("Вихід з програми.")

#Опрацювання випадку якщо введено неіснуючий номер меню

else:

print("Невірний вибір. Оберіть ще раз.")

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

main()

>

ДОДАТОК Б

Скрін-шоти вікна виконання програми

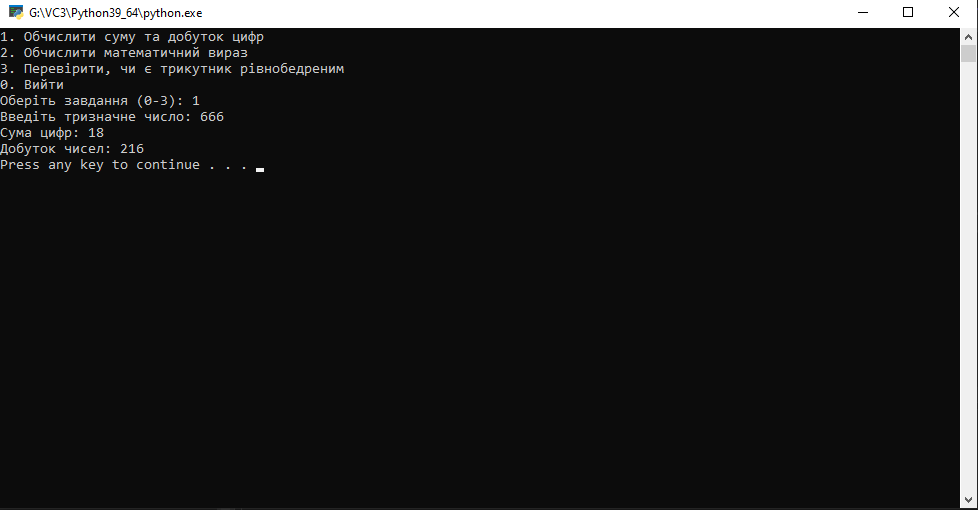


Рисунок Б.1 – Екран виконання програми для вирішення завдання <1 >

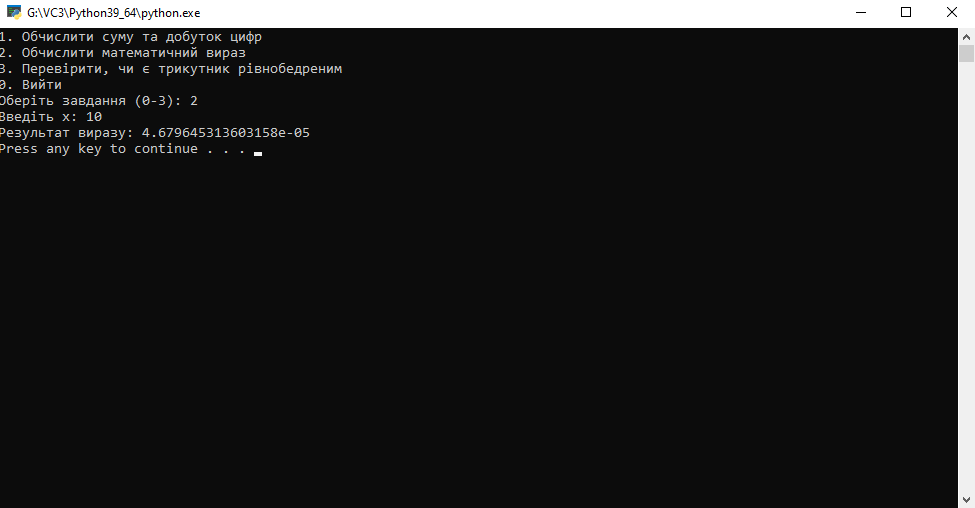


Рисунок Б.2 – Екран виконання програми для вирішення завдання <2 >

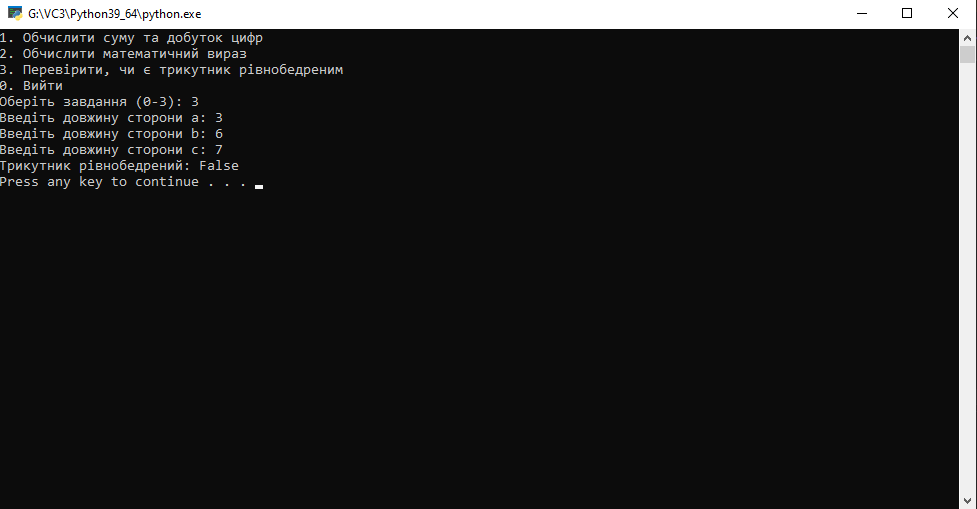


Рисунок Б.3 – Екран виконання програми для вирішення завдання <3 >