

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Національний аерокосмічний університет ім. М. Є. Жуковського  
«Харківський авіаційний інститут»

Кафедра систем управління літальними апаратами

Лабораторна робота № 2

з дисципліни «Об'єктно-орієнтоване проектування СУ»

Тема: «Розробка структурованих програм з  
розгалуженням та повтореннями»

ХАІ.301 . 3.320.2 ЛР

Виконав студент гр. 320

\_\_\_\_\_ Семеняга Ігор\_\_\_\_  
(підпис, дата) (П.І.Б.)

Перевірив

\_\_\_\_\_ к.т.н., доц. О. В. Гавриленко  
(підпис, дата) (П.І.Б.)

## МЕТА РОБОТИ

Вивчити теоретичний матеріал щодо синтаксису на мові Python і поданням у вигляді UML діаграм діяльності алгоритмів з розгалуження та циклами, а також навчитися використовувати функції, інструкції умовного переходу і циклів для реалізації інженерних обчислень.

### ПОСТАНОВКА ЗАДАЧІ

Завдання 1. Вирішити завдання на алгоритми з розгалуженням. Завдання представлено в табл.1.

Завдання 2. Дано дійсні числа  $(x_i, y_i)$ ,  $i = 1, 2, \dots, n$ , – координати точок на площині. Визначити кількість точок, що потрапляють в геометричну область

заданого кольору (або групу областей). Варіанти геометричних областей представлені в табл.2.

Завдання 3. Дослідити ряд на збіжність. Умова закінчення циклу обчислення суми прийняти у вигляді:  $|u_n| < \epsilon$  або  $|u_n| > G$  де  $\epsilon$  – мала величина

для переривання циклу обчислення суми сходиться ряду ( $\epsilon = 10^{-5}$   $10^{-20}$ );  $g$  величина для переривання циклу обчислення суми розходиться ряду ( $g = 10^2 \dots 10^6$ ). Варіанти представлено в табл.3.

Завдання 4. Для багаторазового виконання будь-якого з трьох зазначених вище завдань на вибір розробити циклічний алгоритм організації меню в командному вікні.

## ВИКОНАННЯ РОБОТИ

Завдання 1. Вирішення задачі 1, №6

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

a = float(input("Введіть перше число: ")) string, float

b = float(input("Введіть друге число: ")) string, float

Вихідні дані (ім'я, опис, тип):

print("Більше число:", a) string, float

print("Більше число:", b) string, float

print("Числа рівні.") string

Алгоритм вирішення

```
def task1(): # 6
    # Take two numbers as input from the user
    a = float(input("Введіть перше число: "))
    b = float(input("Введіть друге число: "))

    # Compare the two numbers and print the larger one
    if a > b:
        print("Більше число:", a)
    elif b > a:
        print("Більше число:", b)
    else:
        print("Числа рівні.")
```

Рисунок 1 – рис 1

Лістинг коду вирішення задачі наведено в дод. А (стор. 6). Екран роботи програми показаний на рис. Б.7.

Завдання 2. Вирішення задачі 2, №22

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

a = float(input("Введіть сторону квадрата (a): ")) float

r = float(input("Введіть радіус кола (r): ")) float

Вихідні дані (ім'я, опис, тип):

print("Кількість точок у темно-зеленій області: ", count) int, string

Алгоритм вирішення

```

def task2(): # 22
    # Введення параметрів
    a = float(input("Введіть сторону квадрата (a): "))
    r = float(input("Введіть радіус кола (r): "))

    # Введення координат точок
    n = int(input("Введіть кількість точок (n): "))
    points = []
    for i in range(n):
        x, y = map(float, input(f"Введіть координати точки {i+1} (x, y): ").split())
        points.append((x, y))

    # Перевірка, чи точка належить темно-зеленій області
    def is_in_dark_green_area(x, y, a, r):
        # Темно-зелена область - I and IV чверть квадрата, поза колом
        if -a / 2 <= x <= a / 2 and 0 <= y <= a / 2 and x**2 + y**2 > r**2:
            return True
        return False

    # Підрахунок точок у темно-зеленій області
    count = 0
    for x, y in points:
        if is_in_dark_green_area(x, y, a, r):
            count += 1

    print("Кількість точок у темно-зеленій області: ", count)

```

Рисунок 2 – рис 2

Лістинг коду вирішення задачі наведено в дод. А (стор. 6). Екран роботи програми показаний на рис. Б.7.

Завдання 3. Вирішення задачі 3, №13

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

`N = int(input("N = "))` int

Вихідні дані (ім'я, опис, тип):

`print("The approximate sum of the series for {N} terms is: ", total)` string, int

Алгоритм вирішення

```

# 13
def task3():

    # N input
    N = int(input("N = "))
    if N == 0:
        print("N cannot be 0 in denominator")
    else:
        total = 0
        for n in range(1, N + 1):
            # Calculate each term of the series
            numerator = 2**n * math.factorial(2*n - 1)
            denominator = math.sqrt(math.factorial(n))
            term = numerator / denominator
            total += term

    # total
    print("The approximate sum of the series for {N} terms is: ", total)

```

Рисунок 3 – рис 3

Завдання 4. Вирішення задачі 4

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

choice = input("Введіть номер опції (1-4): ") number, int

Вихідні дані (ім'я, опис, тип):

print("Вихід із програми.") string

print("Невірний вибір. Спробуйте ще раз.") string

Алгоритм вирішення

```
def main_menu():
    while True:
        print("\nМеню:")
        print("1. Виконати завдання 6")
        print("2. Виконати завдання 22")
        print("3. Виконати завдання 13")
        print("4. Вийти")

        choice = input("Введіть номер опції (1-4): ")

        if choice == '1':
            task1()
        elif choice == '2':
            task2()
        elif choice == '3':
            task3()
        elif choice == '4':
            print("Вихід із програми.")
            break
        else:
            print("Невірний вибір. Спробуйте ще раз.")

# Запуск програми
if __name__ == "__main__":
    main_menu()
```

Рисунок 4 – рис 4

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

## ВИСНОВКИ

Було вивчено теоретичний матеріал щодо синтаксису мови Python, а також принципи створення UML-діаграм діяльності для алгоритмів з розгалуженнями та циклами. Закріплено на практиці використання функцій, умовних операторів і циклів для реалізації алгоритмів, зокрема інженерних обчислень. Відпрацьовано застосування цих знань у написанні коду, що дозволило отримати навички побудови логічних конструкцій і структурованого програмування.

## ДОДАТОК А

## Лістинг коду програми до задач 1-№6, 2-№22, 3-№13, 4

```

import math

def task1(): # 6
    # Take two numbers as input from the user
    a = float(input("Введіть перше число: "))
    b = float(input("Введіть друге число: "))

    # Compare the two numbers and print the larger one
    if a > b:
        print("Більше число:", a)
    elif b > a:
        print("Більше число:", b)
    else:
        print("Числа рівні.")

def task2(): # 22
    # Введення параметрів
    a = float(input("Введіть сторону квадрата (a): "))
    r = float(input("Введіть радіус кола (r): "))

    # Введення координат точок
    n = int(input("Введіть кількість точок (n): "))
    points = []
    for i in range(n):
        x, y = map(float, input(f"Введіть координати точки {i+1} (x, y): ").split())
        points.append((x, y))

    # Перевірка, чи точка належить темно-зеленій області
    def is_in_dark_green_area(x, y, a, r):
        # Темно-зелена область - I and IV чверть квадрата, поза колом
        if -a / 2 <= x <= a / 2 and 0 <= y <= a / 2 and x**2 + y**2 > r**2:
            return True
        return False

    # Підрахунок точок у темно-зеленій області
    count = 0
    for x, y in points:
        if is_in_dark_green_area(x, y, a, r):
            count += 1

    print("Кількість точок у темно-зеленій області: ", count)

# 13

def task3():

```

```

# N input
N = int(input("N = "))
if N == 0:
    print("N cannot be 0 in denominator")
else:
    total = 0
    for n in range(1, N + 1):
        # Calculate each term of the series
        numerator = 2**n * math.factorial(2*n - 1)
        denominator = math.sqrt(math.factorial(n))
        term = numerator / denominator
        total += term

# total

print("The approximate sum of the series for {N} terms is: ", total)

def main_menu():

    while True:
        print("\nМеню:")
        print("1. Виконати завдання 6")
        print("2. Виконати завдання 22")
        print("3. Виконати завдання 13")
        print("4. Вийти")

        choice = input("Введіть номер опції (1-4): ")

        if choice == '1':
            task1()
        elif choice == '2':
            task2()
        elif choice == '3':
            task3()
        elif choice == '4':
            print("Вихід із програми.")
            break
        else:
            print("Невірний вибір. Спробуйте ще раз.")

# Запуск програми
if __name__ == "__main__":
    main_menu()

```



## ДОДАТОК Б

### Скрін-шоти вікна виконання програми

```

Меню:
1. Виконати завдання 6
2. Виконати завдання 22
3. Виконати завдання 13
4. Вийти
Введіть номер опції (1-4): 1
Введіть перше число: 12
Введіть друге число: 23
Більше число: 23.0

Меню:
1. Виконати завдання 6
2. Виконати завдання 22
3. Виконати завдання 13
4. Вийти
Введіть номер опції (1-4): 1
Введіть перше число: 10
Введіть друге число: 10
Числа рівні.

Меню:
1. Виконати завдання 6
2. Виконати завдання 22
3. Виконати завдання 13
4. Вийти
Введіть номер опції (1-4): |

```

Рисунок Б.1 – Екран виконання програми для вирішення завдання  
№6

```

Меню:
1. Виконати завдання 6
2. Виконати завдання 22
3. Виконати завдання 13
4. Вийти
Введіть номер опції (1-4): 2
Введіть сторону квадрата (a): 50
Введіть радіус кола (r): 10
Введіть кількість точок (n): 2
Введіть координати точки 1 (x, y): 5 5
Введіть координати точки 2 (x, y): 15 15
Кількість точок у темно-зеленій області: 1

Меню:
1. Виконати завдання 6
2. Виконати завдання 22
3. Виконати завдання 13
4. Вийти
Введіть номер опції (1-4):

```

Рисунок Б.2 – Екран виконання програми для вирішення завдання  
№22

```

Меню:
1. Виконати завдання 6
2. Виконати завдання 22
3. Виконати завдання 13
4. Вийти
Введіть номер опції (1-4): 3
N = 1
The approximate sum of the series for (N) terms is: 2.0

Меню:
1. Виконати завдання 6
2. Виконати завдання 22
3. Виконати завдання 13
4. Вийти
Введіть номер опції (1-4): 3
N = 2
The approximate sum of the series for (N) terms is: 18.97056274847714

Меню:
1. Виконати завдання 6
2. Виконати завдання 22
3. Виконати завдання 13
4. Вийти
Введіть номер опції (1-4):

```

Рисунок Б.3 – Екран виконання програми для вирішення завдання  
№13

Меню:

1. Виконати завдання 6
2. Виконати завдання 22
3. Виконати завдання 13
4. Вийти

Введіть номер опції (1-4) : |

Рисунок Б.4 – Екран виконання програми для вирішення завдання