

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 програм для
мобільних систем»

Тема: «Розробка структурованих програм з
розгалуженням та повтореннями»

ХАІ.301 .151 .322 .2 ЛР

Виконав студент гр. _____322_____

_____Сироватський Дмитро_____
(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

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

ВИКОНАННЯ РОБОТИ

Завдання 1. Вирішення задачі task1_if20

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

- a (координата точки A, дійсне число)
- b (координата точки B, дійсне число)
- c (координата точки C, дійсне число)

Вихідні дані (ім'я, опис, тип):

- closest_point (точка, яка розташована найближче до точки A, дійсне число)
- distance (відстань від точки A до найближчої точки, дійсне число)

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

Завдання 3. Вирішення задачі task3_20

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

- k (кількість членів суми, ціле число)

Вихідні дані (ім'я, опис, тип):

- s (сума k членів ряду, дійсне число)

Лістинг коду вирішення задачі наведено в дод. А (стор. 5). Екран роботи програми показаний на рис. Б.2.

ВИСНОВКИ

У процесі виконання цієї роботи було вивчено та закріплено на практиці основи програмування на Python, зокрема використання умовних операторів та циклів. Було розроблено дві програми: одна для визначення точки, яка розташована найближче до заданої, та інша для обчислення суми ряду. Ці завдання допомогли отримати практичні навички вирішення інженерних задач за допомогою програмування. Загалом, ця робота сприяла кращому розумінню та відпрацюванню основних концепцій програмування.

ДОДАТОК А

Лістинг коду програми до задач Завдання 1. task1_if20

```

<
print("\ntask1_if20\n")
def task1_if20():
    # Ввести координати точок А, В, С
    a = float(input("Введіть координату точки А: "))
    b = float(input("Введіть координату точки В: "))
    c = float(input("Введіть координату точки С: "))

    # Обчислити відстані від А до В і С
    d_ab = abs(a - b) # Модуль різниці координат
    d_ac = abs(a - c)

    # Порівняти відстані і вивести результат
    if d_ab < d_ac:
        print(f"\nТочка В ближче до А, відстань = {d_ab}\n")
    elif d_ab > d_ac:
        print(f"\nТочка С ближче до А, відстань = {d_ac}\n")
    else:
        print(f"\nТочки В і С рівновіддалені від А, відстань = {d_ab}\n")

task1_if20()
>

```

Лістинг коду програми до задач Завдання 3. Task3_20

```

<
print("\ntask3_20\n")
def task3_20():
    # Імпортувати модуль math для використання константи e і функції sqrt
    import math

    # Ввести кількість членів суми
    k = int(input("Введіть кількість членів суми: "))

    # Ініціалізувати змінну для зберігання суми
    s = 0

    # Використати цикл for для обчислення суми
    for n in range(1, k + 1):
        # Обчислити n-й член суми
        t = math.e ** n / n ** math.sqrt(n / 2)
        # Додати член до суми
        s += t

    # Вивести результат
    print(f"\nСума {k} членів ряду = {s}")

```

```
task3_20()
```

```
>
```

ДОДАТОК Б

Скрін-шоти вікна виконання програми

```
task1_if20  
  
Введіть координату точки А: 10  
Введіть координату точки В: 30  
Введіть координату точки С: 50  
  
Точка В ближче до А, відстань = 20.0
```

Рисунок Б.1 – Екран виконання програми для вирішення Завдання 1. task1_if20

```
task3_20  
  
Введіть кількість членів суми: 5  
  
Сума 5 членів ряду = 30.979216546195957
```

Рисунок Б.2 – Екран виконання програми для вирішення Завдання 3. task2_20