

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

Кафедра систем управління літальними апаратами

Лабораторна робота № 3

з дисципліни «Об'єктно-орієнтоване проектування програм для
мобільних систем»

Тема: «Структурування програм з використанням
функцій»

ХАІ.301 .151 .322 .3 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 із синтаксису визначення і виклику функцій та особливостей послідовностей у Python, а також документацію бібліотеки `numpy`; отримати навички реалізації бібліотеки функцій з параметрами, що структурують вирішення завдань «згори – до низу».

ПОСТАНОВКА ЗАДАЧІ

Завдання 1. Описати функцію відповідно до варіанту. Для виклику функції (друга частина задачі) описати іншу функцію, що на вході має список вхідних даних і повертає список вихідних даних. Введення даних, виклик функції та виведення результатів реалізувати в третій функції без параметрів. Завдання наведено в табл.1.

Завдання 2. Розробити дві вкладені функції для вирішення задачі обробки двовимірних масивів відповідно до варіанту: зовнішня – без параметрів, внутрішня має на вході ім'я файлу з даними, на виході – підраховані параметри матриці (перша частина задачі) та перетворену матрицю (друга частина задачі). Для обробки масивів використати функції бібліотеки `numpy`. Завдання представлено в табл.2.

ВИКОНАННЯ РОБОТИ

Завдання 1. Вирішення задачі task1_Proc13

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

- A, B, C: три числа, які потрібно відсортувати за спаданням, цілі числа, немає обмежень
- A1, B1, C1: три числа для першого набору, які потрібно відсортувати за спаданням, цілі числа, немає обмежень
- A2, B2, C2: три числа для другого набору, які потрібно відсортувати за спаданням, цілі числа, немає обмежень

Вихідні дані (ім'я, опис, тип):

- lst[0], lst[1], lst[2]: три числа, відсортовані за спаданням, цілі числа
- A1, B1, C1: три числа для першого набору, відсортовані за спаданням, цілі числа
- A2, B2, C2: три числа для другого набору, відсортовані за спаданням, цілі числа

Лістинг коду вирішення задачі наведено в дод. А (стор. 5). Екран роботи програми показаний на рис. Б.1.(стор. 7)

Завдання 2. Вирішення задачі task2_Matrix18

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

- matrix: вихідна матриця, яку треба обробити, numpy array, розмір M x N, елементи - цілі числа від 1 до 100
- M = 5
- N = 5

Вихідні дані (ім'я, опис, тип):

- unique_cols: кількість стовпців з різними елементами, int
- new_matrix: нова матриця після додавання M рядків з випадковими числами, numpy array, розмір 2M x N, елементи - цілі числа від 1 до 100

Лістинг коду вирішення задачі наведено в дод. А (стор. 5). Екран роботи програми показаний на рис. Б.2. (стор. 7).

ВИСНОВКИ

У процесі роботи було вивчено основи програмування на мові Python, закріплено на практиці використання змінних, циклів, умов та функцій. Також було відпрацьовано в коді програми розв'язання різних задач, отримано навички аналізу та тестування коду.

ДОДАТОК А

Лістинг коду програми до задач Завдання 1. task1_Proc13

```

<
print("\ntask1_Proc13\n") # виводимо назву задачі

def SortDec3(A, B, C):
    """Сортуємо A, B, C за зменшенням"""
    lst = sorted([A, B, C], reverse=True) # створюємо список з A, B, C та сортуємо
    його у зворотному порядку
    return lst[0], lst[1], lst[2] # повертаємо перший, другий та третій елементи
    списку

def Proc13():
    """Сортуємо два набори з трьох чисел за спаданням"""
    A1, B1, C1 = map(int, input("\nВведіть три числа для першого набору (через
    пропуск): ").split()) # зчитуємо три числа для першого набору і перетворюємо їх
    на цілі
    A2, B2, C2 = map(int, input("\nВведіть три числа для другого набору (через
    пропуск): ").split()) # зчитуємо три числа для другого набору і перетворюємо їх
    на цілі
    A1, B1, C1 = SortDec3(A1, B1, C1) # сортуємо перший набір за спаданням
    A2, B2, C2 = SortDec3(A2, B2, C2) # сортуємо другий набір за спаданням
    print(f"\nПерший набір, відсортований за зменшенням: {A1}, {B1}, {C1} ") #
    виводимо перший набір
    print(f"\nДругий набір, відсортований за зменшенням: {A2}, {B2}, {C2} \n") #
    виводимо другий набір
    Proc13() # Викликаємо функцію Proc13>

```

Лістинг коду програми до задач Завдання 2. task2_Matrix18

```

<

print("\ntask2_Matrix18\n") # виводимо назву задачі

def Matrix18(matrix):
    """Знаходимо кількість стовпців з різними елементами і додаємо до матриці
    рядків M з випадковими числами"""
    M, N = matrix.shape # отримуємо розміри матриці
    unique_cols = sum(1 for col in matrix.T if len(set(col)) == len(col)) #
    вважаємо кількість стовпців, у яких всі елементи різні
    print(f"\nКількість стовпців з різними елементами: {unique_cols} \n") #
    виводимо результат
    random_rows = np.random.randint(1, 100, size=(M, N)) # створюємо M рядків із
    випадковими числами від 1 до 100
    new_matrix = np.concatenate([matrix, random_rows], axis=0) # об'єднуємо
    вихідну матрицю та нові рядки по вертикалі
    print("Нова матриця після додавання випадкових рядків: \n") # виводимо
    повідомлення

```

```
print(new_matrix) # виводимо нову матрицю

if __name__ == "__main__":
    M = 5
    N = 5
    matrix = np.random.randint(1, 100, size=(M, N)) # створюємо матрицю з
випадкових чисел від 1 до 100
    Matrix18(matrix) # Викликаємо функцію Matrix18
>
```

ДОДАТОК Б
Скрін-шоти вікна виконання програми

```
task1_Proc13

Введіть три числа для першого набору (через пропуск): 1 2 3
Введіть три числа для другого набору (через пропуск): 3 2 1
Перший набір, відсортований за зменшенням: 3, 2, 1
Другий набір, відсортований за зменшенням: 3, 2, 1
```

Рисунок Б.1 – Екран виконання програми для вирішення Завдання 1.
task1_Proc13

```
task2_Matrix18

Кількість стовпців з різними елементами: 5
Нова матриця після додавання випадкових рядків:

[[ 5 78 54 92 47]
 [31 39 34 90 72]
 [40 26 55 86 37]
 [23 87 24 39  1]
 [42 91 52  4 11]
 [12 47 86 50 15]
 [83 68 68 16 80]
 [12 81 63 84 68]
 [ 7 42 71 40 59]
 [82 68 94 75 90]]
```

Рисунок Б.2 – Екран виконання програми для вирішення Завдання 2.
task2_Matrix18