# Міністерство освіти і науки України Національний університет "Львівська політехніка"

Кафедра ЕОМ



до лабораторної роботи №7

3 дисципліни: «Кросплатформні засоби програмування»

На тему: «ДОСЛІДЖЕННЯ БАЗОВИХ КОНСТРУКЦІЙ МОВИ РҮТНОN» Варіант 16

## Виконав:

ст. групи КІ-306

Мілян М.О.

## Прийняв:

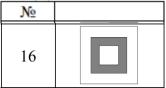
доцент кафедри ЕОМ

Олексів М.В.

Meta: ознайомитися з базовими конструкціями мови Python.

#### Завдання:

- 1. Написати та налагодити програму на мові Python згідно варіанту. Програма має задовольняти наступним вимогам:
  - програма має розміщуватися в окремому модулі;
- програма має генерувати зубчатий список, який міститиме лише заштриховані області квадратної матриці згідно варіанту;
- розмір квадратної матриці і символ-заповнювач масиву вводяться з клавіатури;
- при не введені або введенні кількох символів-заповнювачів відбувається коректне переривання роботи програми;
  - сформований масив вивести на екран;
  - програма має містити коментарі.
- 2. Завантажити код на GitHub згідно методичних вказівок по роботі з GitHub.
- 3. Скласти звіт про виконану роботу з приведенням тексту програми, результату її виконання та фрагменту згенерованої документації та завантажити його у ВНС.
- 4. Дати відповідь на контрольні запитання.



Хід роботи

# Код програми:

# lab7.py

```
# Запитуємо у користувача розмір квадрата
n_rows = int(input("Введіть розмір квадрата: "))
# Якщо розмір квадрата менший за 6, програма завершується
if n_rows < 6:
   print("Введіть кількість більшу за 5")
    exit()
# Запитуємо у користувача символ для заповнення квадрата
symbol = input("Введіть символ для заповнення: ")
if len(symbol) == 0:
    print("Некорректний символ. Відсутній символ.")
    exit()
elif len(symbol) > 1:
    print("Некорректний ввід. Забагато символів.")
    exit()
symbol = symbol[0]
# Визначаємо відступ для порожньої рамки
offset = n_rows//6
```

```
# Створюємо зубчастий масив для зберігання символів квадрата
for i in range(n_rows - 2 * offset):
    if(i < offset or i >= n_rows - 3 * offset):
        # Зовнішні частини матимуть максимальну ширину
        arr.append([symbol] * (n_rows - 2 * offset))
    else:
        # Внутрішні частини матимуть меншу ширину
        arr.append([symbol] * (2 * offset))
# Заповнюємо квадрат символами
for i in range(n_rows):
    arrIndx = 0 # Індекс для внутрішньої частини заповненого
    for j in range(n_rows):
        # Заповнюємо квадрат за допомогою символу або пробілу
        if (i >= offset and i < n_rows - offset and j >= offset and j < n_rows -</pre>
offset and
        (i < 2 * offset or i >= n_rows - 2 * offset or j < 2 * offset or j >=
n_rows - 2 * offset)):
            print(arr[i - offset][arrIndx], end='')
            arrIndx+=1
        else:
            # Якщо поза межами заповненої частини, додаємо пробіл
            print(' ', end='')
    # Додаємо новий рядок на екран
    print()
```

```
Введіть розмір квадрата: 25
Введіть символ для заповнення: #
  *****************
  ####
  ####
         ####
  ####
         ####
         ####
  ####
         ####
  ####
         ####
  ####
         ####
  ####
         ####
  ####
         ####
```

Рис.1 Вивід результату у консоль

**Висновок:** На лабораторній роботі я ознайомився з базовими конструкціями мови Python.