

Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Кафедра «Електронних обчислювальних машин»



Звіт

з лабораторної роботи № 7
з дисципліни: «Кросплатформенні засоби програмування»

На тему: «Дослідження базових конструкцій мови Python»

Виконав:

студент групи КІ-306
Бокало П.М.

Прийняв:

доцент кафедри ЕОМ
Іванов Ю. С.

Мета роботи: ознайомитися з базовими конструкціями мови Python.

Завдання(варіант №2):

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 введенні 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.

Індивідуальне завдання:



Вихідний код програми:

Файл Vokalo.KI_306_1.Lab7.py:

```
# Функція для головного виконання програми
def main():
    # Запитуємо користувача про символ для заповнення та кількість рядків та стовпців
    symbol = input("Введіть символ для заповнення: ")
    N = int(input("Введіть кількість рядків та стовпців: "))

    # Обробка негативного значення N
    if N < 0:
        N *= -1
    # Обробка нульового значення N
    elif N == 0:
        print("Некоректне значення для N")
        return
    # Обробка непарного значення N
    if N % 2 != 0:
        N += 1
        print(f"Ваше число було непарним, тому тепер воно становить: {N}")
```

```

print(f"\nВаш символ для заповнення: {symbol}\n")

# Перевірка, чи введено лише один символ
if len(symbol) > 1:
    print("Ви ввели занадто багато символів")
    return

# Створюємо порожній зубчастий список
matrix = []

# Заповнюємо зубчастий список рядками символів для першого трикутника
for i in range(N // 2):
    row = [symbol] * (i + 1)
    matrix.append(row)

# Заповнюємо зубчастий список рядками символів для другого трикутника
for i in range(N // 2, N):
    row = [""] * (N // 2) + [symbol] * (i - N // 2 + 1)
    matrix.append(row)

# Виводимо зубчастий список на екран та записуємо у файл
with open("Lab1.txt", "w") as fout:
    for row in matrix:
        print("\t".join(row))
        fout.write("\t".join(row) + "\n")

# Викликаємо головну функцію при запуску програми
if __name__ == "__main__":
    main()

```

Результат роботи програми:

```

C:\Windows\system32\cmd.exe
Введіть символ для заповнення: *
Введіть кількість рядків та стовпців: -11
Ваше число було непарним, тому тепер воно становить: 12
Ваш символ для заповнення: *
*
*  *
*  *  *
*  *  *  *
*  *  *  *  *
*  *  *  *  *  *
          *
          *  *
          *  *  *
          *  *  *  *
          *  *  *  *  *
          *  *  *  *  *  *
Press any key to continue . . .

```

Файл Lab1.txt

```
print(f"\nВаш символ для заповнення: {symbol}\n")

if 1 < symbol < 10:
    with open('triangle.txt', 'w') as file:
        for i in range(1, symbol + 1):
            for j in range(1, i + 1):
                file.write(symbol + " ")
            file.write("\n")
        file.close()
    name = input("Ваше ім'я: ")
    main()
else:
    print("Введіть правильне значення!")
    main()

```

Відповідь на контрольні питання

1. Який вигляд має програма мовою Python?

Програма на Python представляє собою текстовий файл з розширенням ".py", який містить послідовність команд і виразів, що виконуються інтерпретатором Python.

2. Як запустити на виконання програму мовою Python?

Для запуску програми Python, ви можете використовувати командний рядок і ввести `python ваша_програма.py`, де `ваша_програма.py` - це назва вашого файлу з програмою.

3. Які коментарі підтримує Python?

Python підтримує однорядкові коментарі, які починаються з символу `#`, а також багаторядкові коментарі, які оточуються трьома однаковими символами, наприклад, `'''` або `"""`.

4. Які типи даних підтримує Python?

Python підтримує різноманітні типи даних, включаючи цілі числа (int), числа з плаваючою точкою (float), рядки (str), списки (list), кортежі (tuple), словники (dict) та інші.

5. Як оголосити змінну?

Змінну в Python можна оголосити просто присвоєнням значення, наприклад, `змінна = 10`.

6. Які керуючі конструкції підтримує Python?

Python підтримує умовні конструкції (if, elif, else), цикли (for, while), виконання функцій та інші керуючі конструкції.

7. Які операції підтримує Python?

Python підтримує різноманітні операції, включаючи арифметичні операції (+, -, *, /), порівняння (==, !=, <, >), логічні операції (and, or, not) та інші.

8. Як здійснити ввід з консолі?

Для здійснення вводу з консолі використовуються функції `input()`, наприклад, `змінна = input("Введіть дані: ")`.

9. Як здійснити вивід у консоль?

Для виводу у консоль використовується функція `print()`, наприклад, `print("Це вивід у консоль")`.

10. Як здійснити приведення типів?

Приведення типів в Python виконується за допомогою функцій, наприклад, `int()`, `float()`, `str()`, які перетворюють значення з одного типу в інший.

Висновок: на цій лабораторній роботі, я ознайомився з базовими конструкціями мови Python.