

Міністерство освіти і науки України
Національний університет «Львівська політехніка»

Кафедра ЕОМ

Звіт



Лабораторна робота № 7

“ ДОСЛІДЖЕННЯ БАЗОВИХ КОНСТРУКЦІЙ МОВИ PYTHON ”
з курсу “Кросплатформні засоби програмування”
Варіант: 4

Виконав:

ст.гр.КІ-205

Воробець Тетяна

Прийняв:

доцент кафедри ЕОМ

Олексів М.В.

Львів 2024

Мета: ознайомитися з базовими конструкціями мови Python.

Теоретичні відомості: Python програми – це набір *.py файлів. Кожен файл – це окремий модуль. Модулі підключаються за допомогою ключового слова `import` після якого вказується назва файлу без розширення `.py`. Щоб доступитися з одного модуля до іншого вони мають бути в одному каталозі. Якщо модуль лежить в іншому каталозі, то щоб до нього доступитися цей каталог має бути оформлений як пакет. Для цього достатньо у нього помістити порожній файл `__init__.py`. При підключенні пакету виконується вміст файлу `__init__.py`, який може містити визначення імен і інші необхідні для роботи з пакетом конфігураційні дії.

Для запуску на виконання програми мовою Python слід виконати в командному рядку: `python.exe .py`. Запустивши інтерпретатор `Python.exe`, можна вводити з командного рядка програму по-рядково і зразу отримувати результат виконання.

Типи даних Python

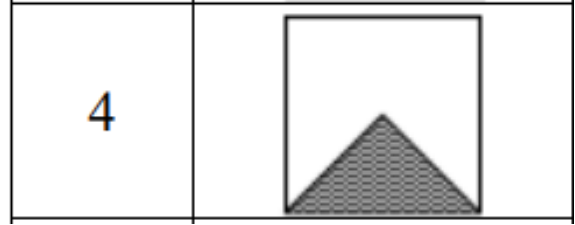
| | |
|------------------------------------|---|
| Текстовий тип: | <code>str</code> |
| Числові типи: | <code>int, float, complex</code> |
| Послідовності: | <code>list, tuple, range</code> |
| Типи-відповідності (Mapping type): | <code>dict</code> |
| Множини: | <code>set, frozenset</code> |
| Булівські типи: | <code>bool</code> |
| Бінарні типи: | <code>bytes, bytearray, memoryview</code> |
| Ніякий тип (None Type): | <code>NoneType</code> |

Цикл `for` у python має деякі відмінності в порівнянні з іншими мовами програмування. Розглянемо принцип його роботи. Змінній `x` по чергово присвоюються елементи, що знаходяться у ітераторі, і для кожного з них виконуються оператори тіла циклу. Після завершення виконання циклу виконується блок операторів після `else`.

Введення даних з клавіатури: зчитування рядка зі стандартного пристрою введення `sys.stdin` (клавіатура) в мові Python здійснюється за допомогою функції `input([prompt])`. Необов'язковий параметр `prompt`, призначений для вказання запрошення до введення, та буде виведений на стандартний пристрій виведення `sys.stdout` (екран). Функція повертає введений користувачем рядок після натискання клаваши `Enter`.

Вивід даних на екран: виведення на стандартний пристрій виведення `sys.stdout` (екран) можна здійснити функцією `print()`. Вона приймає наступні параметри: `print(*objects, sep=' ', end='\n', file=sys.stdout, flush=False)` де: `*objects` – послідовність об'єктів розділених комами (змінні, рядки, константи), значення яких потрібно вивести

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 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.

Код програми:

matrix.py

```
#Write a matrix in file
def write_file(arr):
    with open("MyFile.txt", "w") as fout:
        for row in arr:
            row_str = ' '.join(row)
            print(row_str)
            fout.write(row_str + '\n')

#Create matrix
def create_matrix(size, filler):
    if len(filler) != 1:
        print("Wrong filler")
        return

    arr = []

    for i in range(size):
        row = [' '] * (i + 1)
        for j in range(i + 1):
            if i + j >= size - 1:
                row[j] = filler
        arr.append(row)

    return arr
```

main.py

```
import matrix

#Start project

if __name__ == '__main__':
    #Size square matrix
```

```
size = int(input("Введіть розмір квадратної матриці: "))

#Filler square matrix
filler = input("Введіть символ-заповнювач: ")

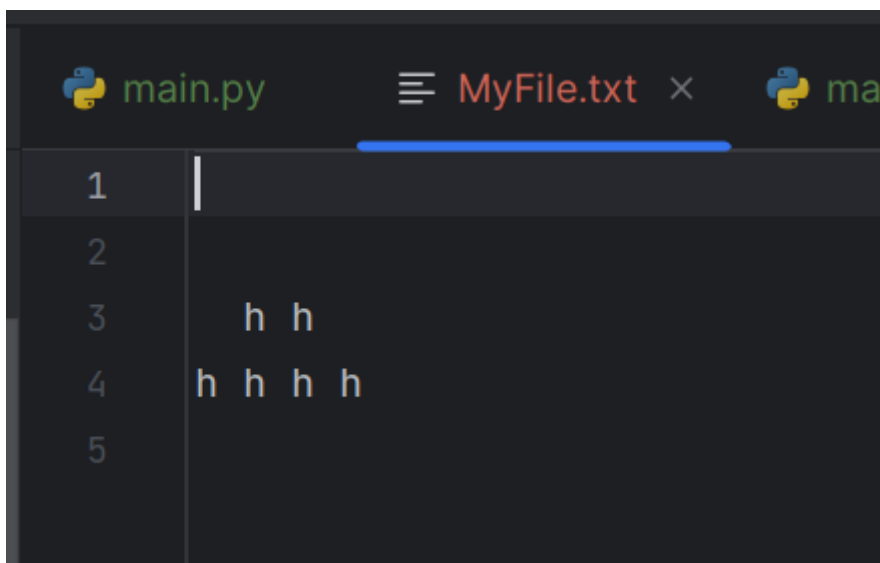
matrix.write_file(matrix.create_matrix(size, filler))
```

Результат:

```
C:\Users\Tanya\AppData\Local\Programs\Python
Введіть розмір квадратної матриці: 4
Введіть символ-заповнювач: h

  h h
h h h h

Process finished with exit code 0
```



Посилання на репозиторій:

https://github.com/NikaDe7/CPPT_Vorobets_TI_KI-35_1.git

Висновок: ознайомила з базовими конструкціями мови Python.