

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 *KI-306*

Олесько Б. А.

Прийняв:

доцент кафедри ЕОМ


Іванов Ю. С.

Львів – 2023

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.

№	
11	

```
# Введення розміру квадратної матриці
n_ROWS = int(input("Введіть розмір квадратної матриці: "))

# Перевірка на додатність
if n_ROWS < 0:
    n_ROWS = abs(n_ROWS)
    print("Розмір квадратної матриці є від'ємний, тому ми перевкладемо його в додатній:", n_ROWS)

# Перевірка на парність
if n_ROWS % 2 != 0:
    print("Розмір квадратної матриці збільшений на 1.")
    n_ROWS += 1

# Введення символу-заповнювача
symbol = input("Введіть символ-заповнювач: ")
symbol1 = input("Введіть символ-заповнювач2: ")
# Перевірка на правильність введенного символу
if len(symbol) != 1:
    print("\nСимвол-заповнювач введено невірно.")
    exit(0)
if len(symbol1) != 1:
    print("\nСимвол-заповнювач введено невірно.")
    exit(0)

# Відкриття файлу для запису
with open("MyFile.txt", "w") as fout:
    # Запис розміру квадратної матриці та символу-заповнювача
    fout.write(f"Розмір квадратної матриці: {n_ROWS}\n")
    fout.write(f"Символ-заповнювач: {symbol}\n")

arr = []

for i in range(n_ROWS):
    if i < n_ROWS // 2:
        print("\t" * (n_ROWS // 2), end="")
        fout.write("\t" * (n_ROWS // 2))

    arr.append(['+' if j == 0 else symbol for j in range(n_ROWS // 2)])
```

```

for j in range(n_ROWS // 2):
    # arr[i][0] = '+'
    arr[i][j] = symbol[0]
    print(arr[i][j] + "\t",end="" )
    fout.write(arr[i][j] + "\t")

print("\n")
print()
fout.write("\n")
else:
    arr.append([symbol for _ in range(n_ROWS)])
    for j in range(n_ROWS):
        if(j<(n_ROWS / 2)):
            arr[i][j] = symbol[0]
            print(arr[i][j] + "\t",end="")
            fout.write(arr[i][j] + "\t")

        else:
            arr[i][j] = symbol1[0]
            print(arr[i][j] + "\t",end="")
            fout.write(arr[i][j] + "\t")

print("\n")

print()
fout.write("\n")

print("Дані були записані в файл MyFile.txt.")

```

Результат

```

Введіть розмір квадратної матриці: 6
Введіть символ-заповнювач: !
Введіть символ-заповнювач2: "

      !      !      !

      !      !      !

      !      !      !

!      !      !      "      "      "

!      !      !      "      "      "

!      !      !      "      "      "

Дані були записані в файл MyFile.txt.

```

Відповіді на контрольні завдання

1. `print("Привіт, світ!")`
2. Для запуску програми мовою Python, ви можете використовувати командний рядок і ввести `python ім'я_файлу.py`, де `ім'я_файлу.py` - назва вашого файлу з програмою.
3. Python підтримує однорядкові коментарі, які починаються з символу `#`, а також багаторядкові коментарі, які обрамляються трьома однаковими символами.
4. Python підтримує різні типи даних, включаючи цілі числа, дійсні числа, рядки, списки, кортежі, словники, множини, булеві значення та інші.
5. Для оголошення змінної в Python використовують ключове слово `var_name = value`, де `var_name` - ім'я змінної, а `value` - значення, яке ви присвоюєте змінній.
6. Python підтримує керуючі конструкції, такі як умовні оператори (`if, elif, else`), цикли (`for, while`), оператори вибору (`break, continue`) і інші.
7. Python підтримує різні операції, включаючи арифметичні операції (`+, -, *, /`), операції порівняння (`==, !=, <, >`), операції логічної зміни (`and, or, not`) та інші.
8. `user_input = input("Введіть щось: ")`
9. `print("Це вивід у консоль")`
10. `num_str = "42"`
`num_int = int(num_str)`

Висновок на даній лабораторній роботі я навчився основам програмування на PYTHON.