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 |
| Ширий Б. І. |
| **Прийняв:** |
| доцент кафедри ЕОМ |
| Іванов Ю. С. |

Львів – 2023

# Методичні відомості роботи

## Мета

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

Для розробленої програми згенерувати документацію

### №3

Завантажити код на GitHub згідно методичних вказівок по роботі з GitHub.

### №4

Скласти звіт про виконану роботу з приведенням тексту програми, результату її виконання та фрагменту згенерованої документації та завантажити його у ВНС.

### №5

Дати відповідь на контрольні запитання.

# Виконання лабораторної роботи

## Вихідний код

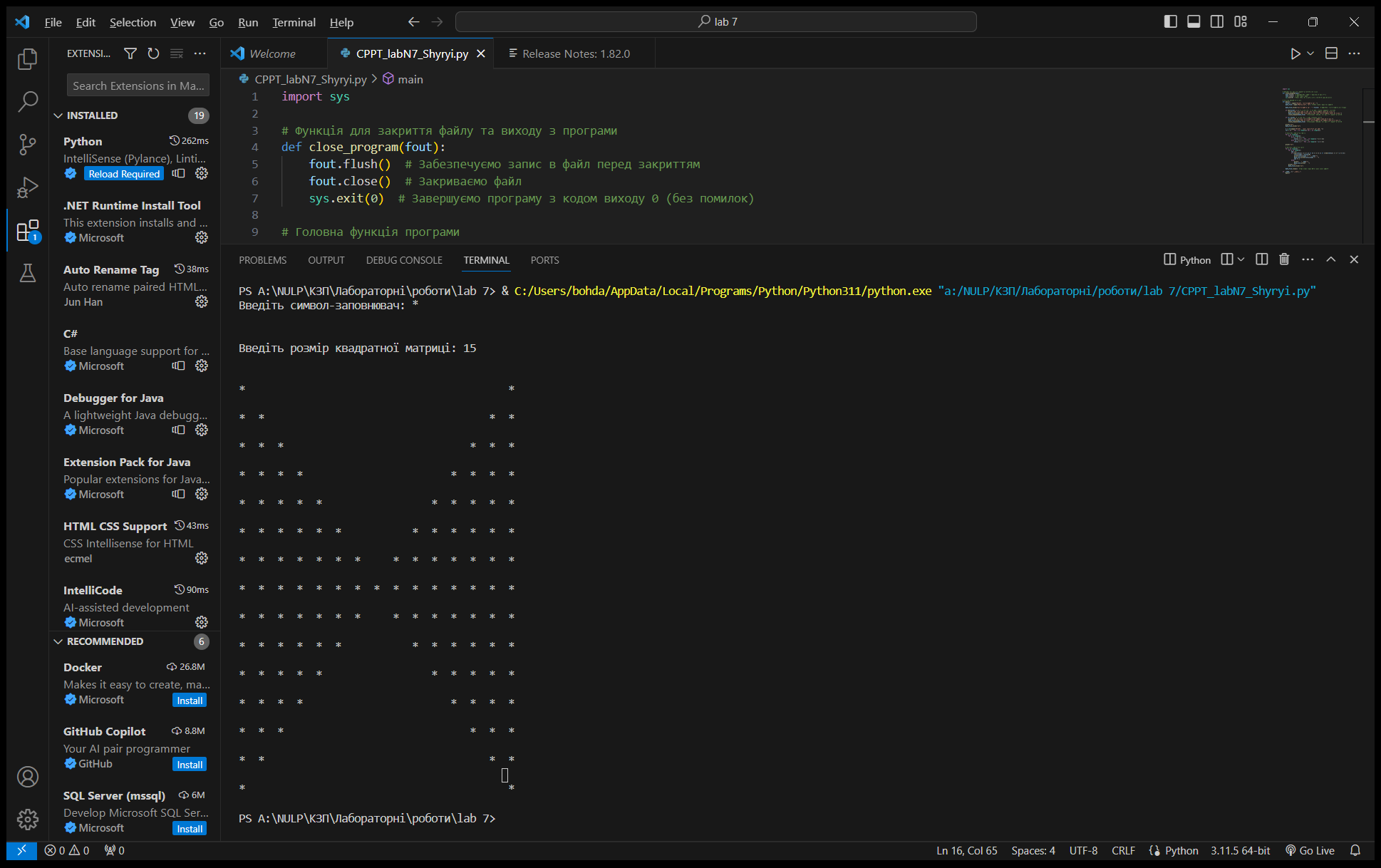
Написав програму, яка генерує зубчатий список, який містить лише заштриховані області квадратної матриці згідно варіанту. Код програми наведено у лістингу 2.1.

*Лістинг 2.1. Код основної програми.*

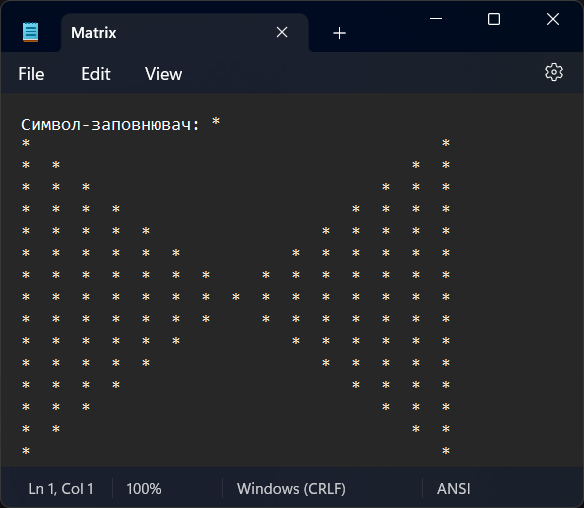
|  |
| --- |
| import sys  # Функція для закриття файлу та виходу з програми  def close\_program(fout):      fout.flush()  # Забезпечуємо запис в файл перед закриттям      fout.close()  # Закриваємо файл      sys.exit(0)  # Завершуємо програму з кодом виходу 0 (без помилок)  # Головна функція програми  def main():      filler = input("Введіть символ-заповнювач: ")      data\_file = open("Matrix.txt", "w")  # Відкриваємо файл для запису      data\_file.write("Символ-заповнювач: " + filler)  # Записуємо символ-заповнювач у файл      if len(filler) > 1: # Перевіряємо чи не було введено забагато символів          print("\nВи ввели більше ніж один символ!\nЗавершення програми.")          data\_file.write("\nВи ввели більше ніж один символ!\nЗавершення програми.")          close\_program(data\_file)  # Викликаємо функцію для закриття файлу та виходу      if not filler: # Перевіряємо чи був введений символ          print("\nВи не ввели символ-заповнювач!\nЗавершення програми.")          data\_file.write("\nВи не ввели символ-заповнювач!\nЗавершення програми.")          close\_program(data\_file)  # Викликаємо функцію для закриття файлу та виходу      print("\n")      data\_file.write("\n")      n = int(input("Введіть розмір квадратної матриці: "))      lst = [[' ' for \_ in range(n)] for \_ in range(n)]      # Генеруємо зубчастий список      for i in range(n):          for j in range(n):              if (i + 1) \* 2 < n:                  lst[i] = [' ' for \_ in range((i + 1) \* 2)]              elif (i + 1) \* 2 > n + 2:                  lst[i] = [' ' for \_ in range((n - i) \* 2)]      print("\n")      # Виводимо зубчастий масив      for i in range(n):          for j in range(n):              num = 0              if (len(lst[i]) // 2 > j) or (n // 2 == i) or ((len(lst[i]) // 2 + j >= n)):                  lst[i][num] = filler[0]                  print(lst[i][num] + "  ", end="")                  data\_file.write(lst[i][num] + "  ")                  num += 1              else:                  print("   ", end="")                  data\_file.write("   ")          print("\n")          data\_file.write("\n")      data\_file.close()  # Закриваємо файл після завершення запису  if \_\_name\_\_ == "\_\_main\_\_":      main() |

## Результати виконання

Вивід програми в консоль наведено на рисунку 2.1, а запис у файл у лістингу 2.2.



*Рисунок 2.1. Вивід в консоль.*



*Рисунок 2.2. Запис у файл.*

## Відповіді на контрольні питання

### Який вигляд має програма мовою Python?

Програма мовою Python має вигляд послідовності інструкцій у текстовому форматі.

### Як запустити на виконання програму мовою Python?

Для запуску програми мовою Python, використовуйте команду python назва\_файлу.py в командному рядку (терміналі).

### Які коментарі підтримує Python?

Python підтримує однорядкові коментарі, починаючи з символу #.

### Які типи даних підтримує Python?

Python підтримує різні типи даних, включаючи цілі числа, дійсні числа, рядки, списки, кортежі, словники та інші.

### Як оголосити змінну?

Змінну можна оголосити, просто вказавши її ім'я і призначивши їй значення. Наприклад, змінна = 10.

### Які керуючі конструкції підтримує Python?

Python підтримує керуючі конструкції, такі як умовні оператори (if, else, elif), цикли (for, while), та інші.

### Які операції підтримує Python?

Python підтримує операції додавання, віднімання, множення, ділення, порівняння, та багато інших.

### Як здійснити ввід з консолі?

Для введення даних з консолі використовується функція input().

### Як здійснити вивід у консоль?

Для виводу даних у консоль використовується функція print().

### Як здійснити приведення типів?

Для приведення типів використовується функція int(), float(), str(), тощо.

## Висновок

Було створено програму мовою Python, яка генерує зубчатий список, містить лише заштриховані області квадратної матриці, розмір і символ-заповнювач яких вводяться з клавіатури. Програма також має коректні перевірки введених даних та виводить сформований масив на екран. Код програми супроводжується коментарями для зрозумілості.