Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій

Кафедра ІСМ



**Звіт**

до лабораторної роботи № 4

з дисципліни

​*Спеціалізовані мови програмування*

на тему:

“ Розробка ASCII ART генератора для візуалізації 2D-фігур ”

Виконав:  
 студент РІ-31

Яворський О. В.

Прийняв:  
к.т.н., доцент  
 Щербак С.С.

Львів – 2024

**Мета роботи:** Створення Генератора ASCII-арту без використання зовнішніх бібліотек

**Завдання лабораторної роботи**

Завдання 1: Введення користувача

Створіть програму Python, яка отримує введення користувача щодо слова або фрази, яку вони хочуть перетворити в ASCII-арт.

Завдання 2: Набір символів

Визначте набір символів (наприклад, '@', '#', '\*', тощо), які будуть використовуватися для створення ASCII-арту. Ці символи будуть відображати різні відтінки.

Завдання 3: Розміри Art-у

Запитайте у користувача розміри (ширина і висота) ASCII-арту, який вони хочуть створити. Переконайтеся, що розміри в межах керованого діапазону

Завдання 4: Функція генерації Art-у

Напишіть функцію, яка генерує ASCII-арт на основі введення користувача, набору символів та розмірів. Використовуйте введення користувача, щоб визначити, які символи використовувати для кожної позиції в Art-у.

Завдання 5: Вирівнювання тексту

Реалізуйте опції вирівнювання тексту (ліво, центр, право), щоб користувачі могли вибирати, як їх ASCII-арт розміщується на екрані.

Завдання 6: Відображення мистецтва

Відобразіть створений ASCII-арт на екрані за допомогою стандартних функцій друку Python.

Завдання 7: Збереження у файл

Додайте можливість зберігати створений ASCII-арт у текстовий файл, щоб користувачі могли легко завантажувати та обмінюватися своїми творіннями.

Завдання 8: Варіанти кольорів

Дозвольте користувачам вибирати опції кольорів (чорно-білий, відтінки сірого) для свого ASCII-арту.

Завдання 9: Функція попереднього перегляду

Реалізуйте функцію попереднього перегляду, яка показує користувачам попередній перегляд їх ASCII-арту перед остаточним збереженням

Завдання 10: Інтерфейс, зрозумілий для користувача

Створіть інтерфейс для користувача у командному рядку, щоб зробити програму легкою та інтуїтивно зрозумілою для використання.

Реалізація програми:

import pyfiglet

# ANSI color codes

COLOR\_CODES = {

    "red": "\033[31m",

    "green": "\033[32m",

    "yellow": "\033[33m",

    "blue": "\033[34m",

    "purple": "\033[35m",

    "cyan": "\033[36m",

    "white": "\033[37m",

    "gray": "\033[90m",

    "reset": "\033[0m"  # Resets color back to default

}

def get\_user\_input():

    text = input("Enter a word or phrase to convert into ASCII art: ")

    characters = input("Enter a set of characters for ASCII art background (e.g., @#\*): ")

    width = int(input("Enter the width of ASCII art (e.g., 40): "))

    height = int(input("Enter the height of ASCII art (e.g., 10): "))

    alignment = input("Enter alignment (left, center, right): ").lower()

    color\_option = input("Choose color option (black and white, grayscale, or a color: red, green, yellow, blue, purple, cyan, white, gray): ").lower()

    font = input("Choose a font for your ASCII art (e.g., 'slant', 'block', 'standard'): ").lower()

    return text, characters, width, height, alignment, color\_option, font

def validate\_dimensions(width, height, max\_width=100, max\_height=40):

    if width > max\_width or height > max\_height:

        print(f"Dimensions exceed the limit {max\_width}x{max\_height}. Please use smaller sizes.")

        return False

    return True

def generate\_ascii\_art(text, font):

    # Generate ASCII art text using pyfiglet with the selected font

    figlet = pyfiglet.Figlet(font=font)

    ascii\_art = figlet.renderText(text)

    return ascii\_art

def display\_ascii\_art(ascii\_art):

    print("\nYour ASCII Art:\n")

    print(ascii\_art)

def save\_to\_file(ascii\_art, filename="ascii\_art.txt"):

    with open(filename, "w") as file:

        file.write(ascii\_art)

    print(f"ASCII art has been saved to {filename}.")

def apply\_color(ascii\_art, color\_option):

    # Apply color if it's a valid color option

    color\_code = COLOR\_CODES.get(color\_option, COLOR\_CODES["reset"])

    colored\_ascii\_art = f"{color\_code}{ascii\_art}{COLOR\_CODES['reset']}"

    return colored\_ascii\_art

def preview\_ascii\_art(ascii\_art):

    print("Preview of ASCII Art:\n")

    print(ascii\_art)

    confirm = input("Do you want to save this? (yes/no): ")

    return confirm.lower() == 'yes'

def main():

    text, characters, width, height, alignment, color\_option, font = get\_user\_input()

    if not validate\_dimensions(width, height):

        return

    ascii\_art = generate\_ascii\_art(text, font)

    ascii\_art = apply\_color(ascii\_art, color\_option)

    if preview\_ascii\_art(ascii\_art):

        save\_to\_file(ascii\_art)

    else:

        print("Save canceled.")

    display\_ascii\_art(ascii\_art)

if \_\_name\_\_ == "\_\_main\_\_":

    main()

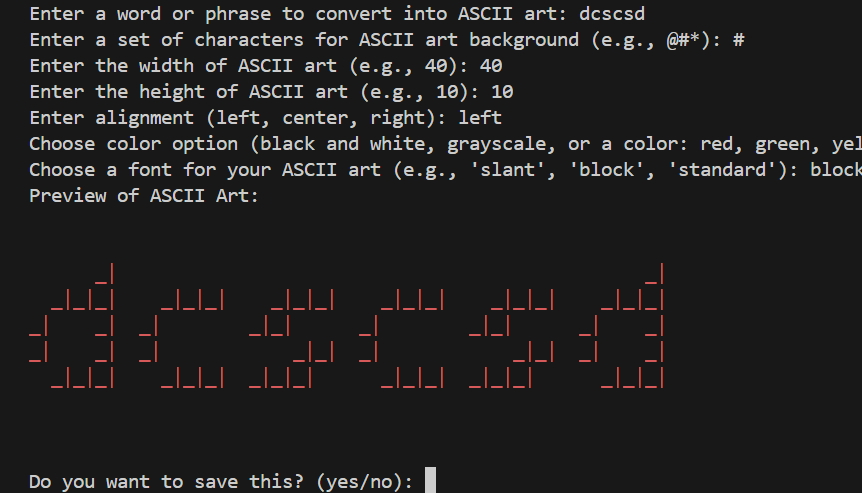


Рис. 1 (Результат роботи програми)

**Висновок:** Виконуючи цю лабораторну роботу ми створили генератор ASCII-арту з нуля, та надали можливість налаштовувати символи, розміри, вирівнювання та кольори, що дозволило нам глибше розібратися як створюється ASCII-арт