МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

**Лабораторна робота №4**

**з дисципліни**

**СПЕЦІАЛІЗОВАНІ МОВИ ПРОГРАМУВАННЯ**

**на тему**

**«Розробка ASCII ART генератора для візуалізації 2D-фігур»**

Виконав:

ст. гр. ІТ-21сп

Одноріг Д.І.

Прийняв:

Щербак С.С.

Львів-2023

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

**Хід 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.

Код програми:

**main.py:**

from modernDraftsman import ModernDraftsman

font = {

'm': '1',

'e':'2',

's':'3'

}

print("Enter text:")

art = ModernDraftsman(input())

print("Choose justify:")

print("- left")

print("- center")

print("- right")

justifyChoice = input()

if justifyChoice == "left":

art.justify = 'left'

art.create()

elif justifyChoice == "center":

art.justify = 'center'

art.create()

elif justifyChoice == "right":

art.justify = 'right'

art.create()

print(art)

**modernDraftsman:**

from Lab3.draftsman import Draftsman

from font import \*

class ModernDraftsman(Draftsman):

\_font =banner3

\_custom\_height = 5

\_custom\_width = 5

\_direction\_text = ""

\_FONTS = {"1": standard, "2": banner3}

def \_custom\_justify(self):

text = self.text

padding = 40 if self.justify == "right" else 20 if self.justify == "center" else 0

if self.justify in ["center", "right"]:

text = text.rjust(padding) if self.justify == "right" else text.center(padding)

return text

def \_create(self, \*\*kwargs):

font = kwargs.get("font", self.\_font)

self.\_direction\_text = self.\_custom\_justify()

art = []

max\_lines = max(len(font.get(letter, "").split("\n")) for letter in self.\_direction\_text)

for line\_num in range(max\_lines):

line = ""

for letter in self.\_direction\_text:

letter\_lines = self.\_font.get(letter, "").split("\n")

line += (letter\_lines[line\_num] if line\_num < len(letter\_lines) else " " \* len(letter\_lines[0])) + " "

justify\_padding = len(letter\_lines[0]) + 1

line = line.ljust(justify\_padding) if self.justify == "left" else line.center(justify\_padding) if self.justify == "center" else line.rjust(justify\_padding)

art.append(line)

return '\n'.join(art)

def \_art\_zoom(self):

return self.\_create(font=banner3)

def prev\_view(self):

art = self.\_color + self.\_create()

return art

def create(self):

try:

self.\_result\_art = self.\_create()

except NameError as e:

raise NameError(f"Error: {e}")

На рис. 1 зображено результат виконання програми.

*Зображення, що містить текст, знімок екрана, Графічний редактор

Автоматично згенерований опис*

*Рис. 1. Результат виконання програми.*

**Висновок:** виконуючи дану лабораторну роботу, я створив генератор ASCII-арту з нуля, та надав можливість налаштовувати символи, розміри, вирівнювання та кольори, що дозволило мені глибше розібратися як створюється ASCII-арт.