Міністерство освіти і науки України

Національний університет «Львівська політехніка»

******

Звіт про виконання лабораторної роботи №4  
з дисципліни  
“Спеціалізовані мови програмування”  
на тему  
“Розробка ASCII ART генератора для візуалізації 2D-фігур”

Виконала:  
студентка групи РІ-31,  
Пятницька Вікторія

Прийняв:

Щербак С.С.

**Мета.** 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:**

***./ascii/ascii\_art.py****:*

import os

from ascii.ascii\_templates import ascii\_templates

class ASCIIArtGenerator:

def \_\_init\_\_(self):

self.text = ""

self.symbol = "\*"

self.generated\_art = ""

self.alignment = "left"

def get\_console\_width(self):

try:

return os.get\_terminal\_size().columns

except OSError:

return 80

def user\_input(self):

return input("Enter text or phrase to convert to ASCII art: ")

def choose\_symbol(self):

symbol = input("Enter a character to create ASCII art (eg. @, #, \*): ").strip()

return symbol if symbol else "\*"

def choose\_alignment(self):

while True:

alignment = input("Select the alignment (left, center, right): ").strip().lower()

if alignment in ["left", "center", "right"]:

return alignment

else:

print("Wrong choice. Please select left, center or right.")

def resize\_ascii\_art(self, art, width\_scale, height\_scale):

lines = art.splitlines()

scaled\_art = []

for line in lines:

scaled\_line = ''.join([char \* width\_scale for char in line])

for \_ in range(height\_scale):

scaled\_art.append(scaled\_line)

return "\n".join(scaled\_art)

def preview\_ascii\_art(self, art):

print("\nASCII art preview:\n")

print(art)

def generate\_ascii\_art(self, text, symbol, width\_scale, height\_scale, alignment):

art\_lines = [""] \* 5

for char in text.upper():

if char in ascii\_templates:

for i in range(5):

line\_with\_symbol = ascii\_templates[char][i].replace('\*', symbol)

art\_lines[i] += line\_with\_symbol + " "

console\_width = self.get\_console\_width()

if alignment == "center":

aligned\_art = [line.center(console\_width) for line in art\_lines]

elif alignment == "right":

aligned\_art = [line.rjust(console\_width) for line in art\_lines]

else: # left

aligned\_art = [line.ljust(console\_width) for line in art\_lines]

final\_art = "\n".join(aligned\_art)

if width\_scale > 1 or height\_scale > 1:

final\_art = self.resize\_ascii\_art(final\_art, width\_scale, height\_scale)

return final\_art

def run(self):

self.text = self.user\_input()

self.symbol = self.choose\_symbol()

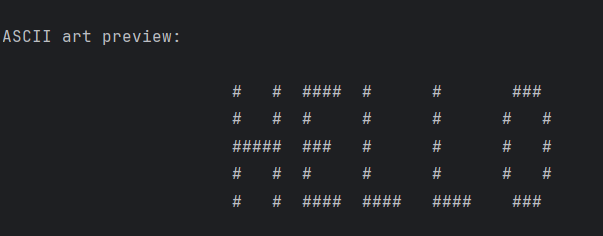
self.alignment = self.choose\_alignment()

width\_scale = int(input("Enter width scale (default 1): ") or 1)

height\_scale = int(input("Enter height scale (default 1): ") or 1)

self.generated\_art = self.generate\_ascii\_art(self.text, self.symbol, width\_scale, height\_scale, self.alignment)

self.preview\_ascii\_art(self.generated\_art)

**

*Рис.1.Приклад виконання програми*

**Висновки:** внаслідок виконання лабораторної роботи я створила генератор ASCII-арту без використання зовнішніх бібліотек

**GitHub:** <https://github.com/viksi01/Calculator/tree/lab4>