МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

ІНСТИТУТ КОМП’ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ



Лабораторна робота №4

з дисципліни “Спеціалізовані мови програмування”

на тему

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

Виконав:

студент групи РІ-32

Тарас ЛИН

Прийняв:

к.т.н.,

доц. кафедри ІСМ

Сергій ЩЕРБАК

Львів-2023

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

**План 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.

Програмний код:

from asciiAlphabet import Alphabet

class ASCIIArt:

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

self.filename = "art.txt"

self.text = ""

self.symbols = ""

self.height = 0

self.width = 0

self.alignment = "left"

self.color = ""

def generate\_art(self):

art\_lines = []

symbol\_index = 0

for i in range(max(self.height, len(Alphabet["A"]))):

art\_line = ""

for char in self.text.upper():

if char in Alphabet:

line = Alphabet[char][i % len(Alphabet[char])]

new\_line = ""

for ch in line:

if ch == "#":

new\_line += self.symbols[symbol\_index % len(self.symbols)]

symbol\_index += 1

else:

new\_line += ch

art\_line += new\_line

else:

art\_line += " " \* self.width

if self.alignment == "center":

art\_line = art\_line.center(

max(self.width, len(Alphabet["A"])) \* len(self.text)

)

elif self.alignment == "right":

art\_line = art\_line.rjust(

max(self.width, len(Alphabet["A"])) \* len(self.text)

)

art\_lines.append(art\_line)

return "\n".join(art\_lines)

def display\_art(self):

color\_code = ''

if self.color.lower() == 'white':

color\_code = '\u001b[37;1m'

elif self.color.lower() == 'black':

color\_code = '\u001b[30m'

elif self.color.lower() == 'gray':

color\_code = '\u001b[90m'

colored\_art = f"{color\_code}{self.generate\_art()}\u001b[0m"

print(colored\_art)

def save\_art(self):

with open(self.filename, "w") as file:

file.write(self.generate\_art())

def preview\_art(self):

print("Preview:")

self.display\_art()

def user\_input(self):

print("Welcome to ASCII art generator!")

self.text = input("Enter the text: ")

self.symbols = input("Enter the symbol: ")

self.height = int(input("Enter the height: "))

self.width = int(input("Enter the width: "))

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

if self.alignment not in ["left", "center", "right"]:

print("Invalid alignment. Defaulting to left.")

self.alignment = "left"

self.color = input("Enter the color(white and black or gray): ")

def run\_art(self):

self.user\_input()

self.display\_art()

if input("Do you want to save the art? (yes/no): ").lower() == "yes":

self.save\_art()

print(f"Art saved to {self.filename}")

self.preview\_art()

def main():

running\_prog = ASCIIArt()

running\_prog.run\_art()

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

main()

Висновок: Виконуючи ці завдання, я створив генератор ASCII-арту з нуля, та надала можливість налаштовувати символи, розміри, вирівнювання та кольори, що дозволило глибше розібратися як створюється ASCII-арт