МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

ІНСТИТУТ КОМП’ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА  ІСМ



ЗВІТ

про виконання лабораторної роботи №3

з дисципліни

«Спеціалізовані мови програмування»

студента групи ІТ-32

Ткачишина Юрія

Прийняв Щербак С. С.

Львів - 2023

**Мета роботи:** Створення додатка Генератора ASCII-арту.

**Індивідуальне завдання**

Завдання 1: Введення користувача Створіть Python-програму, яка приймає введення користувача для слова або фрази, яку треба перетворити в ASCII-арт.

Завдання 2: Бібліотека ASCII-арту Інтегруйте бібліотеку ASCII-арту (наприклад, pyfiglet або art) у вашу програму для генерації ASCII-арту з введення користувача

Завдання 3: Вибір шрифту Дозвольте користувачам вибирати різні стилі шрифтів для свого ASCII-арту. Надайте список доступних шрифтів та дозвольте їм вибрати один.

Завдання 4: Колір тексту Реалізуйте опцію вибору користувачем кольору тексту для їхнього ASCII-арту. Підтримуйте основний вибір кольорів (наприклад, червоний, синій, зелений).

Завдання 5: Форматування виводу Переконайтеся, що створений ASCII-арт правильно відформатований та вирівнюється на екрані для зручності читання.

Завдання 6: Збереження у файл Додайте функціональність для збереження створеного ASCII-арту у текстовому файлі, щоб користувачі могли легко завантажувати та обмінюватися своїми творіннями.

Завдання 7: Розмір ARTу Дозвольте користувачам вказувати розмір (ширина і висота) ASCII-арту, який вони хочуть створити. Масштабуйте текст відповідно.

Завдання 8: Вибір символів Дозвольте користувачам вибирати символи, які вони хочуть використовувати для створення ASCII-арту (наприклад, '@', '#', '\*', тощо).

Завдання 9: Функція попереднього перегляду Реалізуйте функцію попереднього перегляду, яка показує користувачам попередній перегляд їхнього ASCII-арту перед остаточним збереженням.

Завдання 10: Інтерфейс, зрозумілий для користувача Створіть зручний для користувача інтерфейс командного рядку для додатка, щоб зробити його інтуїтивно зрозумілим та легким у використанні.

**Хід виконання:**

Код:  
from colorama import Fore, init

import pyfiglet

import re

class ASCIIGenerator:

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

self.text = ""

self.font = "standard"

self.color = "white"

self.width = 80

self.height = 25

self.custom\_characters = ".,#@$%&\*"

def get\_user\_input(self):

print("Ласкаво просимо до Генератора ASCII-арту!")

self.text = input("Введіть текст, який ви хочете перетворити в ASCII-арт: ")

# Input validation for the font

while True:

self.font = input("Виберіть шрифт (наприклад: standard, banner, big, slant): ")

valid\_fonts = ["standard", "banner", "big", "slant"]

if self.font in valid\_fonts:

break

else:

print("Недійсний шрифт. Будь ласка, виберіть один із наступних: standard, banner, big, slant")

# Input validation for the color

while True:

self.color = input("Виберіть колір тексту (наприклад: red, blue, green, white): ")

valid\_colors = ["red", "blue", "green", "white"]

if self.color in valid\_colors:

break

else:

print("Недійсний колір. Будь ласка, виберіть один із наступних: red, blue, green, white")

# Input validation for the width

while True:

try:

self.width = int(input("Введіть ширину ASCII-арту (за замовчуванням 80): "))

if self.width > 0:

break

else:

print("Ширина повинна бути додатнім числом.")

except ValueError:

print("Недійсне значення ширини. Введіть додатне ціле число.")

# Input validation for the height

while True:

try:

self.height = int(input("Введіть висоту ASCII-арту (за замовчуванням 25): "))

if self.height > 0:

break

else:

print("Висота повинна бути додатнім числом.")

except ValueError:

print("Недійсне значення висоти. Введіть додатне ціле число.")

# Input validation for custom characters

while True:

self.custom\_characters = input("Введіть символи для створення ASCII-арту (наприклад: .,#@$%&\*): ")

valid\_characters = re.compile(r'^[\w\s]+$')

if valid\_characters.match(self.custom\_characters):

break

else:

print("Недійсні символи. Будь ласка, використовуйте букви, цифри, пробіли та спеціальні символи.")

def generate\_ascii\_art(self):

try:

ascii\_art = pyfiglet.Figlet(font=self.font)

result = ascii\_art.renderText(self.text)

return result

except pyfiglet.FigletError as e:

return f"Помилка: {str(e)}"

def resize\_ascii\_art(self, ascii\_art):

lines = ascii\_art.split('\n')

resized\_lines = []

for line in lines:

resized\_line = ""

for char in line:

if char.isalnum() or char.isspace():

resized\_line += char

else:

resized\_line += self.custom\_characters[0]

resized\_lines.append(resized\_line.ljust(self.width)[:self.width])

resized\_lines.extend([' ' \* self.width] \* (self.height - len(resized\_lines)))

return '\n'.join(resized\_lines)

def display\_ascii\_art(self, ascii\_art, color=None):

if color:

init(autoreset=True)

colored\_art = getattr(Fore, self.color.upper()) + self.resize\_ascii\_art(ascii\_art)

print(colored\_art)

else:

print(self.resize\_ascii\_art(ascii\_art))

def save\_ascii\_art(self, filename):

ascii\_art = self.generate\_ascii\_art()

filename\_with\_extension = filename + ".txt"

with open(filename\_with\_extension, 'w') as file:

file.write(self.resize\_ascii\_art(ascii\_art))

def preview\_ascii\_art(self):

print("Попередній перегляд вашого ASCII-арту:")

ascii\_art = self.generate\_ascii\_art()

self.display\_ascii\_art(ascii\_art, self.color)

def run(self):

self.get\_user\_input()

self.preview\_ascii\_art()

save\_option = input("Зберегти ASCII-арт у файл? (y/n): ")

if save\_option.lower() == 'y':

filename = input("Введіть ім'я файлу для збереження (без типу розширення): ")

self.save\_ascii\_art(filename)

print(f"ASCII-арт збережено у файлі: {filename}.txt")

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

generator = ASCIIGenerator()

generator.run()

**Висновок:** Виконавши ці завдання, я створив додаток генератора ASCII-арту.