Міністерство освіти і науки України  
Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій

Кафедра інформаційних систем та мереж

******

**ЗВІТ**

**ДО ЛАБОРАТОРНОЇ РОБОТИ №3  
з дисципліни  
«Спеціалізовані мови програмування»  
на тему:**  
«Розробка ACII ART генератор для візуалізації текстових даних»

***Виконала:*** *студентка групи РІ-31  
 Діана ВІЙЧУК*

***Прийняв:****к. т. н., доцент* *Сергій ЩЕРБАК*

## МЕТА РОБОТИ

Створення додатка Генератора ACII-арту.

## 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.

## ПРОГРАМНА РЕАЛІЗАЦІЯ

Файл *runner.py* .

import sys

import os

lab3\_root = os.path.abspath(os.path.join(os.path.dirname(\_\_file\_\_), "..", ".."))

sys.path.append(lab3\_root)

from src.main import main

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

main()== "\_\_main\_\_":

main()

Файл */src/bll/acii\_art\_generator.py* .

import pyfiglet

from termcolor import colored

def scale\_ascii\_art(ascii\_art, width\_factor, height\_factor):

scaled\_lines = []

for line in ascii\_art.splitlines():

scaled\_line = "".join(char \* width\_factor for char in line)

for \_ in range(height\_factor):

scaled\_lines.append(scaled\_line)

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

def generate\_ascii\_art(text, font="standard", symbol="#"):

ascii\_art = pyfiglet.figlet\_format(text, font=font)

ascii\_art = ascii\_art.replace("#", symbol).replace(".", symbol)

return ascii\_art

def get\_available\_fonts():

return ['tty', 'banner', 'rozzo', 'standard', 'slant', 'big', 'doom', 'alligator', 'digital', 'cybermedium']

def get\_available\_colors():

return ['red', 'green', 'yellow', 'blue', 'magenta', 'cyan', 'white']

Файл */src/ui/art\_unterface.py* .

from termcolor import colored

from src.bll.acii\_art\_generator import generate\_ascii\_art, scale\_ascii\_art, get\_available\_fonts, get\_available\_colors

def get\_user\_input():

return input("Введіть слово або фразу для ASCII-арту: ")

def choose\_font():

fonts = get\_available\_fonts()

print("Доступні шрифти:", fonts)

chosen\_font = input("Виберіть шрифт з наведених: ")

return chosen\_font if chosen\_font in fonts else "standard"

def choose\_color():

colors = get\_available\_colors()

print("Доступні кольори:", colors)

chosen\_color = input("Виберіть колір з наведених: ")

return chosen\_color if chosen\_color in colors else "white"

def preview\_ascii\_art(text, font, color, symbol, width\_factor=1, height\_factor=1):

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

scaled\_art = scale\_ascii\_art(ascii\_art, width\_factor, height\_factor)

colored\_art = colored(scaled\_art, color)

print("\nПопередній перегляд ASCII-арту:\n")

print(colored\_art)

return colored\_art

Файл */src/main.py* .

import sys

import os

lab3\_root = os.path.abspath(os.path.join(os.path.dirname(\_\_file\_\_), "..", ".."))

sys.path.append(lab3\_root)

from src.ui.art\_interface import get\_user\_input, preview\_ascii\_art, choose\_color, choose\_font

from shered.dal.filer import save\_to\_file

from shered.validation import get\_integer\_input

def main():

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

while True:

print("\nВиберіть опцію:")

print("1 - Генератор ASCII-арту")

print("2 - Вихід")

choice = input("Ваш вибір: ")

if choice == '1':

user\_input = get\_user\_input()

chosen\_font = choose\_font()

chosen\_color = choose\_color()

chosen\_symbol = input("Введіть символ для ASCII-арту (за замовчуванням '#'): ") or "#"

width\_factor = get\_integer\_input("Введіть масштаб по ширині (ціле число, 1 для нормального розміру): ")

height\_factor = get\_integer\_input("Введіть масштаб по висоті (ціле число, 1 для нормального розміру): ")

ascii\_art = preview\_ascii\_art(user\_input, chosen\_font, chosen\_color, chosen\_symbol, width\_factor, height\_factor)

if input("Зберегти ASCII-арт? (y/n): ").lower() == 'y':

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

save\_to\_file(ascii\_art, f"{filename}.txt")

elif choice == '2':

print("До побачення!")

break

else:

print("Невірний вибір. Спробуйте ще раз.")

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

main()

## ВИСНОВКИ

На цій лабораторній роботі я створила додаток Генератора ACII-арту.

Посилання на GitHub: <https://github.com/Dinasi4ka/Calculator.git>