мІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

нАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «лЬВІВСЬКА ПОЛІТЕХНІКА»

ІНСТИТУТ КОМП’ЮТЕРНИХ НАУК ТА ІНОРМАЦІЙНИХ ТЕХНОЛОГІЙ

|  |
| --- |
|  |



ЗВІТ

Про виконання лабораторної роботи № 3

З дисципліни

«Спеціалізовані мови програмування»

Виконала:

студентка гр. ІТ-31

Катерина ПОПОВА

Прийняв:

Сергій ЩЕРБАК

Львів – 2023

**Тема роботи :** Розробка ASCII ART генератора для візуалізації текстових даних

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

**План роботи**

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.

**Код**

import art

from termcolor import colored2

class ArtManager:

def get\_art(self, font\_name, text):

try:

return art.text2art(text, font=font\_name)

except Exception as e:

print(f"Помилка: {e}")

return art.text2art(text, font='block')

class ColorManager:

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

self.colors = {

'червоний': 'red',

'синій': 'blue',

'зелений': 'green'

}

def get\_color(self):

color\_name = input("Виберіть колір тексту (червоний/синій/зелений): ").lower()

return self.colors.get(color\_name, 'white')

class SizeManager:

def get\_size(self):

try:

width = int(input("Введіть ширину ASCII-арту: "))

height = int(input("Введіть висоту ASCII-арту: "))

except ValueError:

width, height = 80, 20

return width, height

class FileManager:

def save\_art(self, formatted\_art):

file\_name = input("Введіть ім'я файлу для збереження: ")

with open(file\_name, 'w') as file:

file.write(formatted\_art)

print(f"ASCII-арт був збережений у файлі '{file\_name}'")

class AsciiArtGenerator:

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

self.art\_manager = ArtManager()

self.color\_manager = ColorManager()

self.size\_manager = SizeManager()

self.file\_manager = FileManager()

def get\_user\_input(self, prompt, default=None):

user\_input = input(prompt).strip()

return user\_input if user\_input else default

def run(self):

text = self.get\_user\_input("Введіть слово або фразу для генерації ASCII-арту: ")

font\_name = self.get\_user\_input("Виберіть шрифт (стандартний/керований/блоки): ", default='block')

art\_object = self.art\_manager.get\_art(font\_name, text)

color = self.color\_manager.get\_color()

width, height = self.size\_manager.get\_size()

char = self.get\_user\_input("Введіть символ, який ви хочете використовувати (наприклад, '@','\*'): ")

colored\_art = colored(art\_object, color)

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

formatted\_art = colored\_art.center(width).replace(' ', char)

print(formatted\_art)

save\_option = input("Зберегти ASCII-арт у файл? (так/ні): ").lower()

if save\_option == 'так':

self.file\_manager.save\_art(formatted\_art)

print("Дякуємо за використання нашого генератора ASCII-арту!")

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

generator = AsciiArtGenerator()

generator.run()

**Результат виконання**

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

**Висновок:** Виконуючи ці завдання, я створила універсальний Генератор ASCII-арту, який дозволить користувачам налаштовувати свої творіння з різними шрифтами, кольорами, розмірами та символами. Проект надає практичний досвід роботи з введенням користувача, зовнішніми бібліотеками, роботою з файлами та дизайном інтерфейсу користувача в Python.