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»

на тему «Розробка ASCII ART генератора для візуалізації текстових даних»

Виконала:

студентка гр. ІТ-31

Проців Р.В.

Прийняв:

доц. каф. ІСМ

Щербак С.С.

Львів – 2023

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

**Хід 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.

**Програмний код**:

import pyfiglet

from termcolor import colored

# Function to get user input

def get\_user\_input():

user\_input = input("Enter a word or phrase: ")

return user\_input

# Function to choose a font for ASCII art

def choose\_font():

available\_fonts = pyfiglet.FigletFont.getFonts()

print("Available fonts:")

for font in available\_fonts:

print(font)

selected\_font = input("Choose a font: ")

return selected\_font

# Function to choose a color for ASCII art

def choose\_color():

available\_colors = ["red", "blue", "green", "yellow"]

print("Available colors:", available\_colors)

selected\_color = input("Choose a color: ")

return selected\_color

# Function to choose a symbol for customizing ASCII art

def choose\_symbol():

symbol = input("Enter a character to use for ASCII art: ")

return symbol

# Function to format ASCII art by replacing '#' with a custom symbol

def format\_ascii\_art(art, symbol):

return art.replace('#', symbol)

# Function to resize ASCII art to a specified width and height

def resize\_ascii\_art(ascii\_art, width, height):

# Split the ASCII art into lines

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

# Resize the ASCII art to the specified height

while len(lines) < height:

lines.append(' ' \* len(lines[0]))

# Resize each line to the specified width

for i in range(len(lines)):

lines[i] = lines[i].ljust(width)[:width]

return '\n'.join(lines)

# Function to save ASCII art to a file

def save\_to\_file(ascii\_art):

filename = input("Enter the filename to save the ASCII art: ")

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

file.write(ascii\_art)

# Main function

def main():

user\_input = get\_user\_input()

selected\_font = choose\_font()

selected\_color = choose\_color()

symbol = choose\_symbol()

# Generate ASCII art using pyfiglet and color it

ascii\_art = pyfiglet.figlet\_format(user\_input, font=selected\_font)

colored\_art = colored(ascii\_art, color=selected\_color)

# Format the ASCII art with the chosen symbol

formatted\_art = format\_ascii\_art(colored\_art, symbol)

print(formatted\_art)

# Get user input for desired width and height

width = int(input("Enter the desired width (number of columns): "))

height = int(input("Enter the desired height (number of rows): "))

# Resize the ASCII art

resized\_art = resize\_ascii\_art(formatted\_art, width, height)

print(resized\_art)

# Ask the user if they want to save the ASCII art to a file

save\_option = input("Do you want to save this ASCII art to a file? (yes/no): ")

if save\_option.lower() == "yes":

save\_to\_file(resized\_art)

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

main()

**GitHub Repository:** <https://github.com/fxxwol/python/blob/main/lab_3.py>

**Висновок:** в результаті виконання даної лабораторної роботи створено універсальний Генератор ASCII-арту, який дозволяє користувачам налаштовувати свої творіння з різними шрифтами, кольорами, розмірами та символами, отримано практичний досвід роботи з введенням користувача, зовнішніми бібліотеками, роботою з файлами та дизайном інтерфейсу користувача в Python.