мІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

нАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «лЬВІВСЬКА ПОЛІТЕХНІКА»

ІНСТИТУТ КОМП’ЮТЕРНИХ НАУК ТА ІНОРМАЦІЙНИХ ТЕХНОЛОГІЙ

|  |
| --- |
|  |



ЗВІТ

Про виконання лабораторної роботи № 4

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 2D-фігур

**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.

**Код**

**main.py**

from symbol import ascii\_chars

from termcolor import colored

import os

import sys

from ASCIIArt import ASCIIArt

class ASCIIArtApp:

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

self.run()

def run(self):

while True:

inp\_text = input("Type text: ")

width = int(input("Type width: "))

height = int(input("Type height: "))

print("Color:\n1. White\n2. Red")

selected\_color = input("Choose color by number: ")

art = ASCIIArt(inp\_text, width, height, selected\_color)

ascii\_art = art.generate()

print(ascii\_art)

preview\_response = input("Want to preview saved ASCII art? (1 - yes, 0 - no): ")

if preview\_response == '1':

art.save(ascii\_art)

restart\_response = input("Do you want to create a new ASCII art? (1 - Yes, 0 - No): ")

if restart\_response != '1':

break

os.system('cls' if os.name == 'nt' else 'clear')

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

app = ASCIIArtApp()

**ASCIIArt.py**

from symbol import ascii\_chars

from termcolor import colored

import os

import sys

class ASCIIArt:

def \_\_init\_\_(self, text, width, height, color):

self.text = text.upper()

self.width = width

self.height = height

self.color = self.choose\_color(color)

def choose\_color(self, color):

return 'white' if color == '1' else 'red' if color == '2' else 'white'

def generate(self):

max\_lines = max(len(ascii\_chars['A']), \*map(len, ascii\_chars.values()))

art = [""] \* max\_lines

for char in self.text:

char\_data = ascii\_chars.get(char, ascii\_chars.get(' ', [' ' \* len(ascii\_chars['A'])] \* len(ascii\_chars['A'])))

for i, line in enumerate(char\_data):

art[i] += colored(line[:self.width] + " " \* (self.width - len(line[:self.width])), self.color)

if len(art) < self.height:

art.extend([colored(' ' \* self.width, self.color)] \* (self.height - len(art)))

return "\n".join(art[:self.height])

def save(self, ascii\_art):

with open('figure.txt', 'w') as file:

file.write(ascii\_art)

print("ASCII-арт збережено у файл 'figure.txt'.")

**Результат виконання**

Изображение выглядит как текст, снимок экрана, часы

Автоматически созданное описание

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