МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра інформаційних систем та мереж

Лабораторна робота №3

з дисципліни

СПЕЦІАЛІЗОВАНІ МОВИ ПРОГРАМУВАННЯ

на тему

РОЗРОБКА ASCII ART ГЕНЕРАТОРА ДЛЯ ВІЗУАЛІЗАЦІЇ   
ТЕКСТОВИХ ДАНИХ

Виконав:

ст. гр. РІ-31

ЛАЗАР В.С.

Прийняв:

ЩЕРБАК С.С.

Львів-2024

**Мета роботи:**

Створення додатка Генератора ASCII-арту.

**Хід роботи:**

**Завдання 1: Введення користувача**

Створити Python-програму, яка приймає введення користувача для слова або фрази, яку треба перетворити в ASCII-арт.

**Завдання 2: Бібліотека ASCII-арту**

Інтегрувати бібліотеку ASCII-арту (наприклад, pyfiglet або art) у програму для генерації ASCII-арту з введення користувача

**Завдання 3: Вибір шрифту**

Дозволити користувачам вибирати різні стилі шрифтів для свого ASCII-арту. Надати список доступних шрифтів та дозволити їм вибрати один.

**Завдання 4: Колір тексту**

Реалізувати опцію вибору користувачем кольору тексту для їхнього ASCII-арту. Підтримувати основний вибір кольорів (наприклад, червоний, синій, зелений).

**Завдання 5: Форматування виводу**

Переконатися, що створений ASCII-арт правильно відформатований та вирівнюється на екрані для зручності читання.

**Завдання 6: Збереження у файл**

Додати функціональність для збереження створеного ASCII-арту у текстовому файлі, щоб користувачі могли легко завантажувати та обмінюватися своїми творіннями.

**Завдання 7: Розмір ARTу**

Дозволити користувачам вказувати розмір (ширина і висота) ASCII-арту, який вони хочуть створити. Масштабувати текст відповідно.

**Завдання 8: Вибір символів**

Дозволити користувачам вибирати символи, які вони хочуть використовувати для створення ASCII-арту (наприклад, '@', '#', '\*', тощо).

**Завдання 9: Функція попереднього перегляду**

Реалізувати функцію попереднього перегляду, яка показує користувачам попередній перегляд їхнього ASCII-арту перед остаточним збереженням.

**Завдання 10: Інтерфейс, зрозумілий для користувача**

Створити зручний для користувача інтерфейс командного рядку для додатка, щоб зробити його інтуїтивно зрозумілим та легким у використанні.

**Код програми:**

import random

import textwrap

import GlobalVariables as Global

from BLL.classes.ascii import Ascii

from pyfiglet import FigletFont, figlet\_format

import DAL.functions.upload\_to\_file as file\_upload

class Console:

@staticmethod

def prompt():

Ascii.print("ASCIIFY", True)

while True:

prompt = input("1 - Enter text\n"

"2 - Select font automatically\n"

"3 - Change font\n"

"4 - Current font\n"

"5 - Change width and height\n"

"6 - Change color\n"

"Your choice: ")

match prompt:

case "1":

Console.enter\_text()

case "2":

Console.auto\_font()

case "3":

Console.change\_font()

case "4":

print("Current font: " + Global.font)

case "5":

Console.change\_width\_and\_height()

case "6":

Console.change\_color()

case \_:

return

@staticmethod

def enter\_text():

text = input("Enter text: ")

ftext = Ascii.print(text)

save\_prompt = input("Do you want to save the text? (y/n): ").lower()

if save\_prompt == "y":

while True:

file\_name = input("Enter file name: ")

if file\_name.strip() != "":

if not file\_name.endswith(".txt"):

file\_name += ".txt"

try:

file\_upload.write(ftext, file\_name)

print("Text was uploaded successfully")

break

except IOError:

print("An error occurred during file upload, please try again")

else:

print("Please enter a valid file name")

@staticmethod

def auto\_font():

text = input("Enter text: ")

symbols = input("Enter a set of characters that should be in the ASCII art: ")

font\_symbols = set(symbols) | {" ", "\n"}

fonts = FigletFont.getFonts()

fonts.remove('mshebrew210')

random.shuffle(fonts)

for font in fonts:

random\_art = figlet\_format(text, font=font, width=Global.width)

random\_art\_chars = set(random\_art)

if all(char in font\_symbols for char in random\_art\_chars):

print("Found font:" + font)

Global.font = font

Ascii.print(text)

return

print("No fonts were found, please try again with a wider set of characters")

@staticmethod

def change\_font():

new\_font = input("Enter the new font you want to choose\n"

"You can also use 'font' to see all fonts available or 'random' to choose a random font\n"

"Your choice: ")

if new\_font in FigletFont.getFonts():

Global.font = new\_font

print("Font changed successfully")

elif new\_font.lower() == "font":

print("Available fonts:\n" + textwrap.fill(", ".join(FigletFont.getFonts()), width=Global.width))

elif new\_font.lower() == "random":

Global.font = random.choice(FigletFont.getFonts())

print("Randomly selected font: " + Global.font)

else:

print("Invalid font")

@staticmethod

def change\_width\_and\_height():

while True:

width\_prompt = input("Enter the width of an ASCII art\n"

"(any non-positive value will reset it to default values\n"

"Your choice: ")

try:

width = int(width\_prompt)

Global.width = Ascii.verify\_width(width)

print("Width changed successfully")

except ValueError:

print("Please enter an integer")

continue

height\_prompt = input("Enter the height of an ASCII art\n"

"(any non-positive value will reset it to default values\n"

"Your choice: ")

try:

height = int(height\_prompt)

Global.height = height

print("Height changed successfully")

break

except ValueError:

print("Please enter an integer")

continue

@staticmethod

def change\_color():

color\_prompt = input("Enter the color of your ASCII art:\n"

"1 - Red\n"

"2 - Green\n"

"3 - Yellow\n"

"4 - Blue\n"

"5 - Magenta\n"

"6 - Cyan\n"

"7 - Light gray\n"

"0 - Default\n"

"Your choice: ")

match color\_prompt:

case "1":

Global.color = "\033[31m"

case "2":

Global.color = "\033[32m"

case "3":

Global.color = "\033[33m"

case "4":

Global.color = "\033[34m"

case "5":

Global.color = "\033[35m"

case "6":

Global.color = "\033[36m"

case "7":

Global.color = "\033[37m"

case "0":

Global.color = "\033[39m"

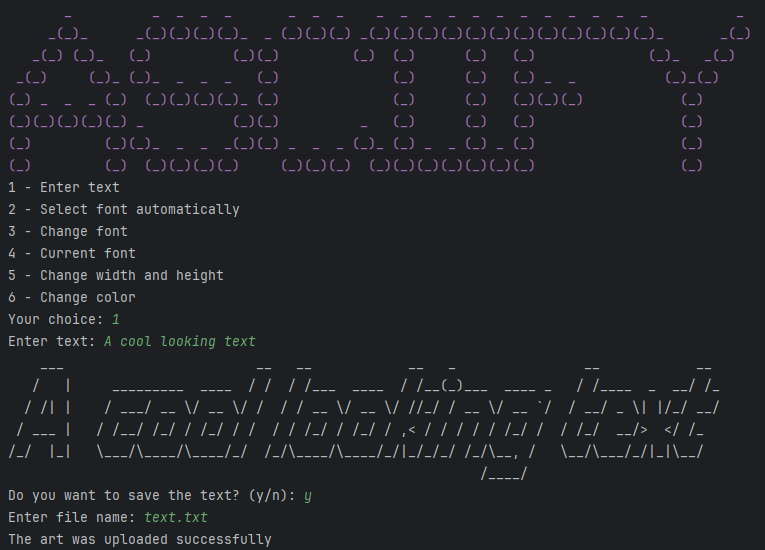
case \_:

print("Invalid color choice, please try again.")

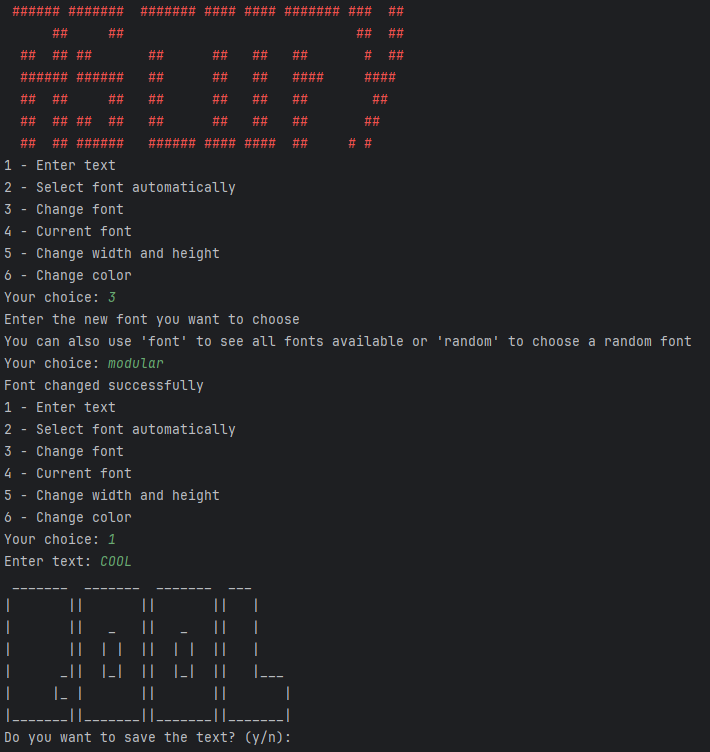
return

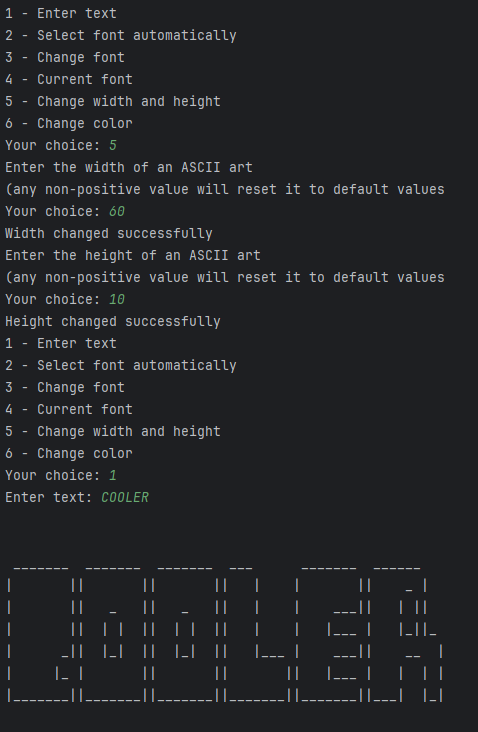
print("Color changed successfully")

На рис. 1-5 зображено результат виконання програми:

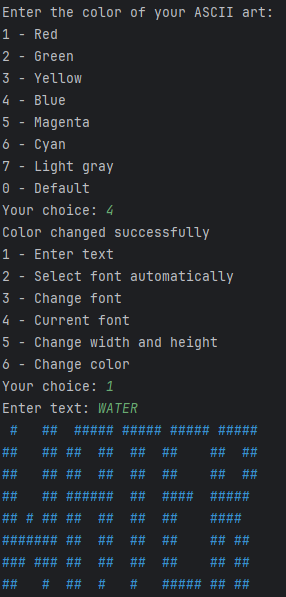


*Рис. 1. Приклад генерації та збереження ASCII-арту*

*  
Рис. 2. Приклад зміни шрифту ASCII-арту*



*Рис. 3. Приклад зміни ширини та висоти арту*



*Рис. 4. Приклад зміни кольору арту*



*Рис. 5. Приклад автоматичного підбору шрифту за вказаними символами*

Посилання на Github: [PaperGlit/Python\_Lab\_3](https://github.com/PaperGlit/Python_Lab_3)

**Висновок:**

Виконавши ці завдання, я створив універсальний Генератор ASCII-арту, який дозволить користувачам налаштовувати свої творіння з різними шрифтами, кольорами, розмірами та символами. Проект надав мені практичний досвід роботи з введенням користувача, зовнішніми бібліотеками, роботою з файлами та дизайном інтерфейсу користувача в Python.