МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра інформаційних систем та мереж

Лабораторна робота №4

з дисципліни

СПЕЦІАЛІЗОВАНІ МОВИ ПРОГРАМУВАННЯ

на тему

РОЗРОБКА ASCII ART ГЕНЕРАТОРА ДЛЯ ВІЗУАЛІЗАЦІЇ 2D-ФІГУР

Виконав:

ст. гр. РІ-31

ЛАЗАР В.С.

Прийняв:

ЩЕРБАК С.С.

Львів-2024

**Мета роботи:**

Створення Генератора ASCII-арту без використання зовнішніх бібліотек

**Хід роботи:**

**Завдання 1: Введення користувача**

Створити програму в Python, яка отримує введення користувача щодо слова або фрази, яку вони хочуть перетворити в ASCII-арт.

**Завдання 2: Набір символів**

Визначити набір символів (наприклад, '@', '#', '\*', тощо), які будуть використовуватися для створення ASCII-арту. Ці символи будуть відображати різні відтінки.

**Завдання 3: Розміри Art-у**

Запитати у користувача розміри (ширина і висота) ASCII-арту, який вони хочуть створити. Переконатися, що розміри в межах керованого діапазону.

**Завдання 4: Функція генерації Art-у**

Написати функцію, яка генерує ASCII-арт на основі введення користувача, набору символів та розмірів. Використовувати введення користувача, щоб визначити, які символи використовувати для кожної позиції в Art-у.

**Завдання 5: Вирівнювання тексту**

Реалізувати опції вирівнювання тексту (ліво, центр, право), щоб користувачі могли вибирати, як їх ASCII-арт розміщується на екрані.

**Завдання 6: Відображення мистецтва**

Відобразити створений ASCII-арт на екрані за допомогою стандартних функцій друку Python.

**Завдання 7: Збереження у файл**

Додати можливість зберігати створений ASCII-арт у текстовий файл, щоб користувачі могли легко завантажувати та обмінюватися своїми творіннями.

**Завдання 8: Варіанти кольорів**

Дозволити користувачам вибирати опції кольорів (чорно-білий, відтінки сірого) для свого ASCII-арту.

**Завдання 9: Функція попереднього перегляду**

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

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

Створити інтерфейс для користувача у командному рядку, щоб зробити програму легкою та інтуїтивно зрозумілою для використання.

**Код програми:**

from BLL.classes.ascii import Ascii

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

class Console:

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

self.ascii = Ascii("ASCIIFY", color="random")

self.\_\_prompt()

def \_\_prompt(self):

self.ascii.print()

self.ascii.color = "\033[39m"

while True:

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

"2 - Change font's symbols\n"

"3 - Change width and height\n"

"4 - Change color\n"

"5 - Change text orientation\n"

"Your choice: ")

match prompt:

case "1":

self.\_\_enter\_text()

case "2":

self.\_\_change\_symbols()

case "3":

self.\_\_change\_width\_and\_height()

case "4":

self.\_\_change\_color()

case "5":

self.\_\_justify()

case \_:

return

def \_\_enter\_text(self):

text = input("Enter text: ")

self.ascii.text = text

ftext = self.ascii.print()

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

if save\_prompt == "y":

try:

file\_upload.write(ftext)

except IOError:

print("The file could not be uploaded, please try again")

def \_\_change\_symbols(self):

while True:

shadow\_prompt = input("Enter symbol for shadows: ")

if shadow\_prompt.strip() != "" or len(shadow\_prompt) == 1:

self.ascii.shadow = shadow\_prompt

else:

print("Please enter a valid shadows symbol (only one allowed)")

continue

text\_prompt = input("Enter symbol for text: ")

if text\_prompt.strip() != "" or len(text\_prompt) == 1:

self.ascii.text\_s = text\_prompt

else:

print("Please enter a valid text symbol (only one allowed)")

continue

highlight\_prompt = input("Enter symbol for highlights: ")

if highlight\_prompt.strip() != "" or len(highlight\_prompt) == 1:

self.ascii.highlight = highlight\_prompt

else:

print("Please enter a valid highlights symbol (only one allowed)")

continue

break

def \_\_change\_width\_and\_height(self):

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)

self.ascii.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)

self.ascii.height = height

print("Height changed successfully")

break

except ValueError:

print("Please enter an integer")

continue

def \_\_change\_color(self):

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"

"8 - Random\n"

"0 - Default\n"

"Your choice: ")

match color\_prompt:

case "1":

self.ascii.color = "\033[31m"

case "2":

self.ascii.color = "\033[32m"

case "3":

self.ascii.color = "\033[33m"

case "4":

self.ascii.color = "\033[34m"

case "5":

self.ascii.color = "\033[35m"

case "6":

self.ascii.color = "\033[36m"

case "7":

self.ascii.color = "\033[37m"

case "8":

self.ascii.color = "random"

case "0":

self.ascii.color = "\033[39m"

case \_:

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

return

print("Color changed successfully")

def \_\_justify(self):

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

"1 - Left\n"

"2 - Center\n"

"3 - Right\n"

"Your choice: ")

match justify\_prompt:

case "1":

self.ascii.justify = "left"

case "2":

self.ascii.justify = "center"

case "3":

self.ascii.justify = "right"

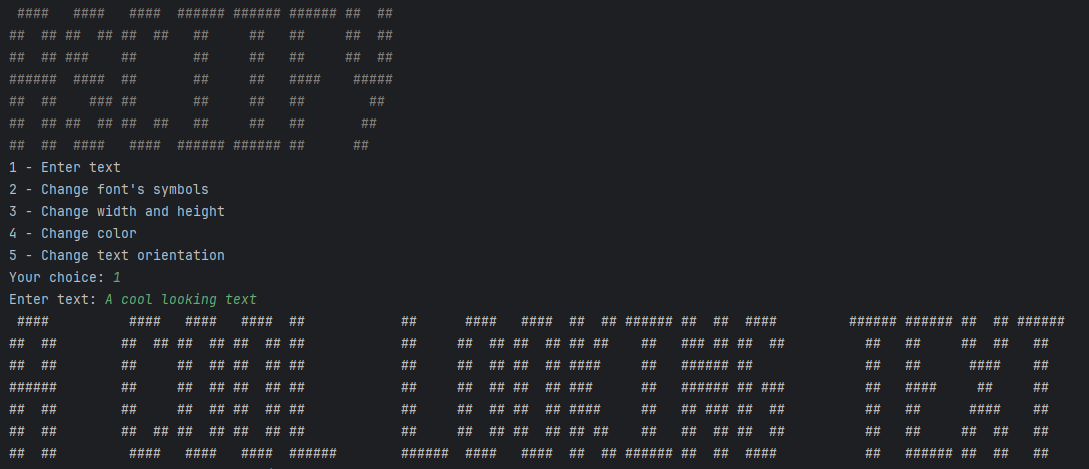
case "\_":

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

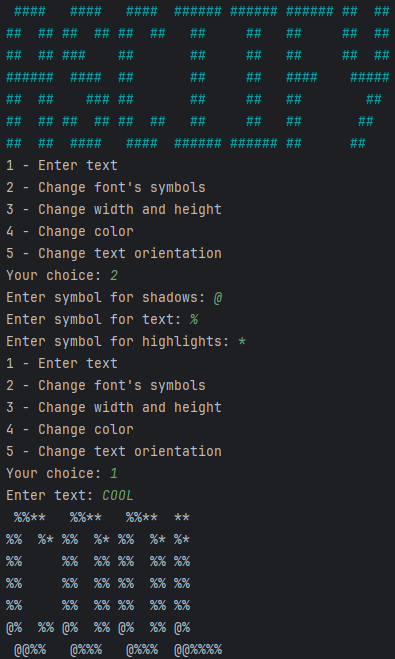
return

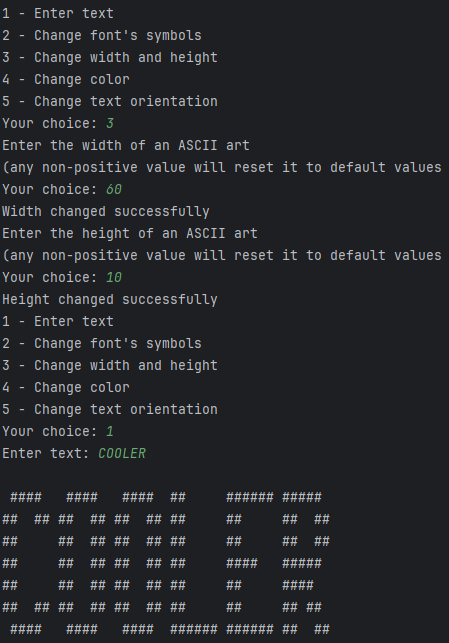
print("Orientation changed successfully")

На рис. 1-5 зображено результат виконання програми:

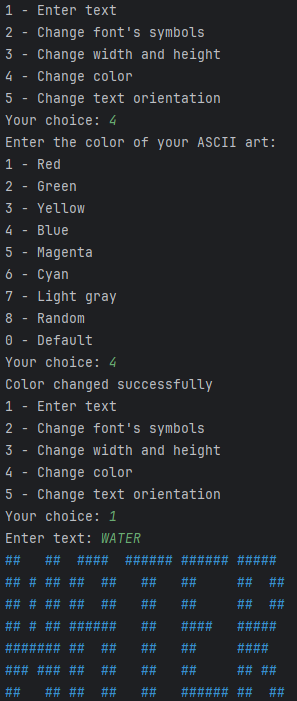


*Рис. 1. Приклад генерації та збереження ASCII-арту*

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



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



*Рис. 4. Приклад зміни кольору арту*



*Рис. 5. Приклад вирівнювання арту*

Посилання на Github: [PaperGlit/Python\_Lab\_4](https://github.com/PaperGlit/Python_Lab_4)

**Висновок:**

Виконуючи ці завдання, я створив генератор ASCII-арту з нуля, та надав можливість налаштовувати символи, розміри, вирівнювання та кольори, що дозволило мені глибше розібратися як створюється ASCII-арт.