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

Виконала:

ст. гр. ІТ-32

Ольга ЧИГИРИК

Прийняв

доцент каф. ІСМ:

Сергій ЩЕРБАК

|  |  |
| --- | --- |
| **Балів** | **Дата** |
|  |  |

Львів-2023

**Мета**: 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.

**Main.py**

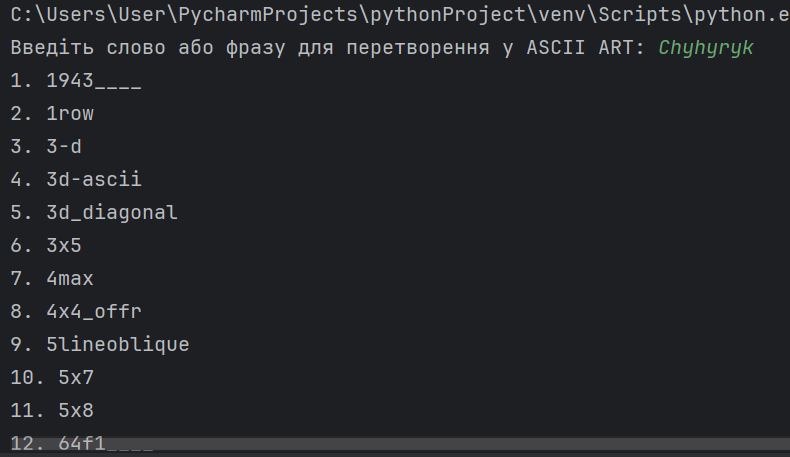
#from console import Console  
#from functions import ASCIIArtGenerator  
from Scripts.all\_python\_labs.shared\_lib.console\_ui import ConsoleUI  
from Scripts.all\_python\_labs.shared\_lib.input\_validator import InputValidator  
  
# def main():  
# console = Console()  
# art\_generator = ASCIIArtGenerator()  
#  
# while True:  
# text = console.get\_input("Введіть слово або фразу для перетворення у ASCII ART: ")  
#  
# if art\_generator.contains\_ukrainian(text):  
# console.print\_message("Будь ласка, змініть розкладку та введіть текст англійською.")  
# continue  
#  
# font = console.choose\_option(art\_generator.available\_fonts(), "Виберіть номер шрифту: ")  
# color = console.choose\_option(art\_generator.available\_colors(), "Виберіть номер кольору: ")  
# width\_multiplier, height\_multiplier = console.choose\_font\_size()  
#  
# ascii\_art = art\_generator.generate(text, font)  
# resized\_art = art\_generator.resize(ascii\_art, width\_multiplier, height\_multiplier)  
# colored\_art = art\_generator.colorize(resized\_art, color)  
#  
# console.print\_colored\_text(colored\_art, color)  
#  
# preview = console.get\_input("Зберегти цей ASCII ART? (y/n) ")  
# if preview.lower() == 'y':  
# art\_generator.save(resized\_art)  
# console.print\_message("ASCII ART збережений у файлі 'ascii\_art.txt'")  
#  
# cont = console.get\_input("Хочете спробувати ще раз? (y/n) ")  
# if cont.lower() != 'y':  
# break  
  
from Scripts.all\_python\_labs.shared\_lib.runnable import Runnable  
  
def run(self):  
 *"""  
 Run the main application loop.  
  
 Returns:  
 - None  
 """* # Run the main application event loop  
 self.root.mainloop()  
  
def main():  
 console\_ui = ConsoleUI()  
 art\_generator = InputValidator()  
  
 while True:  
 text = console\_ui.get\_input("Введіть слово або фразу для перетворення у ASCII ART: ")  
  
 if art\_generator.contains\_ukrainian(text):  
 console\_ui.display\_message("Будь ласка, змініть розкладку та введіть текст англійською.")  
 continue  
  
 font = console\_ui.choose\_option(art\_generator.available\_fonts(), "Виберіть номер шрифту: ")  
 color = console\_ui.choose\_option(art\_generator.available\_colors(), "Виберіть номер кольору: ")  
 width\_multiplier, height\_multiplier = console\_ui.choose\_font\_size()  
  
 ascii\_art = art\_generator.generate(text, font)  
 resized\_art = art\_generator.resize(ascii\_art, width\_multiplier, height\_multiplier)  
 colored\_art = art\_generator.colorize(resized\_art, color)  
  
 console\_ui.print\_colored\_text(colored\_art, color)  
  
 preview = console\_ui.get\_input("Зберегти цей ASCII ART? (y/n) ")  
 if preview.lower() == 'y':  
 art\_generator.save(resized\_art)  
 console\_ui.display\_message("ASCII ART збережений у файлі 'ascii\_art.txt'")  
  
 cont = console\_ui.get\_input("Хочете спробувати ще раз? (y/n) ")  
 if cont.lower() != 'y':  
 break  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

**Functions.py**

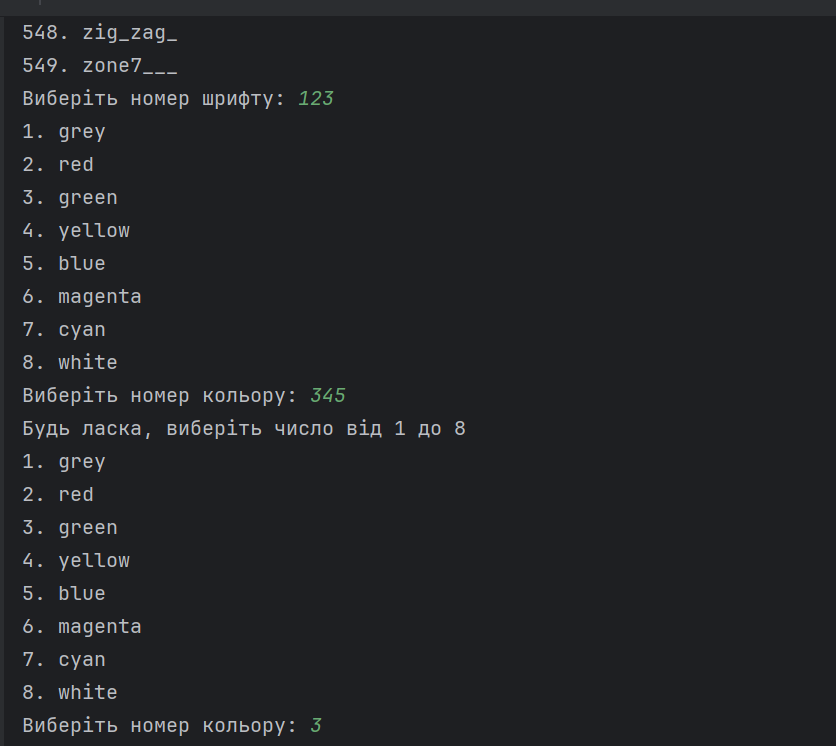
import pyfiglet  
from colorama import init, Fore  
  
class ASCIIArtGenerator:  
 def \_\_init\_\_(self):  
 init(autoreset=True)  
  
 @staticmethod  
 def contains\_ukrainian(text):  
 return any(char in "абвгґдеєжзиіїйклмнопрстуфхцчшщьюя" for char in text.lower())  
  
 def available\_fonts(self):  
 return pyfiglet.FigletFont.getFonts()  
  
 def available\_colors(self):  
 return ['grey', 'red', 'green', 'yellow', 'blue', 'magenta', 'cyan', 'white']  
  
 def generate(self, text, font):  
 return pyfiglet.figlet\_format(text, font=font)  
  
 def colorize(self, text, color):  
 color\_map = {  
 'grey': Fore.LIGHTBLACK\_EX,  
 'red': Fore.RED,  
 'green': Fore.GREEN,  
 'yellow': Fore.YELLOW,  
 'blue': Fore.BLUE,  
 'magenta': Fore.MAGENTA,  
 'cyan': Fore.CYAN,  
 'white': Fore.LIGHTWHITE\_EX  
 }  
 return color\_map[color] + text  
  
 def save(self, ascii\_art):  
 with open('ascii\_art.txt', 'w') as file:  
 file.write(ascii\_art)  
  
 @staticmethod  
 def resize(ascii\_art, width\_multiplier, height\_multiplier):  
 lines = ascii\_art.split("\n")  
 new\_lines = []  
  
 for line in lines:  
 new\_line = "".join([char \* width\_multiplier for char in line])  
 for \_ in range(height\_multiplier):  
 new\_lines.append(new\_line)  
  
 return "\n".join(new\_lines)

**Console.py**

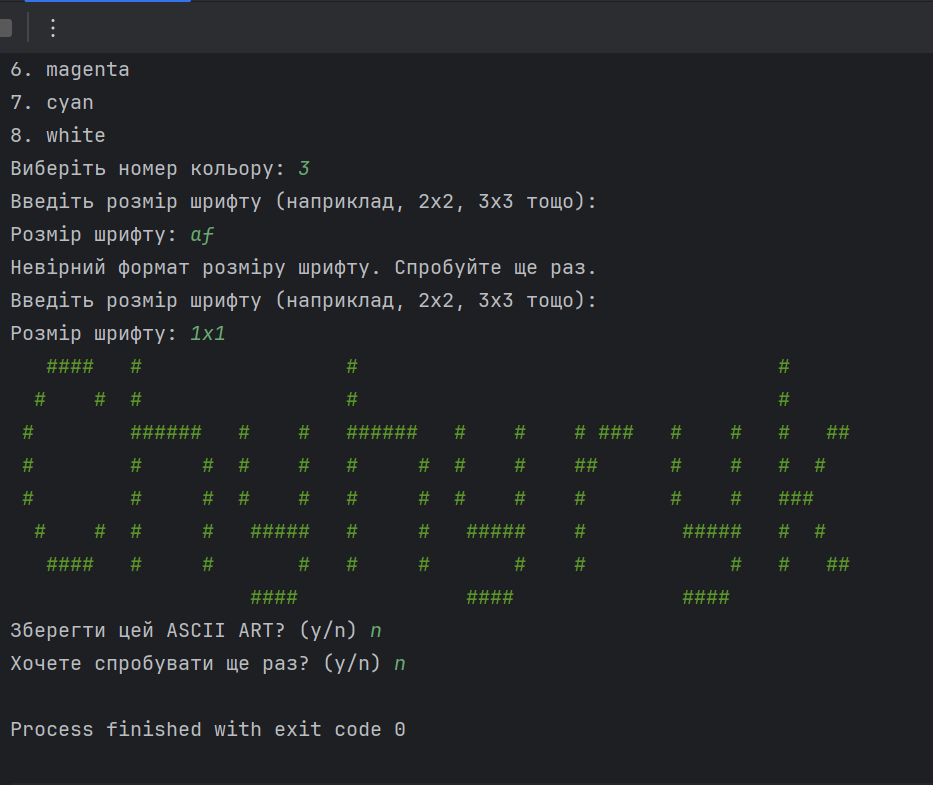
class Console:  
  
 @staticmethod  
 def get\_input(prompt):  
 return input(prompt)  
  
 def choose\_option(self, options, prompt="Виберіть опцію: "):  
 while True:  
 for i, option in enumerate(options, 1):  
 print(f"{i}. {option}")  
  
 try:  
 choice = int(input(prompt))  
 if 1 <= choice <= len(options):  
 return options[choice - 1]  
 else:  
 print(f"Будь ласка, виберіть число від 1 до {len(options)}")  
 except ValueError:  
 print("Введено невірний формат. Будь ласка, введіть номер.")  
  
 def print\_colored\_text(self, text, color):  
 print(text, end="", flush=True)  
  
 def print\_message(self, message):  
 print(message)  
  
 def choose\_font\_size(self):  
 while True:  
 try:  
 print("Введіть розмір шрифту (наприклад, 2x2, 3x3 тощо):")  
 font\_size = input("Розмір шрифту: ")  
 width\_multiplier, height\_multiplier = map(int, font\_size.split('x'))  
 return width\_multiplier, height\_multiplier  
 except ValueError:  
 print("Невірний формат розміру шрифту. Спробуйте ще раз.")



*Рис. 1 Результат виконання завдання*



*Рис. 2 Результат виконання завдання*



*Рис. 3 Результат виконання завдання*

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