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

Виконала:

ст. гр. ІТ-32

Ольга ЧИГИРИК

Прийняв

доцент каф. ІСМ:

Сергій ЩЕРБАК

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

Львів-2023

**Мета**: Створення Генератора ASCII-арту без використання зовнішніх бібліотек

План роботи

**Хід 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.

**Main.py**

# from Scripts.all\_python\_labs.shared\_lib.input\_validator import InputValidator  
# from Scripts.all\_python\_labs.shared\_lib.console\_ui import ConsoleUI  
from ascii\_art\_generator4 import ASCIIArtGenerator  
from console4 import Console  
from alphabet4 import Alphabet # Assuming the updated Alphabet class is in a file named alphabet4.py  
  
def main():  
 console = Console()  
  
 # Setting up the ASCII Art Generator  
 language = 'english'  
 art\_generator = ASCIIArtGenerator(language=language)  
  
 while True:  
 # Customizing output color and symbol  
 output\_color = console.get\_input("Choose an output color (default, red, green, etc.): ")  
 console.set\_output\_color(output\_color)  
  
 # Getting text input for ASCII art  
 text = console.get\_input("Enter a word or phrase for ASCII art: ")  
  
 # Checking for language constraints  
 if language == 'ukrainian' and art\_generator.contains\_ukrainian(text):  
 console.print\_message("Please enter text in Ukrainian.")  
 continue  
  
 # Generating ASCII art  
 ascii\_art = art\_generator.generate(text)  
 console.print\_message(ascii\_art)  
  
 # Option to save ASCII art  
 save = console.get\_input("Save this ASCII ART? (y/n): ").lower() == 'y'  
 if save:  
 art\_generator.save(ascii\_art)  
 console.print\_message("ASCII art saved in 'ascii\_art.txt'.")  
  
 # Option to try again or exit  
 if console.get\_input("Try again? (y/n): ").lower() != 'y':  
 break  
  
# def main():  
# console\_ui = ConsoleUI()  
# language = 'english'  
#  
# art\_generator = ASCIIArtGenerator(language=language)  
#  
# while True:  
# text = console\_ui.get\_input("Enter a word or phrase for ASCII art: ")  
#  
# if language == 'ukrainian' and art\_generator.contains\_ukrainian(text):  
# console\_ui.print\_message("Please enter text in Ukrainian.")  
# continue  
#  
# ascii\_art = art\_generator.generate(text)  
# console\_ui.print\_message(ascii\_art)  
#  
# save = console\_ui.get\_input("Save this ASCII ART? (y/n): ").lower() == 'y'  
# if save:  
# art\_generator.save(ascii\_art)  
# console\_ui.print\_message("ASCII art saved in 'ascii\_art.txt'.")  
#  
# if console\_ui.get\_input("Try again? (y/n): ").lower() != 'y':  
# break  
#  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

**Console.py**

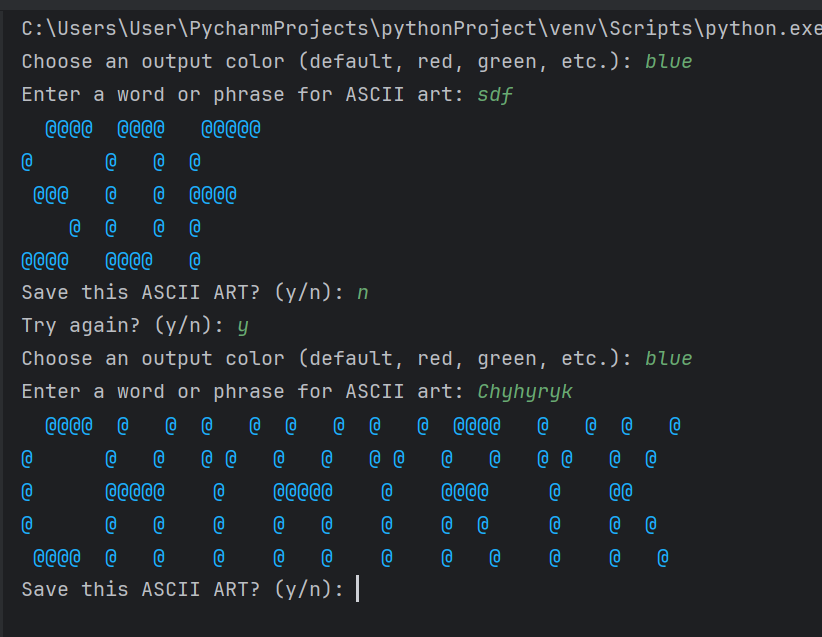
class Console:  
 output\_color = 'default'  
  
 @staticmethod  
 def get\_input(prompt):  
 *""" Get input from the user with error checking. """* while True:  
 user\_input = input(prompt)  
 if Console.\_is\_input\_valid(user\_input):  
 return user\_input  
 else:  
 Console.print\_message("Input error detected. Do you want to rewrite? (yes/no): ")  
 if input().strip().lower() != 'yes':  
 return None  
  
 @staticmethod  
 def print\_message(message):  
 *""" Print a message to the user with the chosen color and symbol. """* color\_start = Console.\_get\_color\_code(Console.output\_color)  
 color\_end = Console.\_get\_color\_code('end')  
 print(f"{color\_start} {message}{color\_end}")  
  
 @staticmethod  
 def set\_output\_color(color):  
 *""" Set the color of the output text. """* Console.output\_color = color  
  
  
 @staticmethod  
 def \_get\_color\_code(color):  
 *""" Get the ANSI color code for the given color. """* colors = {  
 'default': '',  
 'red': '\033[91m',  
 'green': '\033[92m',  
 'yellow': '\033[93m',  
 'blue': '\033[94m',  
 'magenta': '\033[95m',  
 'cyan': '\033[96m',  
 'end': '\033[0m'  
 }  
 return colors.get(color, '')  
  
 @staticmethod  
 def \_is\_input\_valid(user\_input):  
 *""" Check if the user input is valid (not empty or just whitespace). """* return user\_input.strip() != ''  
  
# Example usage  
Console.set\_output\_color('blue')

**Ascii\_art\_generator.py**

from Scripts.all\_python\_labs.lab\_python\_4.alphabet4 import get\_ascii\_art  
  
from Scripts.all\_python\_labs.lab\_python\_4 import \*  
class ASCIIArtGenerator:  
 def \_\_init\_\_(self, language='english'):  
 self.language = language  
  
 def generate(self, text):  
 lines = ['' for \_ in range(5)] # Припускаємо 5 рядків на літеру  
 for char in text:  
 char\_art = get\_ascii\_art(char, self.language)  
 if char\_art:  
 for i, line in enumerate(char\_art):  
 lines[i] += line + " " # Додаємо пробіл між символами  
 else:  
 # Якщо символ не знайдено, повертаємо повідомлення про це  
 return f"Символ '{char}' не знайдено в ASCII арт представленнях."  
 return "\n".join(lines)  
  
 def save(self, ascii\_art, filename='ascii\_art.txt'):  
 *"""Saves ASCII art to a file."""* try:  
 with open(filename, 'w', encoding='utf-8') as file:  
 file.write(ascii\_art)  
 return True  
 except Exception as e:  
 print(f"Error saving ASCII art: {e}")  
 return False

**Alphabet.py**

class Alphabet:  
 ascii\_art\_alphabet = {  
 'A': [" @@@@ ", "@ @", "@@@@@@", "@ @", "@ @"],  
 'B': ["@@@@ ", "@ @", "@@@@ ", "@ @", "@@@@ "],  
 'C': [" @@@@", "@ ", "@ ", "@ ", " @@@@"],  
 'D': ["@@@@ ", "@ @", "@ @", "@ @", "@@@@ "],  
 'E': ["@@@@@", "@ ", "@@@@ ", "@ ", "@@@@@"],  
 'F': ["@@@@@", "@ ", "@@@@ ", "@ ", "@" ],  
 'G': [" @@@@", "@ ", "@ @@", "@ @", " @@@@"],  
 'H': ["@ @", "@ @", "@@@@@", "@ @", "@ @"],  
 'I': [" @@@ ", " @ ", " @ ", " @ ", " @@@ "],  
 'J': [" @@", " @ ", " @ ", "@ @ ", " @@ "],  
 'K': ["@ @", "@ @ ", "@@ ", "@ @ ", "@ @"],  
 'L': ["@ ", "@ ", "@ ", "@ ", "@@@@@"],  
 'M': ["@ @", "@@ @@", "@ @ @", "@ @", "@ @"],  
 'N': ["@ @", "@@ @", "@ @ @", "@ @@", "@ @"],  
 'O': [" @@@ ", "@ @", "@ @", "@ @", " @@@ "],  
 'P': ["@@@@ ", "@ @", "@@@@ ", "@ ", "@ "],  
 'Q': [" @@@ ", "@ @", "@ @ @", "@ @ ", " @@ @"],  
 'R': ["@@@@ ", "@ @", "@@@@ ", "@ @ ", "@ @"],  
 'S': [" @@@@", "@ ", " @@@ ", " @", "@@@@ "],  
 'T': ["@@@@@", " @ ", " @ ", " @ ", " @ "],  
 'U': ["@ @", "@ @", "@ @", "@ @", " @@@ "],  
 'V': ["@ @", "@ @", "@ @", " @ @ ", " @ "],  
 'W': ["@ @", "@ @", "@ @ @", "@@ @@", "@ @"],  
 'X': ["@ @", " @ @ ", " @ ", " @ @ ", "@ @"],  
 'Y': ["@ @", " @ @ ", " @ ", " @ ", " @ "],  
 'Z': ["@@@@@", " @ ", " @ ", " @ ", "@@@@@"],  
 'a': [" @@@ ", " @ @ ", "@ @", "@@@@@@", "@ @"],  
 'b': ["@@@@ ", "@ @ ", "@@@@ ", "@ @ ", "@@@@ "],  
 'c': [" @@@@ ", "@ ", "@ ", "@ ", " @@@@"],  
 'd': ["@@@@ ", "@ @", "@ @", "@ @", "@@@@ "],  
 'e': ["@@@@@", "@ ", "@@@@ ", "@ ", "@@@@@"],  
 'f': ["@@@@@", "@ ", "@@@@ ", "@ ", "@"],  
 'g': [" @@@@", "@ ", "@ @@", "@ @", " @@@@"],  
 'h': ["@ @", "@ @", "@@@@@", "@ @", "@ @"],  
 'i': [" @@@ ", " @ ", " @ ", " @ ", " @@@ "],  
 'j': [" @@", " @ ", " @ ", "@ @ ", " @@ "],  
 'k': ["@ @", "@ @ ", "@@ ", "@ @ ", "@ @"],  
 'l': ["@ ", "@ ", "@ ", "@ ", "@@@@@"],  
 'm': ["@ @", "@@ @@", "@ @ @", "@ @", "@ @"],  
 'n': ["@ @", "@@ @", "@ @ @", "@ @@", "@ @"],  
 'o': [" @@@ ", "@ @", "@ @", "@ @", " @@@ "],  
 'p': ["@@@@ ", "@ @", "@@@@ ", "@ ", "@ "],  
 'q': [" @@@ ", "@ @", "@ @ @", "@ @ ", " @@ @"],  
 'r': ["@@@@ ", "@ @", "@@@@ ", "@ @ ", "@ @"],  
 's': [" @@@@", "@ ", " @@@ ", " @", "@@@@ "],  
 't': ["@@@@@", " @ ", " @ ", " @ ", " @ "],  
 'u': ["@ @", "@ @", "@ @", "@ @", " @@@ "],  
 'v': ["@ @", "@ @", "@ @", " @ @ ", " @ "],  
 'w': ["@ @", "@ @", "@ @ @", "@@ @@", "@ @"],  
 'x': ["@ @", " @ @ ", " @ ", " @ @ ", "@ @"],  
 'y': ["@ @", " @ @ ", " @ ", " @ ", " @ "],  
 'z': ["@@@@@", " @ ", " @ ", " @ ", "@@@@@"],  
 # Symbols  
 '!': [" @ ", " @ ", " @ ", " ", " @ "],  
 '@': [" @@@@ ", "@ @", "@ @@@@", "@ @@@ ", "@ "],  
 '#': [" @ @ ", "@@@@@", " @ @ ", "@@@@@", " @ @ "],  
 '$': [" @$@ ", "@$ ", " @$@ ", " $@", " @$@ "],  
 '%': ["@ @", " @ ", " @ ", " @ ", "@ @"],  
 '^': [" @ ", " @ @ ", "@ @", " ", " "],  
 '&': [" @ ", "@ @ ", " @ @ ", "@ @@", " @@ @"],  
 '\*': [" ", "@ @ @", " @@@ ", "@ @ @", " "],  
 '(': [" @ ", " @ ", "@ ", " @ ", " @ "],  
 ')': [" @ ", " @ ", " @", " @ ", " @ "],  
 '-': [" ", " ", " @@@ ", " ", " "],  
 '+': [" ", " @ ", "@@@@@", " @ ", " "],  
 '=': [" ", "@@@@@", " ", "@@@@@", " "],  
 '[': [" @@@ ", " @ ", " @ ", " @ ", " @@@ "],  
 ']': [" @@@ ", " @ ", " @ ", " @ ", " @@@ "],  
 '{': [" @@ ", " @ ", "@@ ", " @ ", " @@ "],  
 '}': ["@@ ", " @ ", " @@", " @ ", "@@ "],  
 '|': [" @ ", " @ ", " @ ", " @ ", " @ "],  
 '\\': ["@ ", " @ ", " @ ", " @ ", " @"],  
 '/': [" @ ", " @ ", " @ ", " @ ", "@ "],  
 '<': [" @ ", " @ ", "@ ", " @ ", " @ "],  
 '>': [" @ ", " @ ", " @", " @ ", " @ "],  
 '?': [" @@@ ", "@ @", " @ ", " ", " @ "],  
 '"': ["@ @ ", "@ @ ", " ", " ", " "],  
 "'": [" @ ", " @ ", " ", " ", " "],  
 '`': [" @ ", " @ ", " ", " ", " "],  
 '~': [" ", " @ @", "@ @ ", " ", " "],  
 '\_': [" ", " ", " ", " ", "@@@@@"],  
 ':': [" ", " @ ", " ", " @ ", " "],  
 ';': [" @ ", " ", " @ ", " @ ", "@ "],  
 ',': [" ", " ", " @ ", " @ ", "@ "],  
 '.': [" ", " ", " ", " @ ", " "],  
 '/': [" @", " @ ", " @ ", " @ ", "@ "],  
 '0': [" @@@ ", "@ @", "@ @", "@ @", " @@@ "],  
 '1': [" @ ", " @@ ", " @ ", " @ ", "@@@@@"],  
 '2': [" @@@ ", "@ @", " @ ", " @ ", "@@@@@"],  
 '3': [" @@@@", " @", " @@ ", " @", "@@@@ "],  
 '4': [" @ ", " @@ ", " @ @ ", "@@@@@", " @ "],  
 '5': ["@@@@@", "@ ", "@@@@ ", " @", "@@@@ "],  
 '6': [" @@@ ", " @ ", "@@@@ ", "@ @ ", " @@@ "],  
 '7': ["@@@@@", " @", " @ ", " @ ", " @ "],  
 '8': [" @@@ ", "@ @", " @@@ ", "@ @", " @@@ "],  
 '9': [" @@@ ", "@ @", " @@@@", " @", " @@@ "]  
  
 }  
  
 ascii\_art\_ukrainian\_alphabet = {  
 'А': [" @@@@ ", "@ @", "@@@@@@", "@ @", "@ @"],  
 'Б': ["@@@@@ ", "@ ", "@@@@ ", "@ @", "@@@@ "],  
 'В': ["@@@@ ", "@ @", "@@@@ ", "@ @", "@@@@ "],  
 'Г': ["@@@@@", "@ ", "@ ", "@ ", "@ "],  
 'Ґ': ["@@@@@", "@ ", "@@@@@", "@ ", "@ "],  
 'Д': [" @@@@", " @ @ ", "@ @", "@ @", "@@@@@"],  
 'Е': ["@@@@@", "@ ", "@@@@ ", "@ ", "@@@@@"],  
 'Є': [" @@@@", "@ ", "@@@@ ", "@ @", " @@@@"],  
 'Ж': ["@ @ @", " @ @ ", " @ ", " @ @ ", "@ @ @"],  
 'З': [" @@@@", "@ ", " @@ ", " @", "@@@@ "],  
 'И': ["@ @", "@ @@", "@ @ @", "@@ @", "@ @"],  
 'І': [" @@@ ", " @ ", " @ ", " @ ", " @@@ "],  
 'Ї': ["@ @ @", " ", " @@@ ", " @ ", " @@@ "],  
 'Й': [" @ @ ", " ", "@ @", "@ @@", "@ @ @"],  
 'К': ["@ @", "@ @ ", "@@ ", "@ @ ", "@ @"],  
 'Л': [" @@@@", " @ @", "@ @", "@ @", "@ @"],  
 'М': ["@ @", "@@ @@", "@ @ @", "@ @", "@ @"],  
 'Н': ["@ @", "@ @", "@@@@@", "@ @", "@ @"],  
 'О': [" @@@ ", "@ @", "@ @", "@ @", " @@@ "],  
 'П': ["@@@@@", "@ @", "@ @", "@ @", "@ @"],  
 'Р': ["@@@@ ", "@ @", "@@@@ ", "@ ", "@ "],  
 'С': [" @@@@", "@ ", "@ ", "@ ", " @@@@"],  
 'Т': ["@@@@@", " @ ", " @ ", " @ ", " @ "],  
 'У': ["@ @", "@ @", " @@@ ", " @ ", "@@ "],  
 'Ф': [" @@@ ", "@ @ @", " @@@ ", " @ ", " @ "],  
 'Х': ["@ @", " @ @ ", " @ ", " @ @ ", "@ @"],  
 'Ц': ["@ @", "@ @", "@ @", "@@@@@", " @"],  
 'Ч': ["@ @", "@ @", " @@@@", " @", " @"],  
 'Ш': ["@ @ @", "@ @ @", "@ @ @", "@ @ @", "@@@@@"],  
 'Щ': ["@ @ @", "@ @ @", "@ @ @", "@@@@@", " @"],  
 'Ь': ["@ ", "@ ", "@@@@ ", "@ @", "@@@@ "],  
 'Ю': ["@ @@ ", "@ @ @", "@@@ @", "@ @ @", "@ @@ "],  
 'Я': [" @@@@", "@ @", " @@@@", " @ @", "@ @"],  
 # Lowercase Ukrainian letters  
 'а': [" @@@ ", " @ @ ", "@ @", "@ @", " @@@@@ "],  
 'б': [" @@@@@ ", "@ ", "@@@@@ ", "@ @", "@@@@@ "],  
 'в': ["@@@@ ", "@ @ ", "@@@@ ", "@ @ ", "@@@@ "],  
 'г': ["@@@@@", "@ ", "@ ", "@ ", "@ "],  
 'ґ': ["@@@@@", "@ ", "@@@@ ", "@ ", "@ "],  
 'д': [" @@@@", " @ @ ", "@ @", "@ @", "@@@@@"],  
 'е': [" @@@@ ", "@ ", "@@@@ ", "@ ", " @@@@ "],  
 'є': [" @@@@", " @ ", "@@@@ ", "@ @", " @@@@"],  
 'ж': ["@ @ @", " @ @ ", " @ ", " @ @ ", "@ @ @"],  
 'з': [" @@@@", "@ ", " @@ ", " @", "@@@@ "],  
 'и': ["@ @", "@ @@", "@ @ @", "@@ @", "@ @"],  
 'і': [" @@@ ", " @ ", " @ ", " @ ", " @@@ "],  
 'ї': ["@ @ @", " ", " @@@ ", " @ ", " @@@ "],  
 'й': [" @ @ ", " ", "@ @", "@ @@", "@ @ @"],  
 'к': ["@ @", "@ @ ", "@@ ", "@ @ ", "@ @"],  
 'л': [" @@@@", " @ @", "@ @", "@ @", "@ @"],  
 'м': ["@ @", "@@ @@", "@ @ @", "@ @", "@ @"],  
 'н': ["@ @", "@ @", "@@@@@", "@ @", "@ @"],  
 'о': [" @@@ ", "@ @", "@ @", "@ @", " @@@ "],  
 'п': ["@@@@@", "@ @", "@ @", "@ @", "@ @"],  
 'р': ["@@@@ ", "@ @", "@@@@ ", "@ ", "@ "],  
 'с': [" @@@@", "@ ", "@ ", "@ ", " @@@@"],  
 'т': ["@@@@@", " @ ", " @ ", " @ ", " @ "],  
 'у': ["@ @", "@ @", " @@@ ", " @ ", "@@ "],  
 'ф': [" @@@ ", "@ @ @", " @@@ ", " @ ", " @ "],  
 'х': ["@ @", " @ @ ", " @ ", " @ @ ", "@ @"],  
 'ц': ["@ @", "@ @", "@ @", "@@@@@", " @"],  
 'ч': ["@ @", "@ @", " @@@@", " @", " @"],  
 'ш': ["@ @ @", "@ @ @", "@ @ @", "@ @ @", "@@@@@"],  
 'щ': ["@ @ @", "@ @ @", "@ @ @", "@@@@@", " @"],  
 'ь': ["@ ", "@ ", "@@@@ ", "@ @", "@@@@ "],  
 'ю': ["@ @@ ", "@ @ @", "@@@ @", "@ @ @", "@ @@ "],  
 'я': [" @@@@", "@ @", " @@@@", " @ @", "@ @"],  
 '!': [" @ ", " @ ", " @ ", " ", " @ "],  
 '@': [" @@@@ ", "@ @", "@ @@@@", "@ @@@ ", "@ "],  
 '#': [" @ @ ", "@@@@@", " @ @ ", "@@@@@", " @ @ "],  
 '$': [" @$@ ", "@$ ", " @$@ ", " $@", " @$@ "],  
 '%': ["@ @", " @ ", " @ ", " @ ", "@ @"],  
 '^': [" @ ", " @ @ ", "@ @", " ", " "],  
 '&': [" @ ", "@ @ ", " @ @ ", "@ @@", " @@ @"],  
 '\*': [" ", "@ @ @", " @@@ ", "@ @ @", " "],  
 '(': [" @ ", " @ ", "@ ", " @ ", " @ "],  
 ')': [" @ ", " @ ", " @", " @ ", " @ "],  
 '-': [" ", " ", " @@@ ", " ", " "],  
 '+': [" ", " @ ", "@@@@@", " @ ", " "],  
 '=': [" ", "@@@@@", " ", "@@@@@", " "],  
 '[': [" @@@ ", " @ ", " @ ", " @ ", " @@@ "],  
 ']': [" @@@ ", " @ ", " @ ", " @ ", " @@@ "],  
 '{': [" @@ ", " @ ", "@@ ", " @ ", " @@ "],  
 '}': ["@@ ", " @ ", " @@", " @ ", "@@ "],  
 '|': [" @ ", " @ ", " @ ", " @ ", " @ "],  
 '\\': ["@ ", " @ ", " @ ", " @ ", " @"],  
 '/': [" @ ", " @ ", " @ ", " @ ", "@ "],  
 '<': [" @ ", " @ ", "@ ", " @ ", " @ "],  
 '>': [" @ ", " @ ", " @", " @ ", " @ "],  
 '?': [" @@@ ", "@ @", " @ ", " ", " @ "],  
 '"': ["@ @ ", "@ @ ", " ", " ", " "],  
 "'": [" @ ", " @ ", " ", " ", " "],  
 '`': [" @ ", " @ ", " ", " ", " "],  
 '~': [" ", " @ @", "@ @ ", " ", " "],  
 '\_': [" ", " ", " ", " ", "@@@@@"],  
 ':': [" ", " @ ", " ", " @ ", " "],  
 ';': [" @ ", " ", " @ ", " @ ", "@ "],  
 ',': [" ", " ", " @ ", " @ ", "@ "],  
 '.': [" ", " ", " ", " @ ", " "],  
 '/': [" @", " @ ", " @ ", " @ ", "@ "],  
 '0': [" @@@ ", "@ @", "@ @", "@ @", " @@@ "],  
 '1': [" @ ", " @@ ", " @ ", " @ ", "@@@@@"],  
 '2': [" @@@ ", "@ @", " @ ", " @ ", "@@@@@"],  
 '3': [" @@@@", " @", " @@ ", " @", "@@@@ "],  
 '4': [" @ ", " @@ ", " @ @ ", "@@@@@", " @ "],  
 '5': ["@@@@@", "@ ", "@@@@ ", " @", "@@@@ "],  
 '6': [" @@@ ", " @ ", "@@@@ ", "@ @ ", " @@@ "],  
 '7': ["@@@@@", " @", " @ ", " @ ", " @ "],  
 '8': [" @@@ ", "@ @", " @@@ ", "@ @", " @@@ "],  
 '9': [" @@@ ", "@ @", " @@@@", " @", " @@@ "]  
  
}  
  
  
@staticmethod  
def get\_ascii\_art(letter, language='english'):  
 if language == 'ukrainian':  
 return Alphabet.ascii\_art\_ukrainian\_alphabet.get(letter.upper())  
 else:  
 return Alphabet.ascii\_art\_alphabet.get(letter.upper())



*Рис. 1 Результат виконання завдання*

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