МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

ІНСТИТУТ КОМП’ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ



Лабораторна робота №3

з дисципліни “Спеціалізовані мови програмування”

на тему

“Розробка ASCII ART генератора для візуалізації текстових даних.”

Виконала:

студент групи РІ-32

Царук В’ячеслав

Прийняв:

к.т.н.,

доц. кафедри ІСМ

Сергій ЩЕРБАК

Львів-2023

**Мета роботи:** Розробка консольного калькулятора в об’єктно-орієнтованому стилі з використанням класів.

**План роботи**

Завдання 1: Введення користувача

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

Код програми:

def user\_input(self):

self.text = input("Write your text: ")

Завдання 2: Бібліотека ASCII-арту

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

Код програми:

import pyfiglet

import termcolor

import coloramaЗавдання 3: Вибір шрифту

Дозвольте користувачам вибирати різні стилі шрифтів для свого ASCII-арту. Надайте список доступних шрифтів та дозвольте їм вибрати один.

Код програми:

def text\_2\_art(*self*):  
 figlet = pyfiglet.Figlet(font=*self*.font or "slant")  
 *self*.art = figlet.renderText(*self*.text)  
   
 *self*.resize\_art()  
   
 if *self*.color:  
 *self*.art = termcolor.colored(*self*.art, *self*.color)  
 *self*.display\_preview()  
  
 if not *self*.ask\_for\_preview\_approval():  
 print("Let's modify your choices.")  
 *self*.running\_text\_to\_art()

Завдання 4: Колір тексту

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

Код програми:

def color\_choose(self):

colors = ["red", "green", "yellow", "blue", "magenta", "cyan", "white", "grey"]

print("Available colors:", ', '.join(colors))

chosen\_color = input("Choose a color from the list or press enter for default: ").strip().lower()

if chosen\_color in colors:

self.color = chosen\_color

else:

print("Unknown color. Using default.")

Завдання 5: Форматування виводу

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

Код програми:

ef display\_preview(self):

try:

terminal\_width = os.get\_terminal\_size().columns

except OSError:

terminal\_width = 80

for line in self.art.splitlines():

print(line.center(terminal\_width))

Завдання 6: Збереження у файл

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

Код програми:

def save\_to\_file(self, filename):

save = input("Do you want to save the ASCII art to a file? (yes/no): ")

if save.lower() == "yes":

with open("ascii\_art.txt", "w") as file:

file.write(self.art)

print("Saved to ascii\_art.txt!")

exit()

if save.lower() == "no":

print("Ok! Bye!")

exit()

if save.lower() != "yes" or "no":

print("Invalid input")

Завдання 7: Розмір ARTу

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

Код програми:

def set\_art\_size(self):

while True:

try:

width\_input = input("Set the width (default is 80, press enter for default): ")

if width\_input: # Check if user input something

self.width = int(width\_input)

height\_input = input("Set the height (press enter to skip): ")

if height\_input: # Check if user input something

self.height = int(height\_input)

break # Exit the loop if inputs are correct

except ValueError:

print("Invalid input. Please enter a valid number or press enter for default values.")

Завдання 8: Вибір символів

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

Код програми:

def replace\_symbols(self):

old\_symbol = input("Enter the symbol in the ASCII art you want to replace: ")

new\_symbol = input(f"Enter the symbol you want to replace '{old\_symbol}' with: ")

self.art = self.art.replace(old\_symbol, new\_symbol)

Завдання 9: Функція попереднього перегляду

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

Код програми:

def display\_preview(self):

terminal\_width, \_ = shutil.get\_terminal\_size((80, 20))

for line in self.art.splitlines():

print(line.center(terminal\_width))

Завдання 10: Інтерфейс, зрозумілий для користувача

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

Код програми:

import pyfiglet

import termcolor

import colorama

import os

import shutil

import re

class TextArt:

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

self.text = ""

self.art = ""

self.font = ""

self.color = ""

self.width = 80

self.height = None

def user\_input(self):

self.text = input("Write your text: ")

def text\_2\_art(self):

figlet = pyfiglet.Figlet(font=self.font or "slant")

self.art = figlet.renderText(self.text)

self.resize\_art()

if self.color:

self.art = termcolor.colored(self.art, self.color)

self.display\_preview()

if not self.ask\_for\_preview\_approval():

print("Let's modify your choices.")

self.running\_text\_to\_art()

def running\_text\_to\_art(self):

colorama.init()

print("Welcome to Text2Art")

self.user\_input()

self.font\_choose()

self.set\_art\_size()

self.replace\_symbols()

self.color\_choose()

self.text\_2\_art()

self.display\_preview()

self.save\_to\_file("art.txt")

class Settings\_TextArt(TextArt):

def font\_choose(self):

fonts = ["slant", "3-d", "roman"]

print("Available fonts:", ', '.join(fonts))

chosen\_font = input("Choose a font from the list or press enter for default: ").strip().lower()

if chosen\_font in fonts:

self.font = chosen\_font

else:

print("Unknown font. Using default.")

def color\_choose(self):

colors = ["red", "green", "blue", "cyan", "white", "black"]

print("Available colors:", ', '.join(colors))

chosen\_color = input("Choose a color from the list or press enter for default: ").strip().lower()

if chosen\_color in colors:

self.color = chosen\_color

else:

print("Unknown color. Using default.")

def save\_to\_file(self, filename):

save = input("Do you want to save the ASCII art to a file? (yes/no): ")

if save.lower() == "yes":

with open(filename, "w") as file:

file.write(self.art)

print(f"Saved to {filename}!")

exit()

if save.lower() == "no":

print("Ok! Bye!")

exit()

else:

print("Invalid input")

def display\_preview(self):

terminal\_width, \_ = shutil.get\_terminal\_size((80, 20))

for line in self.art.splitlines():

print(line.center(terminal\_width))

def ask\_for\_preview\_approval(self):

while True:

response = input("Are you satisfied with the preview? (yes/no): ").strip().lower()

if response == "yes":

return True

if response == "no":

return False

else:

print("Invalid input. Please enter 'yes' or 'no'.")

def set\_art\_size(self):

while True:

try:

width\_input = input("Set the width (default is 80, press enter for default): ")

if width\_input: # Check if user input something

self.width = int(width\_input)

height\_input = input("Set the height (press enter to skip): ")

if height\_input: # Check if user input something

self.height = int(height\_input)

break # Exit the loop if inputs are correct

except ValueError:

print("Invalid input. Please enter a valid number or press enter for default values.")

def resize\_art(self):

lines = self.art.split("\n")

resized\_art = []

for line in lines:

resized\_art.append(line[:self.width])

self.art = "\n".join(resized\_art)

def replace\_symbols(self):

old\_symbol = input("Enter the symbol in the ASCII art you want to replace: ")

new\_symbol = input(f"Enter the symbol you want to replace '{old\_symbol}' with: ")

self.art = self.art.replace(old\_symbol, new\_symbol)

text\_art\_running = Settings\_TextArt()

text\_art\_running.running\_text\_to\_art()

**GitHub-репозиторій:**

**Висновок:** Зробивши завдання цієї лабораторної роботи, я створив універсальний Генератор ASCII-арту, який дозволить користувачам налаштовувати свої творіння з різними шрифтами, кольорами, розмірами та символами.