Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Лабораторна роботи №3

з дисципліни «Спеціалізовані мови програмування»

на тему

«Розробка ASCII ART генератора для візуалізації текстових даних»

Виконав:

Костик В. Ю.

Перевірив:

Щербак С. С.

Львів 2024

**Мета:** створення додатка Генератора ASCII-арту

**План 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.

**Реалізація:**

**Папка AppSettings, файл AppSettings.py:**  
from classes.ASCIIArtApp import ASCIIArtApp

if \_\_name\_\_ == "\_\_main\_\_":

app = ASCIIArtApp()

app.run()

**Папка assets, файл assets.py:**

import pyfiglet

def get\_available\_fonts():

"""Повертає список доступних шрифтів для ASCII-арту."""

return pyfiglet.FigletFont.getFonts()

def preview\_font\_example(font):

"""Показує приклад шрифту."""

return pyfiglet.figlet\_format("Example", font=font)

**Папка classes, файл ASCIIArt.py:**

import pyfiglet

from colorama import Fore

class ASCIIArt:

def \_\_init\_\_(self, text, font="standard", color=Fore.WHITE, width=80, custom\_char="#"):

self.text = text

self.font = font

self.color = color

self.width = width

self.custom\_char = custom\_char

self.ascii\_art = ""

def generate\_art(self):

self.ascii\_art = pyfiglet.figlet\_format(self.text, font=self.font, width=self.width)

self.\_apply\_custom\_characters()

return self.color + self.ascii\_art + Fore.RESET

def \_apply\_custom\_characters(self):

self.ascii\_art = self.ascii\_art.replace('@', self.custom\_char)

self.ascii\_art = self.ascii\_art.replace('#', self.custom\_char)

def preview(self):

print("Попередній перегляд ASCII-арту:")

print(self.generate\_art())

def save\_to\_file(self, file\_path):

with open(file\_path, "w") as f:

f.write(self.ascii\_art)

print(f"ASCII-арт збережено у файл {file\_path}")

**Папка classes, файл ASCIIArtApp.py:**

from colorama import Fore

import re

from classes.ASCIIArt import ASCIIArt

from functions import functions

class ASCIIArtApp:

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

self.colors = {

1: Fore.RED,

2: Fore.BLUE,

3: Fore.GREEN,

4: Fore.WHITE,

5: Fore.YELLOW

}

def get\_font\_choice(self):

fonts = functions.get\_available\_fonts()

return functions.font\_selection(fonts)

def get\_user\_input(self):

# Завдання 1: Введення користувача

text = functions.get\_validated\_text()

# Завдання 2: Вибір шрифту

print("Доступні шрифти:")

selected\_font = self.get\_font\_choice()

# Завдання 8: Вибір символів

custom\_char = functions.get\_validated\_custom\_char()

# Завдання 7: Розмір ARTу

width = functions.get\_width()

# Завдання 4: Колір тексту

color = functions.get\_color\_choice(self.colors)

return ASCIIArt(text, selected\_font, color, width, custom\_char)

def run(self):

art = self.get\_user\_input()

# Завдання 5: Форматування виводу

print(art.generate\_art())

# Завдання 6: Збереження у файл

save\_choice = functions.get\_save\_choice()

if save\_choice == "так":

file\_path = r"C:\Users\Vlad\Desktop\унік\пітоній\Calculator\_Pro\_Max\_beta\_v033\art.txt"

art.save\_to\_file(file\_path)

# Завдання 9: Попередній перегляд

art.preview()

**Папка functions, файл functions.py:**import pyfiglet

import re

from colorama import Fore

def validate\_latin\_input(text):

"""Перевірка, чи містить введений текст тільки латинські символи."""

if re.match("^[A-Za-z0-9 ]+$", text):

return True

print("Помилка: Текст має містити тільки латинські символи та цифри.")

return False

def validate\_custom\_char(char):

"""Перевірка, чи введений символ не містить кирилицю."""

if re.match("^[^\u0400-\u04FF]+$", char): # Регулярний вираз, що виключає кирилицю

return True

print("Помилка: Символ не має містити кириличні символи.")

return False

def font\_selection(fonts):

page\_size = 10

current\_page = 0

total\_pages = len(fonts) // page\_size + (1 if len(fonts) % page\_size else 0)

while True:

print(f"\nСторінка {current\_page + 1}/{total\_pages}")

start\_index = current\_page \* page\_size

end\_index = min(start\_index + page\_size, len(fonts))

for index, font in enumerate(fonts[start\_index:end\_index], start=start\_index + 1):

print(f"{index}. {font}")

user\_action = input("\nВведіть номер шрифту (від 1 до 549), або '>' для наступної сторінки, '<' для попередньої, або 'приклад [номер]' для перегляду прикладу: ")

if user\_action == '>':

if current\_page < total\_pages - 1:

current\_page += 1

else:

print("Ви на останній сторінці.")

elif user\_action == '<':

if current\_page > 0:

current\_page -= 1

else:

print("Ви на першій сторінці.")

elif user\_action.startswith("приклад"):

try:

font\_index = int(user\_action.split()[1]) - 1

if 0 <= font\_index < len(fonts):

print(f"\nПриклад шрифту '{fonts[font\_index]}':")

print(pyfiglet.figlet\_format("Example", font=fonts[font\_index]))

else:

print("Неправильний номер шрифту.")

except (IndexError, ValueError):

print("Неправильний формат команди.")

else:

try:

font\_choice = int(user\_action) - 1

if 0 <= font\_choice < len(fonts):

return fonts[font\_choice]

else:

print("Помилка: Виберіть правильний номер шрифту.")

except ValueError:

print("Помилка: Введіть число або команду.")

def get\_available\_fonts():

return pyfiglet.FigletFont.getFonts()

def get\_validated\_text():

while True:

text = input("Введіть слово або фразу для генерації ASCII-арту (латиницею): ")

if validate\_latin\_input(text):

return text

def get\_validated\_custom\_char():

while True:

custom\_char = input("Введіть символ для ASCII-арту (або натисніть Enter для використання '#'): ")

if not custom\_char:

return '#'

if validate\_custom\_char(custom\_char):

return custom\_char

def get\_width():

default\_width = 80

while True:

width\_input = input(f"Введіть ширину ASCII-арту (за замовчуванням {default\_width}): ")

if not width\_input:

return default\_width

try:

return int(width\_input)

except ValueError:

print("Помилка: Введіть число для ширини.")

def get\_color\_choice(colors):

print("Доступні кольори:")

print("1. Червоний")

print("2. Синій")

print("3. Зелений")

print("4. Білий")

print("5. Жовтий")

while True:

try:

color\_choice = int(input("Виберіть номер кольору: "))

if 1 <= color\_choice <= 5:

return colors.get(color\_choice, Fore.WHITE)

else:

print("Помилка: Виберіть номер кольору від 1 до 5.")

except ValueError:

print("Помилка: Введіть число.")

def get\_save\_choice():

while True:

save\_choice = input("Бажаєте зберегти ASCII-арт у файл? (так/ні): ").strip().lower()

if save\_choice in ["так", "ні"]:

return save\_choice

print("Помилка: Введіть 'так' або 'ні'.")

**Папка GlobalVariables, файл GlobalVariables.py:**

GLOBAL\_DATA = {}

**Папка interface, файл interface.py:**

def display\_menu():

print("Вітаємо в Генераторі ASCII-арту!")

print("Оберіть необхідну дію:")

**Папка logs, файл logs.py:**

def log\_action(action):

with open("log.txt", "a") as log\_file:

log\_file.write(action + "\n")

**Файл main.py:**

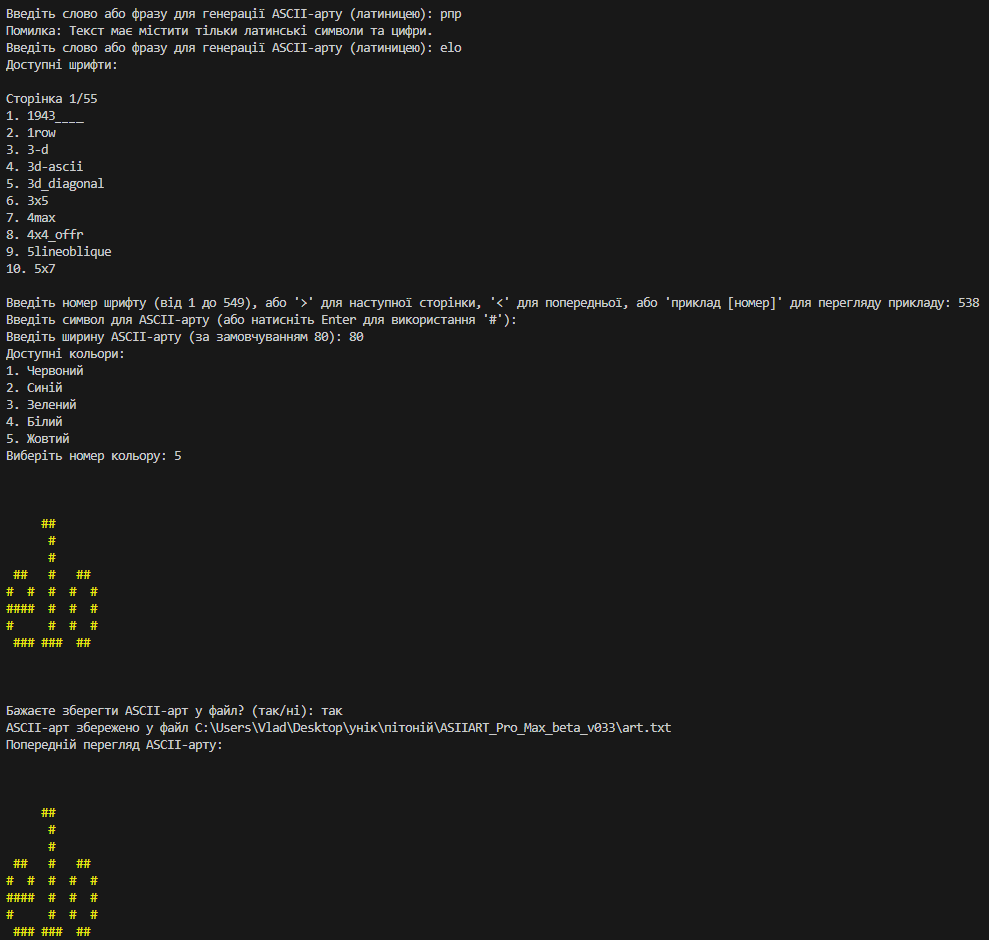
from classes.ASCIIArtApp import ASCIIArtApp

if \_\_name\_\_ == "\_\_main\_\_":

app = ASCIIArtApp()

app.run()

**Результат виконання:**

****

**Висновок:** на цій лабораторній роботі я створив універсальний Генератор ASCII-арту, який дозволить користувачам налаштовувати свої творіння з різними шрифтами, кольорами, розмірами та символами