МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

Кафедра інформаційних систем та мереж

**Звіт**

До лабораторної роботи №3

**Розробка ASCII ART генератора для візуалізації текстових даних**

Виконав:

ст. гр. ІТ-21сп

**Микола ПЛЕБАНСЬКИЙ**

Прийняв:

доц. каф. ІСМ

**Сергій ЩЕРБАК**

Львів ‒ 2023

**Лабораторна робота №3**

**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.

Файл **art.py**

import pyfiglet  
  
  
class Art:  
 def \_\_init\_\_(self):  
 self.font = 'banner'  
 self.justify = 'center'  
  
 def \_create\_ascii\_art(self, text, width, character):  
 ascii\_art = pyfiglet.figlet\_format(text, justify=f'{self.justify}', font=self.font, width=width)  
  
 lines = ascii\_art.split('\n')  
  
 for i in range(len(lines)):  
 line = lines[i]  
 new\_line = ''  
 for char in line:  
 if char != ' ':  
 new\_line += character  
 else:  
 new\_line += ' '  
 lines[i] = new\_line  
  
 modified\_ascii\_art = '\n'.join(lines)  
  
 return modified\_ascii\_art

Файл **functions.py**

from colorama import Fore  
from labwork\_3.art import Art  
  
  
class Function(Art):  
  
 def \_choose\_font(self, user\_font):  
 if user\_font == '1':  
 return "5lineoblique"  
 elif user\_font == '2':  
 return "banner"  
 elif user\_font == '3':  
 return "slant"  
 elif user\_font == '4':  
 return "doom"  
 elif user\_font == '5':  
 return "cyberlarge"  
 elif user\_font == '6':  
 return "big"  
  
 def \_choose\_color(self, user\_color):  
 if user\_color == '1':  
 return Fore.BLACK  
 elif user\_color == '2':  
 return Fore.RED  
 elif user\_color == '3':  
 return Fore.GREEN  
 elif user\_color == '4':  
 return Fore.YELLOW  
 elif user\_color == '5':  
 return Fore.BLUE  
 elif user\_color == '6':  
 return Fore.WHITE  
 elif user\_color == '7':  
 return Fore.CYAN  
 elif user\_color == '8':  
 return Fore.MAGENTA  
  
 def \_scale\_ascii\_art\_height(self, art, height\_scale):  
 lines = art.split('\n')  
 scaled\_lines = []  
  
 for line in lines:  
 scaled\_lines.extend([line] \* height\_scale)  
  
 return '\n'.join(scaled\_lines)

Файл **variables.py**

user\_word\_input = "Please enter a word or phrase: "  
user\_font\_input = "Fronts: \n 1. '5lineoblique'\n 2. 'banner'\n 3. 'slant'\n 4. 'doom'\n 5. 'cyberlarge'\n 6. 'big'\nPlease choose font: "  
user\_color\_input = "Colors: \n 1. 'black'\n 2. 'red'\n 3. 'green'\n 4. 'yellow'\n 5. 'blue'\n 6. 'white'\n 7. 'cyan'\n 8. 'magenta'\nPlease choose color : "  
user\_choice\_input = "If you want to use your characters, please write 'custom', if not, you can enter anything or leave it out: "  
width\_input = "Please enter the width (optimal 100): "  
height\_input = "Please enter the height (optimal 1): "  
user\_character\_input = "What character you want use?: "  
error\_font = "Error: You have selected the wrong font number"  
error\_color = "Error: You have selected the wrong color number"  
error\_width = "Error: Cannot be less than 50"  
error\_height = "Error: Cannot be less than 0"  
choose\_exit = "Do you want to make another art? (Yes/No): "  
ask\_change\_art = "Do you want to change the art? (Yes/No): "  
file\_name = "Please enter file name: "

Файл **errors.py**

class Error:  
 def check\_widht(self, parameter):  
 try:  
 parameter = int(parameter)  
 if parameter < 50:  
 raise ValueError  
 except ValueError:  
 return ValueError()  
  
 def check\_height(self, parameter):  
 try:  
 parameter = int(parameter)  
 if parameter <= 0:  
 raise ValueError  
 except ValueError:  
 return ValueError()  
  
 def check\_font(self, number):  
 try:  
 number = int(number)  
 if number <= 0 or number > 6:  
 raise ValueError  
 except ValueError:  
 return ValueError()  
  
 def check\_color(self, number):  
 try:  
 number = int(number)  
 if number <= 0 or number > 8:  
 raise ValueError  
 except ValueError:  
 return ValueError()

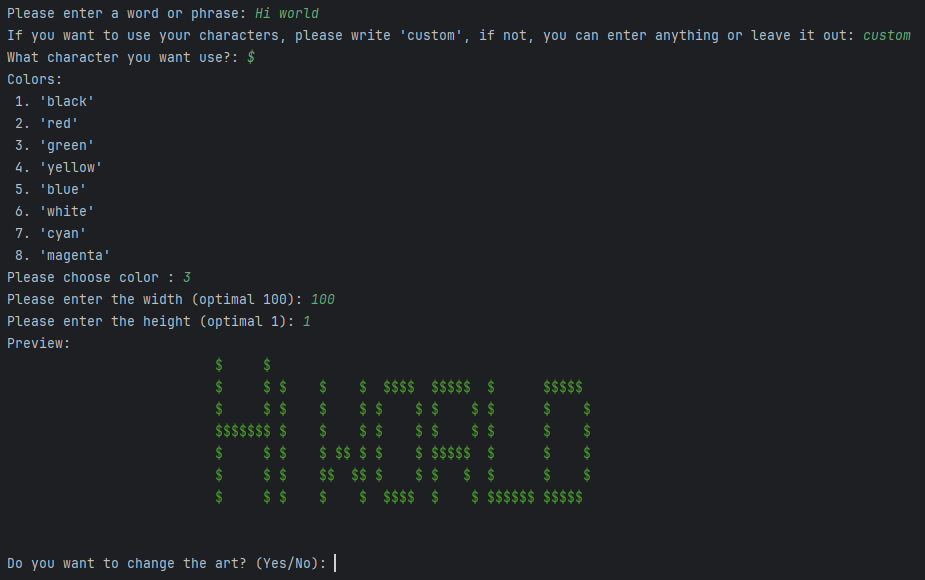
Файл **interface.py**

import pyfiglet  
from colorama import Style, Fore  
  
from labwork\_3 import variables  
from labwork\_3.art import Art  
from labwork\_3.errors import Error  
from labwork\_3.functions import Function  
  
  
class Interface(Function, Art):  
  
 def save\_to\_file(self, file\_name, user\_word\_modify):  
 file\_name = f"{file\_name}.txt"  
  
 with open(file\_name, "w") as file:  
 file.write(user\_word\_modify)  
  
 def get\_input\_size(self, number, error\_check, error\_message):  
 while True:  
 value = input(number)  
 if not error\_check(value):  
 return int(value)  
 print(Fore.RED + Style.BRIGHT + error\_message + Style.RESET\_ALL)  
  
 def get\_input\_settings(self, number, error\_check, error\_message):  
 while True:  
 value = input(number)  
 if not error\_check(value):  
 return value  
 print(Fore.RED + Style.BRIGHT + error\_message + Style.RESET\_ALL)  
  
 def \_choice(self, choice):  
 if choice.lower() != 'yes':  
 return True  
  
 def user\_input(self):  
 while True:  
 error = Error()  
 user\_word = input(variables.user\_word\_input)  
 user\_choice = input(variables.user\_choice\_input)  
 if user\_choice == 'custom':  
 user\_character = input(variables.user\_character\_input)  
 else:  
 user\_font = self.get\_input\_settings(variables.user\_font\_input, error.check\_font, variables.error\_font)  
  
 user\_color = self.get\_input\_settings(variables.user\_color\_input, error.check\_color, variables.error\_color)  
  
 user\_width = self.get\_input\_size(variables.width\_input, error.check\_widht, variables.error\_width)  
  
 user\_height = self.get\_input\_size(variables.height\_input, error.check\_height, variables.error\_height)  
  
 if user\_choice == 'custom':  
 user\_word\_modify = self.\_create\_ascii\_art(user\_word, width=user\_width, character=user\_character)  
 else:  
 user\_word\_modify = pyfiglet.figlet\_format(user\_word, font=self.\_choose\_font(user\_font), justify="center",  
 width=user\_width)  
  
 user\_word\_modify = self.\_scale\_ascii\_art\_height(user\_word\_modify, user\_height)  
  
 print(f"Preview:\n{self.\_choose\_color(user\_color)}{user\_word\_modify}{Style.RESET\_ALL}")  
  
 change\_selection = input(variables.ask\_change\_art)  
 if self.\_choice(change\_selection):  
 name\_file = input(variables.file\_name)  
 self.save\_to\_file(name\_file, user\_word\_modify)  
 else:  
 continue  
  
 choose\_exit = input(variables.choose\_exit)  
 if self.\_choice(choose\_exit):  
 break

Файл **main.py**

from labwork\_3.interface import Interface  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 interface = Interface()  
 interface.user\_input()

Результат виконання всіх завдань:





**Рис.1** Результат виконання

**Висновок:** Виконавши ці завдання, я розробив додаток Генератора ASCII-арту.