

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»
ІНСТИТУТ КОМП'ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА ІСМ



ЗВІТ

про виконання лабораторної роботи №4

з дисципліни

«Спеціалізовані мови програмування»

студента групи ІТ-32

Ткачишина Юрія

Прийняв Щербак С. С.

Львів - 2023

Мета роботи: Створення генератора ASCII-арту без використання зовнішніх бібліотек.

Індивідуальне завдання

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.

Хід виконання:

На рис. 1 зображено знімок екрану виконання програми.


```
break

else:

    print("Invalid choice. Please try again.")
```

Файл functions.py:

```
import os

import pyfiglet

from colorama import init, Fore, Style#, strip_ansi

import re
```

```
def create_art():

    text = input("Enter your phrase: ")

    font = input("Choose font style (standard, slant, shadow, etc.): ")

    color = input("Choose text color (red, blue, green, etc.): ")

    width = int(input("Enter width of ASCII art: "))

    height = int(input("Enter height of ASCII art: "))

    characters = input("Enter characters to use (default: '@#*'): ") or "@#*"

    try:

        ascii_art = pyfiglet.figlet_format(text, font=font)

        colored_art = f"{Fore.__dict__[color.upper()]} {ascii_art} {Style.RESET_ALL}"

        scaled_art = "\n".join([line.center(width) for line in
colored_art.splitlines()]).splitlines()

        final_art = "\n".join(["".join([char for char in line]) for line in scaled_art])

        return final_art, characters

    except Exception as e:

        print(f"Error: {e}")

        return None, None
```

```
# Function to save ASCII art to a file
```

```
def save_to_file(art, characters):
```

```
    filename = input("Enter the name of the file to save the art: ")
```

```
    try:
```

```
        with open(f"{filename}.txt", "w") as file:
```

```
            for line in art.splitlines():
```

```
                # Remove colorama control codes before writing to the file
```

```
                clean_line = "".join(char for char in line if char.isprintable())
```

```
                replaced_line = clean_line
```

```
                for i, char in enumerate(characters):
```

```
                    replaced_line = replaced_line.replace(f"@{i+1}", char)
```

```
                file.write(replaced_line + "\n")
```

```
    print(f"ASCII art saved to {filename}.txt")
```

```
    except Exception as e:
```

```
        print(f"Error: {e}")
```

```
# Function to view ASCII art from files
```

```
def view_art():
```

```
    try:
```

```
        files = sorted([file for file in os.listdir() if file.endswith('.txt')])
```

```
        for idx, file in enumerate(files):
```

```
            print(f"{idx + 1}. {file}")
```

```
        file_number = int(input("Enter the number of the file you want to view: ")) - 1
```

```
        with open(files[file_number], "r") as file:
```

```
            content = file.read()
```

```
        # Use regular expression to remove ANSI escape codes
```

```
        ansi_escape = re.compile(r'\x1B(?:[@-Z\\-_]|\\[[0-?]*[ -/]*[@-~])')
```

```
content = ansi_escape.sub("", content)

print(content)

except Exception as e:

print(f"Error: {e}")
```

Function to delete ASCII art files

```
def delete_art():

    try:

        files = sorted([file for file in os.listdir() if file.endswith('.txt')])

        for idx, file in enumerate(files):

            print(f"{idx + 1}. {file}")

            file_number = int(input("Enter the number of the file you want to delete: ")) - 1

            file_name = files[file_number]

            os.remove(file_name)

            print(f"{file_name} deleted successfully.")

        except Exception as e:

            print(f"Error: {e}")
```

Висновок: Виконуючи ці завдання, я створив генератор ASCII-арту з нуля, та надав можливість налаштовувати символи, розміри, вирівнювання та кольори, що дозволило мені глибше розібратися як створюється ASCII-арт.