

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

Виконав:
студент групи РІ-21сп
Владислав РИБАК

Львів – 2024

Мета виконання лабораторної роботи: Створення Генератора ASCII-арту без використання зовнішніх бібліотек.

План 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-y

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-y

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-y.

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.

Текст програмної реалізації:

ascii_generator.py:

```
import os
```

```
class ASCIIGenerator:
```

```
    def __init__(self):
```

```
        self.char_set = {}
```

```
        self.load_characters()
```

```
        self.width = 0
```

```
        self.height = 0
```

```
        self.text = ""
```

```
        self.alignment = "left"
```

```
    def load_characters(self):
```

```
        for char in
"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789!@#$%^
&*()_+~[]{}|;'\",.<>?/ ":
```

```
        if char.isalpha():
```

```
            filename = f"lab_4/alphabets/{char}.txt"
```

```
        elif char.isdigit():
```

```
            filename = f"lab_4/numbers/{char}.txt"
```

```
        else:
```

```
            filename = f"lab_4/symbols/{char}.txt"
```

```
        if os.path.exists(filename):
```

```
            with open(filename, 'r') as file:
```

```

        self.char_set[char] = [line.rstrip('\n') for line in file.readlines()]

    else:

        self.char_set[char] = [' ' * 10] * 10 # Default to blank if file doesn't exist


def get_user_input(self):

    self.text = input("Введіть слово або фразу для перетворення в ASCII-арт: ")

    self.width = int(input("Введіть ширину для кожного символу (за замовчуванням 10): ")
or 10)

    self.height = int(input("Введіть висоту для кожного символу (за замовчуванням 10): ")
or 10)


# Покращений UX для вирівнювання тексту
print("Оберіть вирівнювання тексту:")

print("1. Зліва")

print("2. По центру")

print("3. Справа")

alignment_option = input("Оберіть опцію (1, 2 або 3): ").strip()


if alignment_option == '1':

    self.alignment = 'left'

elif alignment_option == '2':

    self.alignment = 'center'

elif alignment_option == '3':

    self.alignment = 'right'

else:

    print("Невірний вибір. За замовчуванням використано вирівнювання зліва.")

```

```
self.alignment = 'left'
```

```
# Завдання 2-5: Генерація ASCII-арту
```

```
def generate_art(self):
```

```
    ascii_lines = [" " for _ in range(self.height)]
```

```
    for char in self.text:
```

```
        if char in self.char_set:
```

```
            char_lines = self.char_set[char]
```

```
        else:
```

```
            # Обробка невизначених символів за допомогою порожнього простору
```

```
            char_lines = [' ' * self.width] * self.height
```

```
    # Перевірка, чи є достатньо рядків у char_lines
```

```
    if len(char_lines) < self.height:
```

```
        char_lines += [' ' * self.width] * (self.height - len(char_lines))
```

```
    for i in range(self.height):
```

```
        ascii_lines[i] += char_lines[i] + ' ' # Додаємо один пробіл між символами
```

```
    return ascii_lines
```

```
# Завдання 6: Виведення арту
```

```
def display_art(self, ascii_art):
```

```
    for line in ascii_art:
```

```
        aligned_line = self._align_line(line, self.width * len(self.text))
```

```
        print(aligned_line)
```

```
def _align_line(self, line, width):
```

```
    if self.alignment == 'left':
```

```
        return line.ljust(width)
```

```
    elif self.alignment == 'center':
```

```
        return line.center(width)
```

```
    elif self.alignment == 'right':
```

```
        return line.rjust(width)
```

```
    else:
```

```
        return line
```

```
# Новий метод: Зберегти арт у файл
```

```
def save_art_to_file(self, ascii_art):
```

```
    filename = input("Введіть ім'я файлу для збереження ASCII-арту (наприклад, art.txt): ")
```

```
    with open(filename, 'w') as file:
```

```
        for line in ascii_art:
```

```
            file.write(line + '\n')
```

```
    print(f"ASCII-арт збережено в {filename}")
```

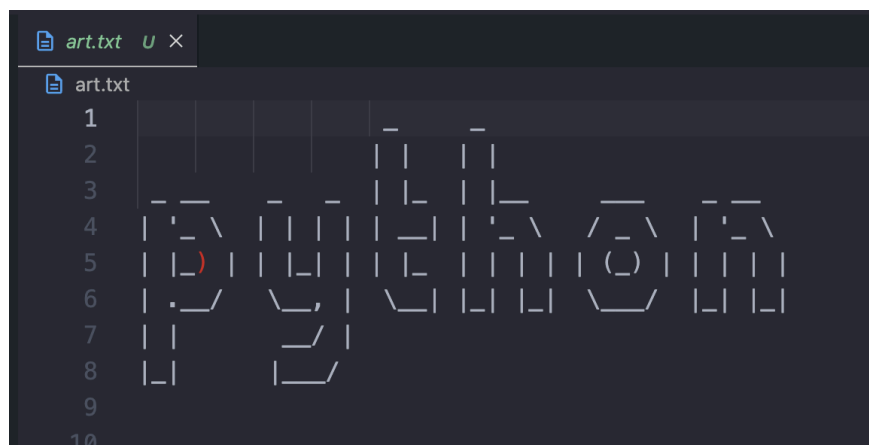
Результати тестування:

```
Введіть слово або фразу для перетворення в ASCII-арт: python
Введіть ширину для кожного символу (за замовчуванням 10):
Введіть висоту для кожного символу (за замовчуванням 10):
Оберіть вирівнювання тексту:
1. Зліва
2. По центру
3. Справа
Оберіть опцію (1, 2 або 3): 2

python

Do you want to save the ASCII art to a file?
1. Yes
2. No
Choose an option (1 or 2): █
```

Рис. 1. Результат виведення в консоль



The image shows a text editor window with a tab labeled 'art.txt'. The file contains the ASCII art of the word 'python' as generated in the previous step. The art is centered and uses a monospaced font. The word 'python' is rendered in a stylized, blocky font where each letter is composed of vertical and horizontal lines. The 'p' and 'y' have descenders. The 't' is a simple vertical line with a horizontal crossbar. The 'h' has a vertical stem and a curved top. The 'o' is a circle made of lines. The 'n' has a vertical stem and a curved top. The 'a' is a circle with a vertical line through the middle. The 't' is a simple vertical line with a horizontal crossbar. The 'h' has a vertical stem and a curved top. The 'o' is a circle made of lines. The 'n' has a vertical stem and a curved top. The 'a' is a circle with a vertical line through the middle.

Рис. 2. Результат збережений у файл

Висновки: на цій лабораторній роботі було створено Генератор ASCII-арту без використання зовнішніх бібліотек.