

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:
ст. гр. РІ-21сп
Рак В.П.
Прийняв:
Щербак С.С.

Львів-2024

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

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-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.

Виконання роботи

Текст програмної реалізації:

```
def get_user_input():
```

```
    """
```

```
    Запитує у користувача текст для ASCII-арту.
```

```
    Повертає:
```

```
        str: Текст, введений користувачем для створення ASCII-арту.
```

```
    """
```

```
    return input("Введіть текст для ASCII-арту: ").strip()
```

```
def get_art_dimensions():
```

```
    """
```

```
    Запитує у користувача ширину та висоту ASCII-арту.
```

```
    Повертає:
```

```
        tuple: Кортеж, що містить ширину та висоту ASCII-арту (int, int).
```

```
    """
```

```
    while True:
```

```
        try:
```

```
            width = int(input("Введіть ширину ASCII-арту (1-100): "))
```

```
            height = int(input("Введіть висоту ASCII-арту (1-30): "))
```

```
            if 1 <= width <= 100 and 1 <= height <= 30:
```

```
                return width, height
```

```
            else:
```

```
                print("Розміри повинні бути в межах: ширина (1-100), висота (1-30).")
```

```
        except ValueError:
```

```
            print("Будь ласка, введіть дійсні числа.")
```

```
# ASCII шаблони для літер
```

```
ASCII_LETTERS = {
```

```
    'A': [" @ ", " @ @ ", "@@@", "@ @", "@ @"],
```

```

'B': ["@@@@ ", "@ @", "@@@", "@ @", "@@@@ "],
'C': [" @@@ ", "@ @", "@ ", "@ @", " @@@ "],
'D': ["@@@@@ ", "@ @", "@ @", "@ @", "@@@@@ "],
'E': ["@@@@@@", "@ ", "@@@@ ", "@ ", "@@@@@@"],
'F': ["@@@@@@", "@ ", "@@@@ ", "@ ", "@ "],
'G': [" @@@ ", "@ @", "@ @@", "@ @", " @@@ "],
'H': ["@ @", "@ @", "@@@@@", "@ @", "@ @"],
'I': [" @@@ ", " @ ", " @ ", " @ ", " @@@ "],
'J': [" @ ", " @ ", " @ ", "@ @ ", "@@ "],
'K': ["@ @ ", "@ @ ", "@@@ ", "@ @ ", "@ @ "],
'L': ["@ ", "@ ", "@ ", "@ ", "@@@@@@"],
'M': ["@ @", "@@@@@", "@ @", "@ @", "@ @"],
'N': ["@ @", "@@ @", "@ @ @", "@ @@", "@ @"],
'O': [" @@@ ", "@ @", "@ @", "@ @", " @@@ "],
'P': ["@@@@@ ", "@ @", "@@@@ ", "@ ", "@ "],
'Q': [" @@@ ", "@ @", "@ @", "@ @@", "@@ @"],
'R': ["@@@@@ ", "@ @", "@@@@ ", "@ @ ", "@ @ "],
'S': [" @@@ ", "@ @", " @@@ ", " @", " @@@ "],
'T': ["@@@@@@", " @ ", " @ ", " @ ", " @ "],
'U': ["@ @", "@ @", "@ @", "@ @", " @@@ "],
'V': ["@ @", "@ @", "@ @", " @ @", " @ "],
'W': ["@ @", "@ @", "@ @", "@@@@@", " @ @"],
'X': ["@ @", " @ @", " @ ", " @ @", "@ @"],
'Y': ["@ @", " @ @", " @ ", " @ ", " @ "],
'Z': ["@@@@@@", " @ ", " @ ", " @ ", "@@@@@@"],
':': [" ", " ", " ", " ", " "] # Пробіл
}

```

```
def get_symbol_set():
```

```
    """
```

Запитує у користувача набір символів для ASCII-арту.

Повертає:

str: Набір символів для генерації ASCII-арту. Якщо користувач не введе символи, використовується за замовчуванням набір "@#*".

"""

```
symbols = input("Введіть набір символів для генерації ASCII-арту (за замовчуванням: '@ # *'): ").strip()
```

```
if not symbols:
```

```
    symbols = "@#*"
```

```
while len(symbols) < 3:
```

```
    symbols += symbols[-1]
```

```
return symbols
```

```
def choose_color_scheme():
```

"""

Запитує у користувача колірну схему.

Повертає:

str: Вибір користувача для колірної схеми (1 або 2).

"""

```
print("Доступні варіанти кольорів:")
```

```
print("1. Чорно-білий")
```

```
print("2. Відтінки сірого")
```

```
choice = input("Виберіть колірну схему (1 або 2): ").strip()
```

```
return choice
```

```
def generate_ascii_art(text, height, symbols, color_scheme):
```

"""

Генерує ASCII-арт на основі введення користувача, набору символів та висоти.

Аргументи:

text (str): Текст для генерації ASCII-арту.

height (int): Висота ASCII-арту.

symbols (str): Набір символів для використання в ASCII-арті.

color_scheme (str): Вибір колірної схеми.

Повертає:

list: Список рядків, що складають ASCII-арт.

"""

```
art_lines = [" for _ in range(height)]
```

```
for char in text.upper():
```

```
    if char in ASCII_LETTERS:
```

```
        letter = ASCII_LETTERS[char]
```

```
        letter = letter[:height]
```

```
        while len(letter) < height:
```

```
            letter.append(' ' * len(letter[0]))
```

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

```
        line = letter[i].replace('@', symbols[0]).replace('#', symbols[1]).replace('*', symbols[2])
```

```
        if color_scheme == "1":
```

```
            line = line.replace('@', symbols[0]).replace('#', symbols[1]).replace('*', symbols[2])
```

```
        elif color_scheme == "2":
```

```
            line = line.replace('@', '\033[0;37m' + symbols[0] + '\033[0m')
```

```
            line = line.replace('#', '\033[0;90m' + symbols[1] + '\033[0m')
```

```
        art_lines[i] += line + " "
```

```
    return art_lines
```

```
def align_art(art, width, alignment):
```

```
    """
```

Вирівнює текст ASCII-арту відповідно до обраного варіанту.

Аргументи:

art (list): Список рядків ASCII-арту.

width (int): Ширина, до якої потрібно вирівняти кожен рядок.

alignment (str): Вибір вирівнювання ("ліворуч", "по центру", "праворуч").

Повертає:

list: Список вирівняних рядків.

"""

```
aligned_lines = []
```

```
for line in art:
```

```
    if alignment == "ліворуч":
```

```
        aligned_lines.append(line.ljust(width))
```

```
    elif alignment == "по центру":
```

```
        aligned_lines.append(line.center(width))
```

```
    elif alignment == "праворуч":
```

```
        aligned_lines.append(line.rjust(width))
```

```
return aligned_lines
```

```
def display_art(art):
```

"""

Відображає ASCII-арт на екрані.

Аргументи:

art (list): Список рядків ASCII-арту для відображення.

"""

```
print("\nВаш ASCII-арт:")
```

```
print("\n".join(art))
```

```
def save_art_to_file(art):
```

"""

Зберігає ASCII-арт у файл.

Аргументи:

art (list): Список рядків ASCII-арту для збереження.

"""

```
filename = input("Введіть ім'я файлу для збереження (з розширенням .txt): ").strip()
```

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

```
    file.write("\n".join(art))
```

```
print(f'ASCII-арт збережено у файл '{filename}'.')
```

```
def choose_alignment():
```

```
    """
```

```
    Запитує у користувача варіант вирівнювання.
```

```
    Повертає:
```

```
        str: Вибір користувача для вирівнювання ("ліворуч", "по центру", "праворуч").
```

```
    """
```

```
    print("Варіанти вирівнювання: ліворуч, по центру, праворуч.")
```

```
    alignment = input("Виберіть вирівнювання: ").strip().lower()
```

```
    return alignment if alignment in ["ліворуч", "по центру", "праворуч"] else "ліворуч"
```

```
def preview_art(art):
```

```
    """
```

```
    Показує попередній перегляд ASCII-арту.
```

```
    Аргументи:
```

```
        art (list): Список рядків ASCII-арту для попереднього перегляду.
```

```
    """
```

```
    print("\nПопередній перегляд ASCII-арту:")
```

```
    print("\n".join(art))
```

```
def main():
```

```
    """
```

```
    Основна функція програми для створення ASCII-арту.
```

```
    Запитує вхідні дані від користувача, генерує ASCII-арт і надає можливість збереження.
```

```
    """
```

```
    text = get_user_input()
```

```
    width, height = get_art_dimensions()
```

```
    symbols = get_symbol_set()
```

```
    color_scheme = choose_color_scheme()
```

```
    art = generate_ascii_art(text, height, symbols, color_scheme)
```



```

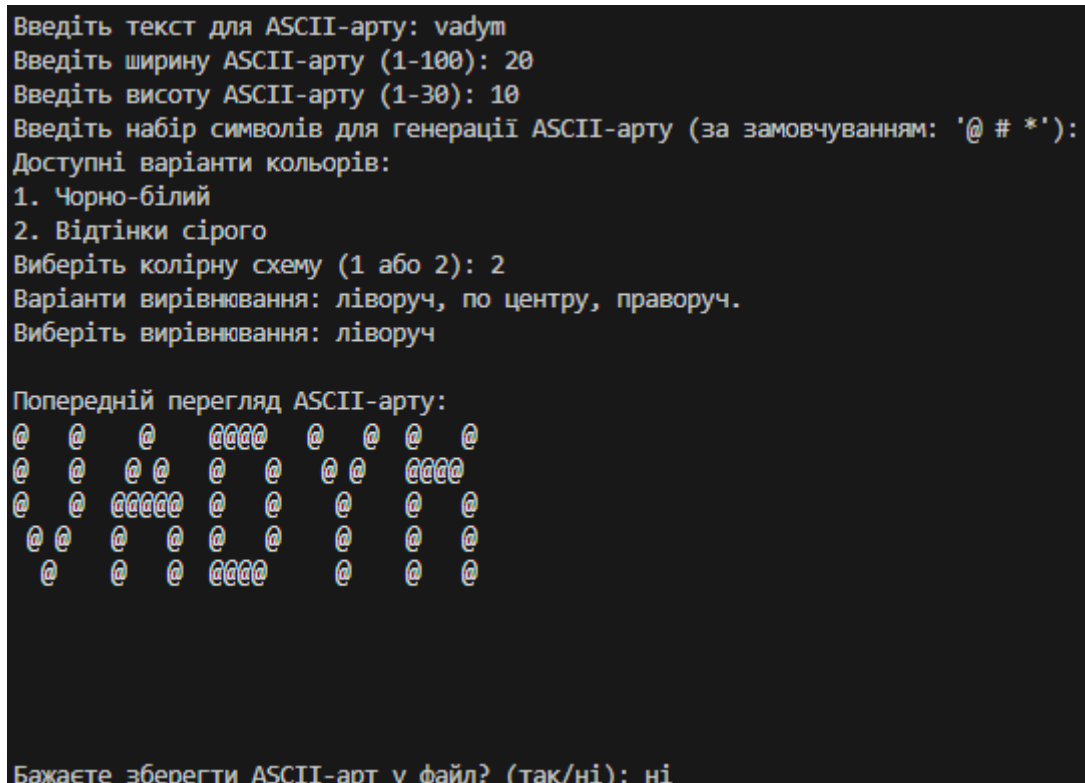
aligned_art = align_art(art, width, choose_alignment())
preview_art(aligned_art)

if input("Бажаєте зберегти ASCII-арт у файл? (так/ні): ").strip().lower() == "так":
    save_art_to_file(aligned_art)

if __name__ == "__main__":
    main()

```

Результат роботи програми:



```

Введіть текст для ASCII-арту: vadym
Введіть ширину ASCII-арту (1-100): 20
Введіть висоту ASCII-арту (1-30): 10
Введіть набір символів для генерації ASCII-арту (за замовчуванням: '@ # *'):
Доступні варіанти кольорів:
1. Чорно-білий
2. Відтінки сірого
Виберіть кольорну схему (1 або 2): 2
Варіанти вирівнювання: ліворуч, по центру, праворуч.
Виберіть вирівнювання: ліворуч

Попередній перегляд ASCII-арту:
@ @ @ @@@@ @ @ @ @
@ @ @ @ @ @ @ @ @@@@
@ @ @@@@ @ @ @ @ @
@ @ @ @ @ @ @ @ @
 @ @ @ @@@@ @ @ @

Бажаєте зберегти ASCII-арт у файл? (так/ні): ні

```

Рис. 1 - Робота програми

Висновок: У ході виконання лабораторної роботи я створив генератор ASCII-арту з нуля, який надає можливість налаштовувати символи, розміри, вирівнювання та кольори для арту, а також має можливість зберігати арт у файлі.