# **МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА” ІНСТИТУТ КОМП'ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ** **Кафедра ІСМ**

# **Звіт**

# **до лабораторної роботи №4**

# **З дисципліни “Спеціальні мови програмування”**

**Виконав:  
ст.гр.ІТ-31  
Шельвах Максим  
  
Прийняв:  
Щербак С.С.**

# **Львів — 2023**

**Тема роботи:** Розробка ASCII ART генератора для візуалізації 2D-фігур.

**Мета 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-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 роботи

**Реалізований код:**

ascii\_former.py

from ASCII\_art\_generator.generator.ascii\_art import generate\_custom\_text, replaceSymbol, paintText

class ASCIIArt:

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

self.text = ""

self.font = ""

def setText(self, text):

self.text = text

def setFont(self, font):

self.font = font

def generateArt(self):

return generate\_custom\_text(input\_string=self.text, font\_name=self.font)

class Art(ASCIIArt):

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

super().\_\_init\_\_()

self.symbol\_to\_replace = ""

self.color = ""

def setSymbolToReplace(self, symbol\_to\_replace):

self.symbol\_to\_replace = symbol\_to\_replace

def setColor(self, color):

self.color = color

def generateASCIIArt(self):

generated\_art = self.generateArt()

text\_art\_replaced = replaceSymbol(

generated\_art, self.symbol\_to\_replace)

text\_art\_painted = paintText(

text\_art\_replaced, self.color)

return text\_art\_painted

interface.py

from ASCII\_art\_generator.ascii\_former import Art

def artFileSaver(filename, text):

with open(filename, "w") as file:

file.write(text)

print(f"text was saved into {filename}")

class ASCIIArtInterface:

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

self.art = Art()

def setPrimaryData(self):

text = input("Enter your text: ")

self.art.setText(text)

font = input("Choose art font (small, medium, big): ")

self.art.setFont(font)

symbol\_to\_replace = input(

"Pick art symbol: ")

self.art.setSymbolToReplace(symbol\_to\_replace)

color = input("Choose art color: ")

self.art.setColor(color)

save\_text = input("Do you want to save the text? (yes/no): ").lower()

if save\_text == "yes":

self.saveToFile()

def saveToFile(self):

filename = input("Enter filename for saving art: ")

artFileSaver(filename, self.art.generateASCIIArt())

def launch(self):

self.setPrimaryData()

print(self.art.generateASCIIArt())

main.py

from interface.interface import ASCIIArtInterface

def main():

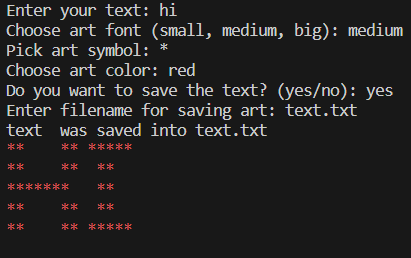
custon\_text\_art\_interface = ASCIIArtInterface()

custon\_text\_art\_interface.launch()

if \_\_name\_\_ == '\_\_main\_\_':

main()

**Результат виконання програми:**



**Висновок:** виконавши 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.