Міністерство освіти і науки України  
Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій

Кафедра інформаційних систем та мереж

******

**ЗВІТ**

**ДО ЛАБОРАТОРНОЇ РОБОТИ №4  
з дисципліни  
«Спеціалізовані мови програмування»  
на тему:**  
«Розробка ACII ART генератор для візуалізації 2D фігур»

***Виконала:*** *студентка групи РІ-31  
 Діана ВІЙЧУК*

***Прийняв:****к. т. н., доцент* *Сергій ЩЕРБАК*

## МЕТА РОБОТИ

Cтворення додатка для малювання 2D-фігур у ASCII-арті на основі об’єктно - орієнтованого підходу та мови Python.

## ЗАВДАННЯ

Завдання 1: Проектування класів

Розробіть структуру класів для вашого генератора 3D ASCII-арту. Визначте основні компоненти, атрибути та методи, необхідні для програми.

Завдання 2: Введення користувача

Створіть методи у межах класу для введення користувача та вказання 3D-фігури, яку вони хочуть намалювати, та її параметрів (наприклад, розмір, кольори).

Завдання 3: Представлення фігури

Визначте структури даних у межах класу для представлення 3D-фігури. Це може включати використання списків, матриць або інших структур даних для зберігання форми фігури та її властивостей.

Завдання 4: Проектування з 3D в 2D

Реалізуйте метод, який перетворює 3D-представлення фігури у 2D-представлення, придатне для ASCII-арту.

Завдання 5: Відображення ASCII-арту

Напишіть метод у межах класу для відображення 2D-представлення 3D-фігури як ASCII-арту. Це може включати відображення кольорів і форми за допомогою символів ASCII.

Завдання 6: Інтерфейс, зрозумілий для користувача

Створіть зручний для користувача командний рядок або графічний інтерфейс користувача (GUI) за допомогою об'єктно-орієнтованих принципів, щоб дозволити користувачам спілкуватися з програмою.

Завдання 7: Маніпуляція фігурою

Реалізуйте методи для маніпулювання 3D-фігурою, такі масштабування або зміщення, щоб надавати користувачам контроль над її виглядом.

Завдання 8: Варіанти кольорів

Дозвольте користувачам вибирати варіанти кольорів для їхніх 3D ASCII-арт-фігур. Реалізуйте методи для призначення кольорів різним частинам фігури.

Завдання 9: Збереження та експорт

Додайте функціональність для зберігання згенерованого 3D ASCII-арту у текстовий файл

Завдання 10: Розширені функції

Розгляньте можливість додавання розширених функцій, таких як тінь, освітлення та ефекти перспективи, для підвищення реалізму 3D ASCII-арту.

## ПРОГРАМНА РЕАЛІЗАЦІЯ

Файл *runner.py* .

from src.main import main

if \_\_name\_\_ == "\_\_main\_\_":

main()

Файл */src/bll/classes/threeD\_art\_generator.py* .

import sys

import os

lab5\_root = os.path.abspath(os.path.join(os.path.dirname(\_\_file\_\_), "..", ".."))

sys.path.append(lab5\_root)

from src.bll.classes.figures.Cube import Cube

from src.bll.classes.figures.Pyramid import Pyramid

from shered.dal.filer import save\_to\_file

from shered.validation import validate\_color\_choice

class ThreeDArtGenerator:

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

self.art = None

self.color = 32

self.direction = True

def choose\_shape(self):

shape = input("Виберіть фігуру (1 - Куб, 2 - Пірамід): ")

size = int(input("Введіть розмір: "))

if shape == '1':

self.art = Cube(size)

elif shape == '2':

self.art = Pyramid(size)

else:

print("Невірний вибір. Спробуйте ще раз.")

return self.choose\_shape()

def change\_size(self):

try:

size = int(input("Введи новий розмір (не менше 3): "))

if size < 3:

print("Розмір має бути не менше 3.")

return self.change\_size()

if self.art:

self.art.size = size

else:

print("Спочатку виберіть фігуру.")

except ValueError:

print("Неправильний вибір, спробуй ще раз")

self.change\_size()

def get\_available\_colors(self):

return ['red', 'green', 'yellow', 'blue', 'magenta', 'cyan', 'white']

def choose\_color(self):

colors = self.get\_available\_colors()

print("Доступні кольори:", colors)

chosen\_color = input("Виберіть колір з наведених: ").strip().lower()

self.color = validate\_color\_choice(chosen\_color, colors)

color\_map = {

'red': 31,

'green': 32,

'yellow': 33,

'blue': 34,

'magenta': 35,

'cyan': 36,

'white': 37

}

self.color = color\_map[chosen\_color]

def change\_color(self):

color = int(input("Введіть код кольору (30-37): "))

if 30 <= color <= 37:

self.color = color

else:

print("Неправильний код кольору. Спробуйте ще раз.")

self.change\_color()

def change\_direction(self):

self.direction = not self.direction

print("Напрям успішно змінено")

def get\_art(self) -> str:

color\_text = '\033[%dm%s\033[0m'

art = self.art.get\_three\_d\_art()

colored\_art = ""

for line in art.split('\n'):

colored\_art += color\_text % (self.color, line) + '\n'

return colored\_art

def print\_art(self):

if self.art:

print(self.get\_art())

else:

print("Спочатку виберіть фігуру.")

def save\_art\_into\_file(self):

if self.art:

filename = input("Введіть ім'я файлу для збереження (без розширення): ") + ".txt"

save\_to\_file(self.get\_art(), filename)

else:

print("Спочатку виберіть фігуру.")

def run(self):

self.choose\_shape()

self.choose\_color()

self.print\_art()

save\_choice = input("Бажаєте зберегти зображення у файл? (y/n): ").strip().lower()

if save\_choice == 'y':

self.save\_art\_into\_file()

else:

print("Зображення не буде збережено.")

Файл */src/bll/classes/figures/BaseFigure.py* .

from abc import ABC, abstractmethod

class BaseFigure(ABC):

def \_\_init\_\_(self, size: int):

self.size = size

@abstractmethod

def get\_two\_d\_art(self):

pass

@abstractmethod

def get\_three\_d\_art(self):

pass

Файл */src/bll/classes/figures/BaseFigure.py* .

from bll.classes.figures.BaseFigure import BaseFigure

class Cube(BaseFigure):

def \_\_init\_\_(self, side\_a: int):

super().\_\_init\_\_(side\_a)

self.side\_a = side\_a

def get\_two\_d\_art(self):

result = ""

for side\_a\_pixel in range(self.side\_a):

result += "\* "

for side\_b\_pixel in range(self.side\_a - 2):

if side\_a\_pixel % self.side\_a == 0 or side\_a\_pixel + 1 == self.side\_a:

result += "\* "

else:

result += " "

result += "\*\n"

return result

def get\_three\_d\_art(self):

result = ""

art\_size = int(self.side\_a + self.side\_a / 2)

side\_side\_size = int(self.side\_a / 1.5)

space\_counter = 0

for x in range(art\_size):

if x == 0:

result += "\* "

elif x < side\_side\_size:

result += "\* " + space\_counter \* " " + "\* "

space\_counter += 1

elif side\_side\_size <= x <= art\_size - side\_side\_size:

result += "\* " + (side\_side\_size - 2) \* " " + "\* "

space\_counter = 1

elif x < art\_size - 1:

result += space\_counter \* " " + "\* " + (art\_size - x - 2) \* " " + "\* "

space\_counter += 1

else:

result += space\_counter \* " " + "\* "

for y in range(self.side\_a - 2):

if x == 0 or x == side\_side\_size - 1 or x == art\_size - 1:

result += "\* "

else:

result += " "

result += "\*\n"

return result

Файл */src/bll/classes/figures/Pyramid.py* .

import math

from bll.classes.figures.BaseFigure import BaseFigure

class Pyramid(BaseFigure):

def \_\_init\_\_(self, radius: int):

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

self.radius = radius

def get\_two\_d\_art(self):

result = ""

for h in range(-self.radius, self.radius):

row = ""

for w in range(-self.radius, self.radius):

dist = math.sqrt(h \*\* 2 + w \*\* 2)

if dist < self.radius - 0.5:

row += "\*"

elif dist < self.radius:

row += "."

else:

row += " "

result += row + "\n"

return result

def get\_three\_d\_art(self):

result = ""

levels = self.radius

for i in range(levels):

padding = ' ' \* (levels - i - 1)

main\_layer = '\*' \* (2 \* i + 1)

side\_layer = '/' if i < levels - 1 else ''

layer = f"{padding}{main\_layer} {side\_layer}"

result += layer + "\n"

return result

Файл */src/main.py* .

import sys

import os

lab5\_root = os.path.abspath(os.path.join(os.path.dirname(\_\_file\_\_), "..", ".."))

sys.path.append(lab5\_root)

from src.ui.menu import display\_menu, get\_user\_choice

from src.ui.menuBuilder import run\_three\_d\_art\_generator

def main():

print("\nЛаскаво просимо до генератора 3D ASCII-арту!")

while True:

display\_menu()

choice = get\_user\_choice()

if choice == '1':

run\_three\_d\_art\_generator()

elif choice == '2':

print("До побачення!")

break

else:

print("Невірний вибір. Спробуйте ще раз.")

if \_\_name\_\_ == "\_\_main\_\_":

main()

## ВИСНОВКИ

На цій лабораторній роботі я створила додаток для малювання 3D-фігур у ASCII-арті на основі об’єктно - орієнтованого підходу та мови Python.

Посилання на GitHub: <https://github.com/Dinasi4ka/Calculator.git>