МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ



**Дніпровський національний університет  
залізничного транспорту імені академіка В. Лазаряна**

Кафедра «Комп’ютерні інформаційні технології»

**Лабораторна робота №7**

**з дисципліни «Об’єктно-орієнтоване програмування»**

**на тему: «****Спадкування, поліморфізм»**

Виконав: студент гр. ПЗ2011

Кулик Сергій Вадимович

Прийняла:

Демидович Інна Миколаївна

Дніпро, 2021

**Тема:** Спадкування, поліморфізм

**Мета:** Навчитися працювати із спадкуванням та використовувати поліморфізм.

**Завдання**

Для будь-якої предметної області реалізувати патерн компонувальник.

**Код програми**

#include <iostream>

#include <vector>

using namespace std;

//Composite

class Shape

{

public:

virtual void Draw() = 0;

};

class Square : public Shape

{

public:

void Draw() override

{

cout << "I'm a square" << endl;

}

};

class Triangle : public Shape

{

public:

void Draw() override

{

cout << "I'm a triangle" << endl;

}

};

class Circle : public Shape

{

public:

void Draw() override

{

cout << "I'm a circle" << endl;

}

};

class ShapeComposite :public Shape

{

private:

vector<Shape\*> components;

public:

void addComponent(Shape\* component)

{

components.push\_back(component);

}

void removeComponent()

{

components.pop\_back();

}

void Draw() override

{

for (int i = 0; i < components.size(); i++)

{

components[i]->Draw();

}

}

};

int main()

{

ShapeComposite a, b, c, d;

Square square1;

Circle c1;

Triangle t1;

a.addComponent(&t1);

b.addComponent(&c1);

c.addComponent(&square1);

c.Draw();

d.addComponent(&b);

d.addComponent(&t1);

d.removeComponent();

d.Draw();

}

**Результат роботи**

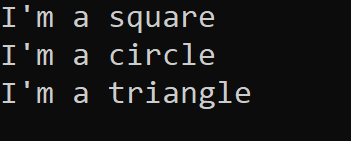
****

Рис. 1 – Результат роботи

**Висновок**

Під час виконнаня лабораторної роботи працював з чисто віртуальними функціями, реалізовував шаблон компнувальник. Від дозволяє об’єднувати об’єкти в деревовидну структуру, та дозволяє звертатися як до кожного об’єкта окремо так і до всіх разом. Чисто віртуальні функції дозволяють створити абстрактний клас, який буде шаблоном для написання інших подібних класів нащадків.