

Лабораторна робота № 4

Тема: Відмова від спадку. Альтернативні класи з різними інтерфейсами

Мета: Рефакторинг коду коли підклас використовує лише малу частину успадкованих методів і властивостей суперкласа.

Завдання:

1. У вас є базовий клас `Animal`, який має різні методи, і підклас `Dog`, який успадковує від базового класу. Ваше завдання - переписати код так, щоб відмовитися від спадкування та замість цього використовувати делегування для виклику методів базового класу..

```
class Animal:
    def speak(self):
        pass

    def move(self):
        pass

class Dog(Animal):
    pass

# Приклад використання класів
dog = Dog()
dog.speak()
dog.move()
shape.calculate_area()
```

Після рефакторингу підклас `Dog` повинен викликати відповідні методи базового класу `Animal` через делегування.

2. У вас є два класи Square і Circle, які виконують однакову функцію - обчислення площі фігури, але мають різний інтерфейс. Ваше завдання - переписати код так, щоб обидва класи мали спільний інтерфейс

```
class Square:
    def area_square(self, side):
        return side ** 2

class Circle:
    def area_circle(self, radius):
        return 3.14 * radius ** 2

# Приклад використання класів
square = Square()
print("Area of square:", square.area_square(5))

circle = Circle()
print("Area of circle:", circle.area_circle(3))
```

Після рефакторингу класи Square і Circle повинні мати спільний інтерфейс, наприклад, метод area(), який приймає необхідні параметри для обчислення площі фігури.

Звіт:

Звіт має включати в себе два файли:

1. **output_1.py** - код після рефакторингу з п.1
2. **output_2.py** - код після рефакторингу з п.2

За необхідності можна додати файл **README.md** для опису прийнятих рішень щодо рефакторінгу.

Робота може бути виконана на будь якій мові програмування або на псевдокодi.