

5. Prototype Design pattern

Create a new object by copying an existing object. instead of creating from scratch.

- Clone an existing object.
- Avoid expensive object creation
- Create obj without knowing their exact class.

Why Prototype exists:-

Sometimes object creation is:

- costly (DB calls, network calls)
- complex and repetitive.

Instead of `new Object()` we do `existingObject.clone()`;

- creation is delegate to object itself.
- client doesn't know concrete class details.

interface Shape extends Cloneable {
 Shape clone();
}

class Circle implements Shape {

int radius

String color;

Circle (int radius, String color) {

this.radius = radius;

this.color = color;

}
 public Shape clone() {
 return new Circle (this.radius, this.color);
 }
}

Shallow copy.

- Copies object reference.
- Nested obj are shared.

Prototype pattern creates new object by cloning existing ones, improving performance and flexibility.

Client Code

Shape original = new
 Circle (10, "Red");

Shape copy = ~~original~~ original.clone();

Deep copy

- Copies everything recursively.
- Fully independent object.