Prototype

Creational Design Pattern

**What is a Prototype:**

1. A complex object that is costly to create. To create more instances of such a class, we use an existing instance as our prototype.
2. Prototype will allow us to make copies of an existing object & save us from having to recreate objects from scratch.

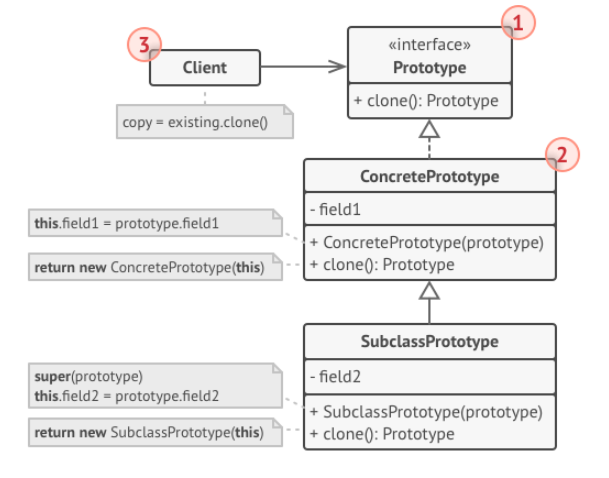
**Problem:**

1. You want to create an exact copy of an object. But not all objects can be copied easy because some of the object’s fields may be private and not visible from outside the object itself.

**Solution:**

1. Prototype pattern delegates the cloning process to the actual objects that are being cloned. Declares a common interface for all objects that support cloning.

**UML:**



1. Prototype interface declares the cloning methods.
2. Concrete Prototype class implements the cloning method.
3. Clients can produce a copy of any object that follows the prototype interface.

**Implementation:**

1. Create an abstract class implementing Clonable interface.
2. Create concrete classes extending the abstract class.
3. Implement reset method that changes the variables of a concrete class to their default values.