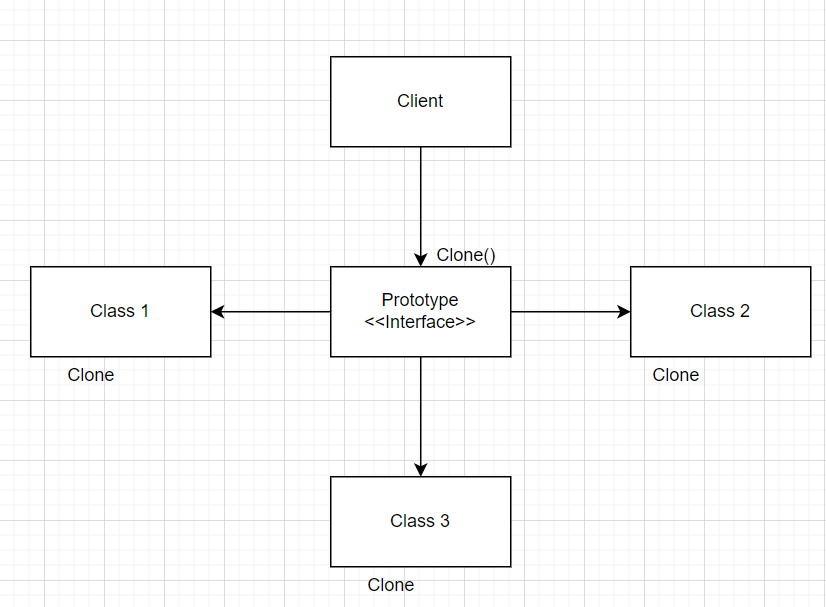
Design Patterns

1. Creational Design Pattern
2. Factory Method
3. Abstract Factory
4. Builder
5. Prototype

* It is used when we have to make copy/clone from existing object.



Code E.g.

public interface Prototype {

Prototype Clone();

}

public class Student : Prototype {

int age;

private rollNumber;

string name;

Student(int age, private rollNumber, string name) {

this. age = age;

this.rollNumber = this.rollNumber;

this.name = name;

}

@Override

Public Prototype Clone() {

return new Student(age, rollNumber, name);

}

}

1. Singleton
2. Structural Design Pattern
3. Adapter
4. Bridge
5. Composite
6. Decorator
7. Flyweight
8. Proxy
9. Behavioral Design Pattern
10. Chain of Responsibility
11. Command
12. Iterator
13. Mediater
14. Memonto
15. Observer
16. State
17. Strategy
18. Visitor
19. Template Method

Design Principles

SOLID

1. Single Responsibility Principle
2. Open-Closed Principle
3. Liskov Substitution Principle
4. Interface Segregation Principle
5. Dependency Inversion Principle