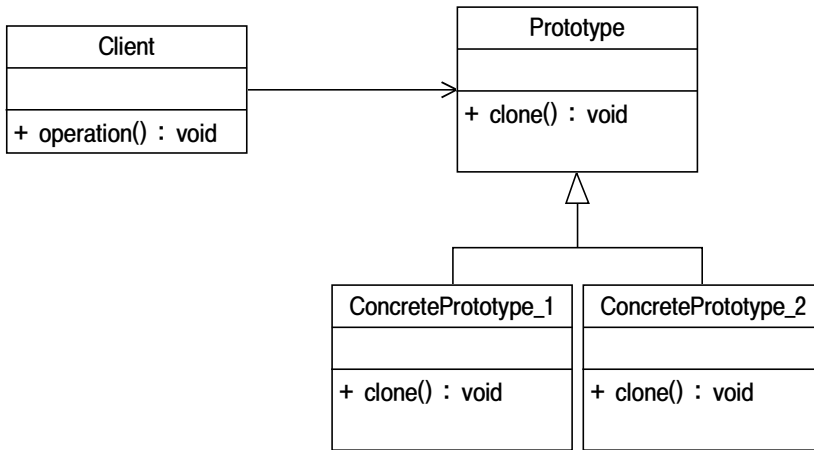


# 1. 원형 패턴

원형 패턴(prototype pattern)은 미리 만들어진 객체를 복제(clone)하는 패턴이다.

〈원형 패턴〉



〈위키백과 참조〉

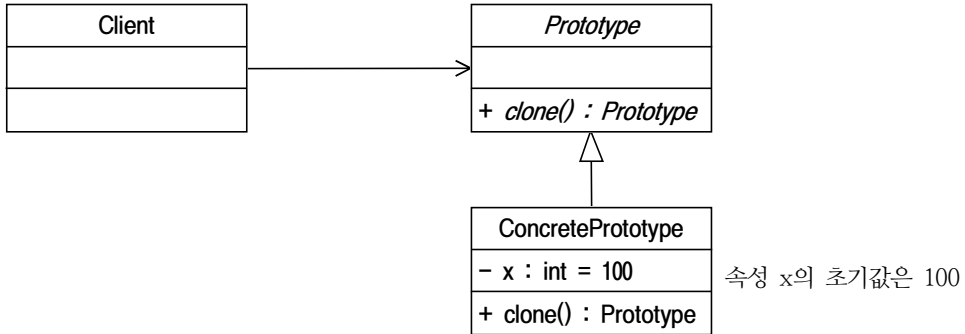
// 원형 패턴 등장인물

Prototype	<ul style="list-style-type: none"> <li>• 객체(instance)를 복사하여 새로운 객체를 생성하는 메서드 결정한다.</li> <li>• 주어진 그림에서 메서드 clone()이 선언되었다.</li> <li>• <b>Prototype</b>은 인터페이스나 추상클래스로 구현할 수 있다.</li> </ul>
ConcretePototype	<ul style="list-style-type: none"> <li>• 객체를 복사하는 <b>Prototype</b>을 구현한다.</li> <li>• 객체를 복사하는 메서드 clone()을 실제로 구현한다.</li> </ul>
Client	<ul style="list-style-type: none"> <li>• 객체 복사를 위해 동작을 시작하는 클래스</li> <li>• 객체 원형을 생성하고, 객체 복사를 요청한다.</li> </ul>

- 원형 패턴은 객체 생성에 원형이 되는 **견본을 이용하는 패턴**이다.
- 원형 패턴은 다수의 객체를 생성하는 경우에 **비용**을 효과적으로 줄일 수 있다.  
→ new를 이용하여 객체를 **생성**하는 것보다 생성된 객체를 **복제**하는 것이 더 빠르다.
- 원형 패턴은 자식클래스 수를 줄인다.(**새로운 상속**을 하지 않고, 원형을 복제)
- 원형 패턴은 **런타임**에 새로운 객체를 추가하고 삭제할 수 있다.
- 원형 패턴은 **속성값**을 다양화 하는 방법으로 새로운 객체를 명세할 수 있다.
- 원형 패턴은 **구조**를 다양화 하는 방법으로 새로운 객체를 명세할 수 있다.

[예제] 원형 패턴 - 객체를 복제하는 간단한 프로그램

〈원형 패턴 응용〉



↓  
↓ 자바로 구현하면  
↓

〈원형 패턴 응용프로그램〉

```
abstract class Prototype //추상클래스
{
    abstract public Prototype clone(); //추상메서드 clone() - 복제 메서드
}
class ConcretePrototype extends Prototype //추상클래스 상속
{
    private int x = 100; //원형(건본품)
    public ConcretePrototype(){ System.out.println("원형 : " + x); } //생성자
    public ConcretePrototype(ConcretePrototype p){ ++x; } //복제하면서 x값 1 증가
    public Prototype clone(){ return new ConcretePrototype(this); } //객체 자신 복제
}
public class Client
{
    public static void main(String args[])
    {
        Prototype p1 = new ConcretePrototype(); //new를 이용하여 객체 p1을 생성
        Prototype p2 = p1.clone(); //객체 p2는 p1을 복제(new를 사용하지 않음)
    }
}
```

// 복제한 것을 사용하는 경우

```

abstract class Prototype                                //추상클래스
{
    abstract public Prototype clone();                //추상메서드 clone() - 복제 메서드
    abstract public int get();
}

class ConcretePrototype extends Prototype             //추상클래스 상속
{
    private int x = 100;                               //원형(견본품)
    public ConcretePrototype(){                       //기본생성자
    public ConcretePrototype(ConcretePrototype p){ ++x; } //복제하면서 x값 1 증가
    public Prototype clone(){ return new ConcretePrototype(this); } //객체 자신 복제
    public int get(){ return x; }
}

public class Client
{
    public static void main(String args[])
    {
        Prototype p1 = new ConcretePrototype(); //new를 이용하여 객체 p1을 생성
        Prototype p2 = p1.clone();              //객체 p2는 p1을 복제(new를 사용하지 않음)
        System.out.println(p1.get()); //출력 : 100 - 원형
        System.out.println(p2.get()); //출력 : 101 - 복제
    }
}

```

위의 프로그램은 복제하면서 속성 x값 1 증가시킨 것이 주 특징이다.

원형 객체 p1의 x = 100

복제 객체 p2의 x = 101