

DESIGN PATTERNS

[Young B. Park \(ybpark@dankook.ac.kr\)](mailto:ybpark@dankook.ac.kr)



POLYMORPHISM

```
1. public interface Pet {
2.     public void cry();
3. }

4. public class Dog implements Pet{
5.     public void cry(){
6.         bark();
7.     }
8.     private void bark() {
9.         System.out.println("멍멍!");
10.    }
11. }

12. public class Cat implements Pet{
13.     public void cry(){
14.         meow();
15.     }
16.     private void meow(){
17.         system.out.println("야옹!");
18.     }
19. }
```

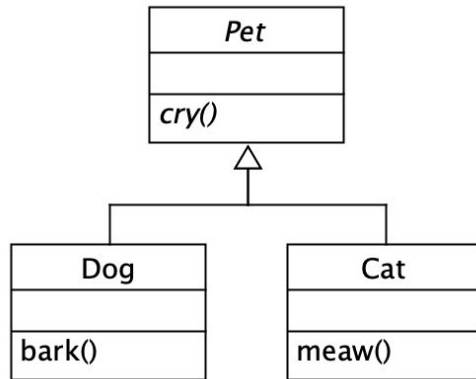




POLYMORPHISM

다형성을 활용한 코드 ↔ 암기하기 ☆☆☆

```
1. public class PetMain() {  
2.     public static void main(String[] args) {  
3.         Pet myPet;  
4.         myPet = new Dog();  
5.         myPet.cry();  
6.         myPet = new Cat();  
7.         myPet.cry();  
8.     }  
9. }
```





STRATEGY PATTERN

1. Goal: make code more understandable and/or more flexible.
2. Design Critiques 실습

SIMPLE SIMULATION OF DUCK BEHAVIOR

