

java

21강_인터페이스

객체가 다양한 데이터 타입을 가질 수 있는 방법에 대해서 학습합니다.

21-1 인터페이스란?

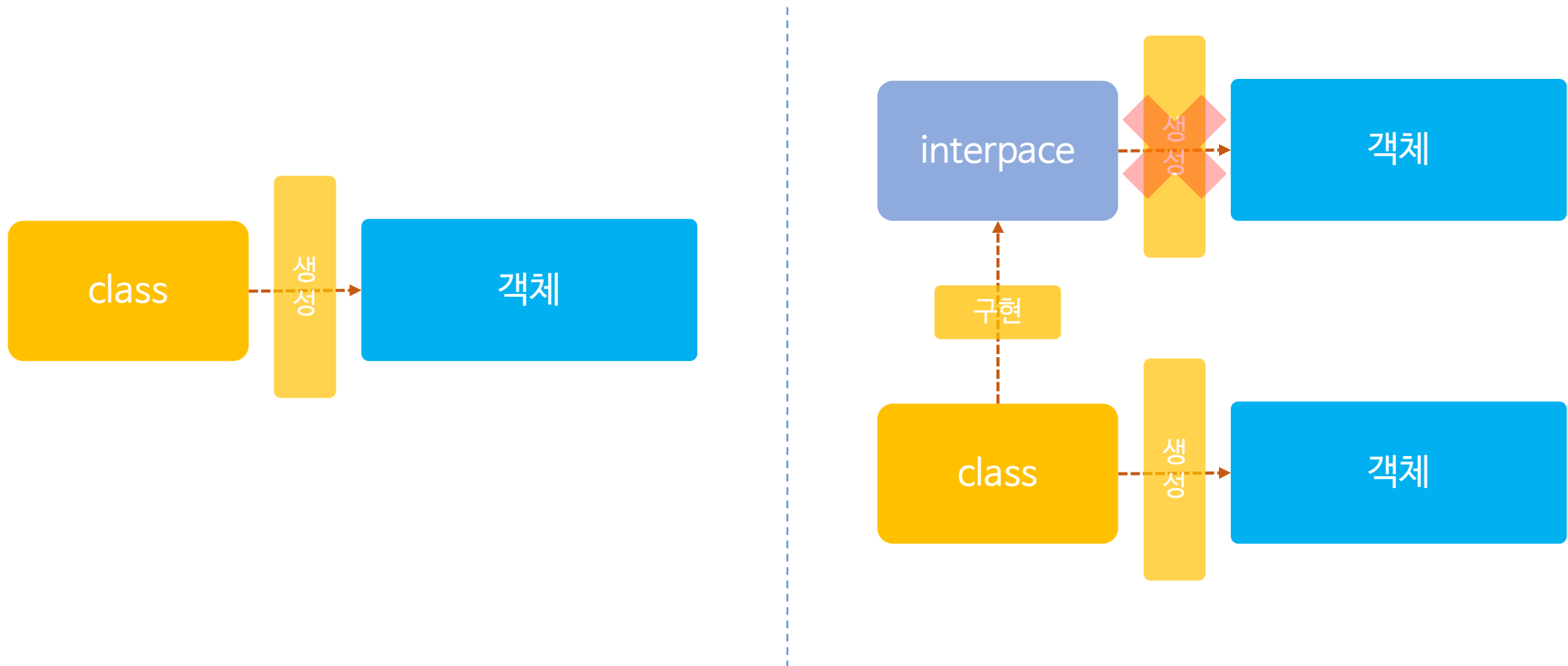
21-2 인터페이스를 사용하는 이유

22-3 인터페이스 구현

22-4 장난감 인터페이스

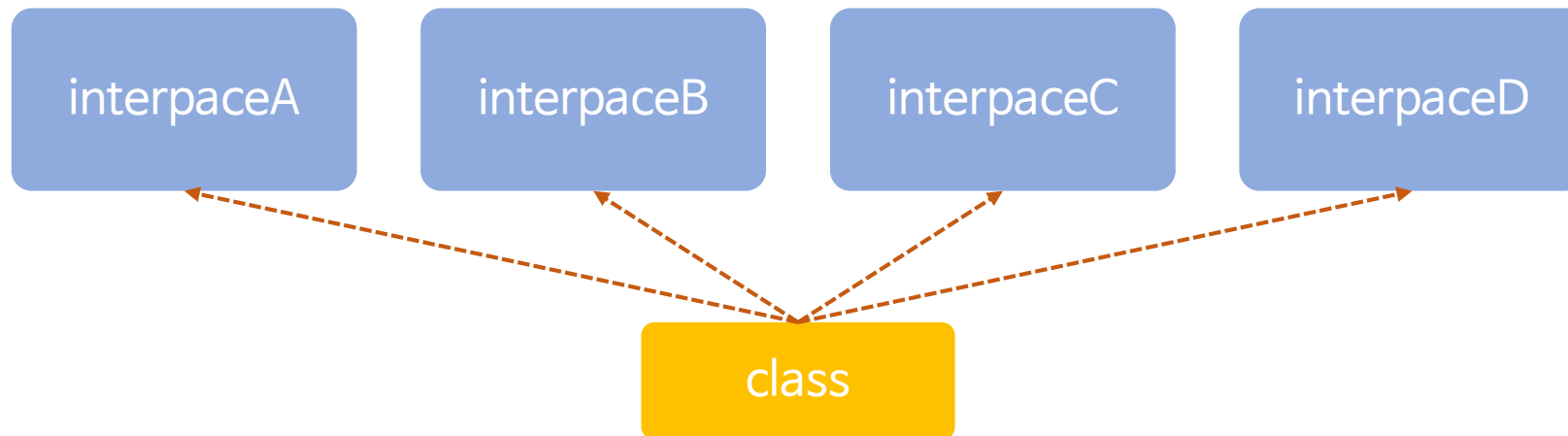
21-1 : 인터페이스란?

클래스와 달리 객체를 생성할 수는 없으며, 클래스에서 구현해야 하는 작업 명세서이다.



21-2 : 인터페이스를 사용하는 이유

인터페이스를 사용하는 이유는 많지만, 가장 큰 이유는 객체가 다양한 자료형(타입)을 가질 수 있기 때문이다.



```

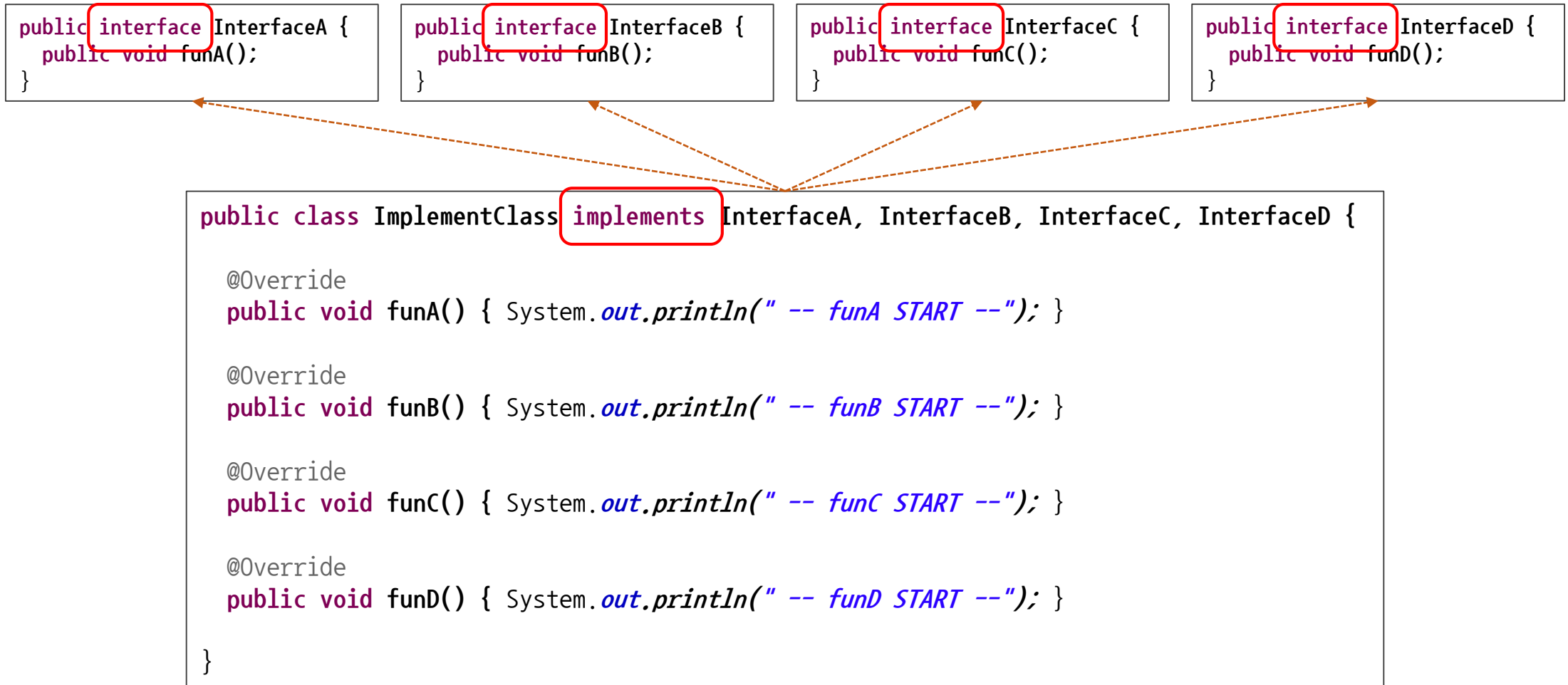
public class ImplementClass implements InterfaceA, InterfaceB, InterfaceC, InterfaceD {
    public ImplementClass() {
        System.out.println("ImplementClass constructor");
    }
}
  
```

```

InterfaceA ia = new ImplementClass();
InterfaceB ib = new ImplementClass();
InterfaceC ic = new ImplementClass();
InterfaceD id = new ImplementClass();
  
```

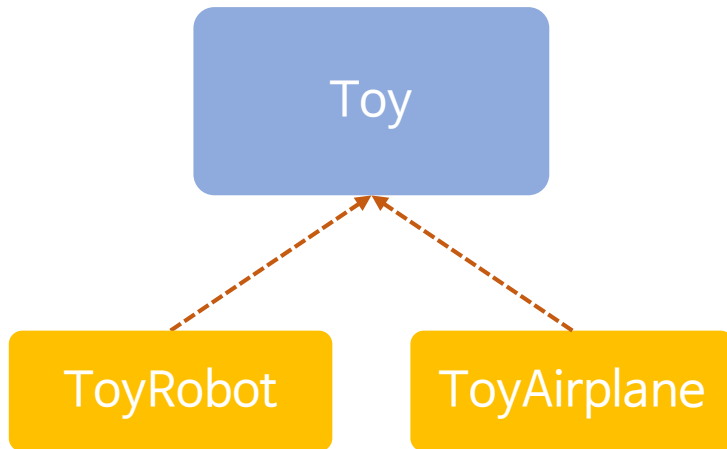
21-3 : 인터페이스 구현

class 대신 interface 키워드를 사용하며, extend 대신 implements 키워드를 이용한다.



21-4 : 장난감 인터페이스

interface를 이용하면 객체가 다양한 자료형(타입)을 가질 수 있다.



```

Toy robot = new ToyRobot();
Toy airplane = new ToyAirplane();

Toy toys[] = {robot, airplane};

for (int i = 0; i < toys.length; i++) {
    toys[i].walk();
    toys[i].run();
    toys[i].alarm();
    toys[i].light();

    System.out.println();
}

```



```

<terminated> MainClass (19) [Java Application] C:\Program
The robot can walk.
The robot can run.
The robot has no alarm function.
The robot has light function.

The airplane can not walk.
The airplane can not run.
The airplane has alarm function.
The airplane has no light function.

```