

Item 21

인터페이스는 구현하는 쪽을 생각해 설계하라

목차

1

인터페이스 메서드 추가

2

Default 메서드란?

3

Default 메서드 문제점

4

Default 메서드 해결책

5

Default 메서드 주의점

목차

1

인터페이스 메서드 추가

2

Default 메서드란?

3

Default 메서드 문제점

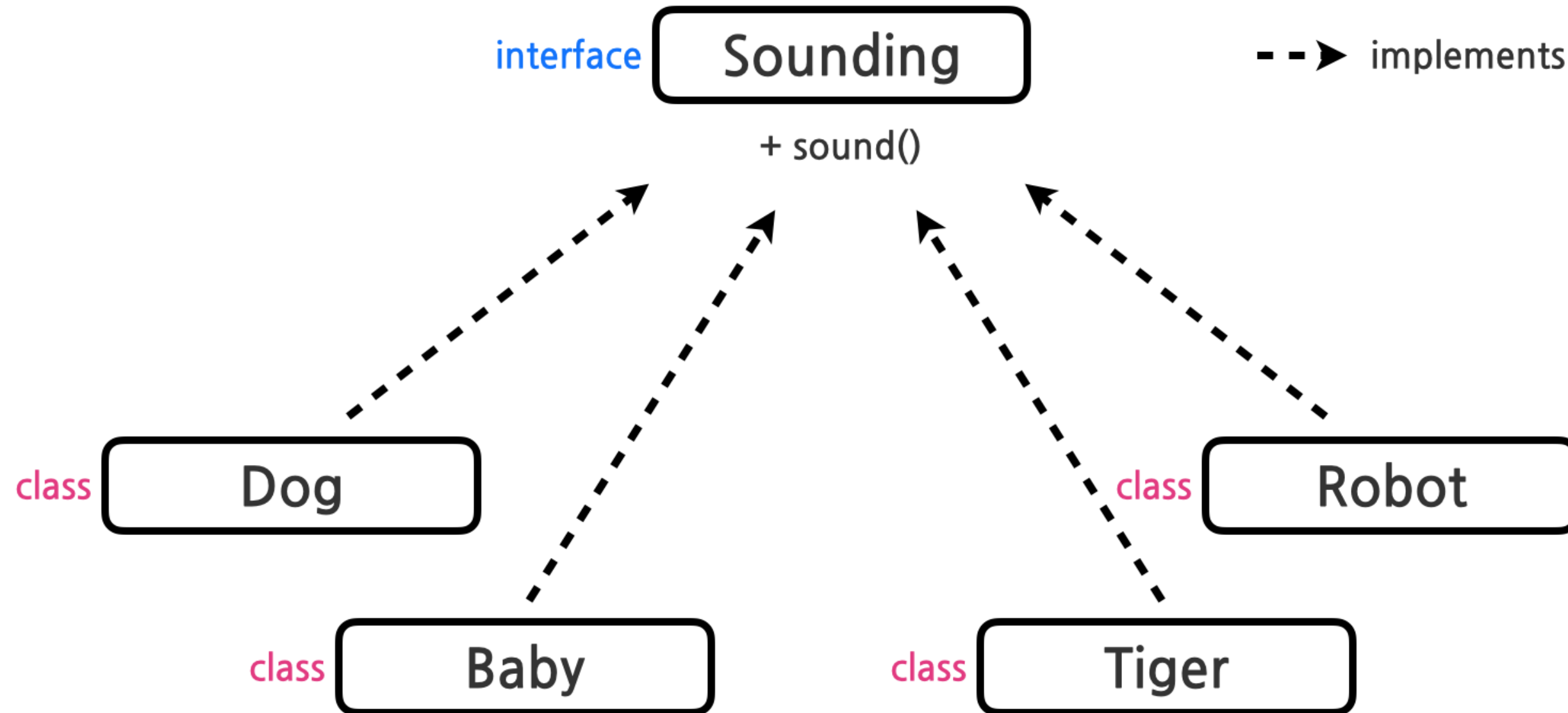
4

Default 메서드 해결책

5

Default 메서드 주의점

인터페이스에 메서드 추가



다형성 구현

**공통 기능
정의 가능**

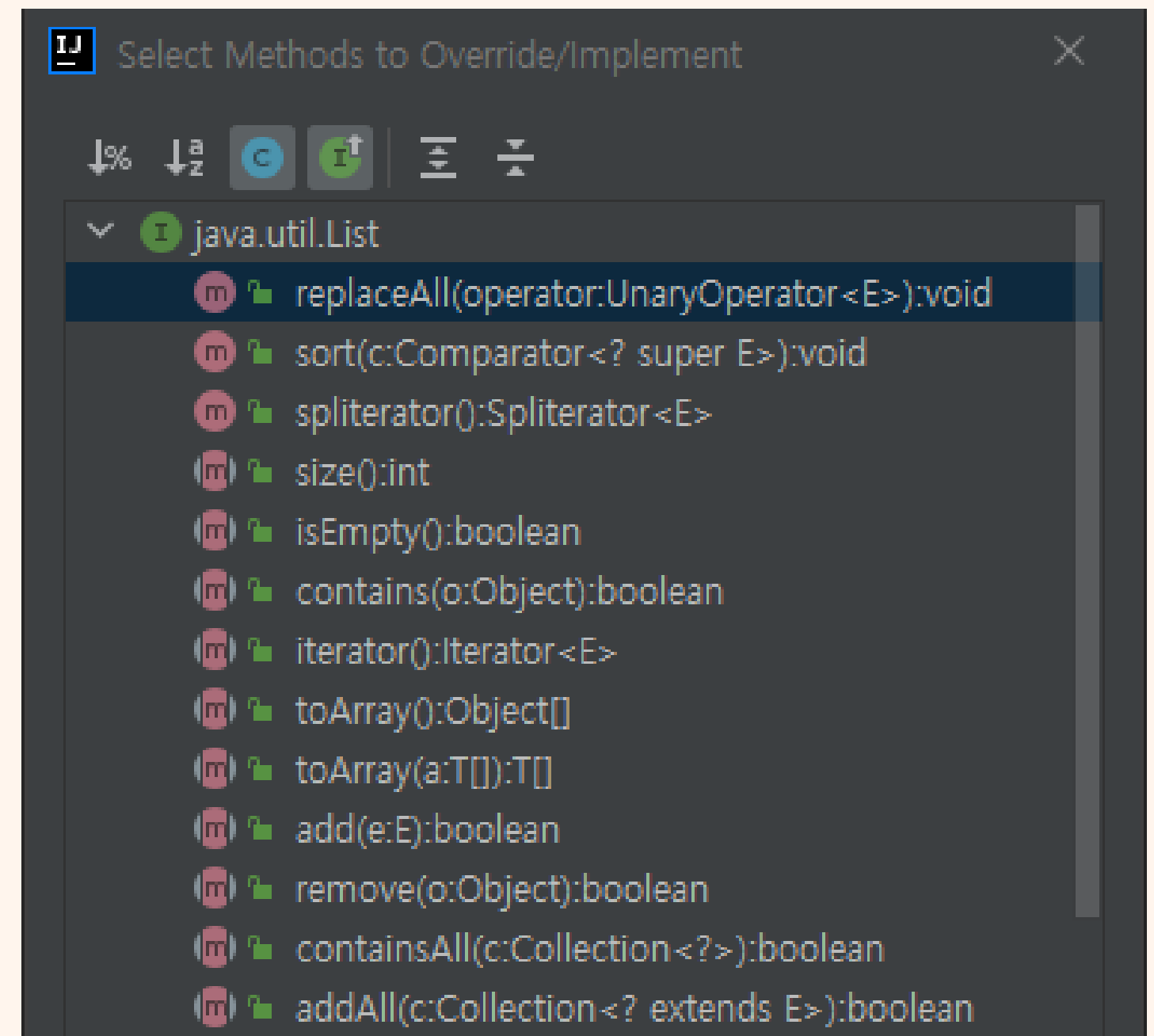
인터페이스에 메서드 추가

```
import java.util.List;

public class TestList implements List {

}
```

인터페이스 메서드 **오버라이딩** 필수



인터페이스에 메서드 추가

인터페이스에 새로운 메서드 추가



구현 클래스에서 메서드 오버라이딩



구현 클래스 매우 많으면? →



default 메서드



목차

1

인터페이스 메서드 추가

2

Default 메서드란?

3

Default 메서드 문제점

4

Default 메서드 해결책

5

Default 메서드 주의점

default 메서드란?

```
public interface MyInterface {  
    void print();  
  
    default void printHello() {  
        System.out.println("Hello World");  
    }  
}
```

인터페이스 내부에도 **로직이 포함된 메서드** 선언하는 것

default 메서드란?

default 메서드 오버라이딩 하지 않아도 오류 X

오버라이딩 하지 않으면 default 메서드 로직 작동



목차

1

인터페이스 메서드 추가

2

Default 메서드란?

3

Default 메서드 문제점

4

Default 메서드 해결책

5

Default 메서드 주의점

default 메서드 문제점

```
default boolean removeIf(Predicate<? super E> filter) {  
    Objects.requireNonNull(filter);  
    boolean removed = false;  
    final Iterator<E> each = iterator();  
    while (each.hasNext()) {  
        if (filter.test(each.next())) {  
            each.remove();  
            removed = true;  
        }  
    }  
    return removed;  
}
```

default 메서드와 **안 맞는 클래스 존재**

default 메서드 문제점

**org.apache.commons.collections4.collection.
SynchronizedCollection**

다중 스레드 환경에서 안전한 컬렉션 구현에 사용

default 메서드 문제점

모든 메서드에서 동기화 후 진행



removeif 메서드 동기화 하도록 재정의 X



인터페이스의 default 구현 동작



removeif 메서드는 동기화 X →



목차

1

인터페이스 메서드 추가

2

Default 메서드란?

3

Default 메서드 문제점

4

Default 메서드 해결책

5

Default 메서드 주의점

default 메서드 해결책

1. 구현 클래스 **default 메서드 재정의** 하도록 함
2. 다른 메서드에서 **default 메서드 호출 전 작업 수행**하도록 함

default 메서드 해결책

```
static class SynchronizedCollection<E> implements Collection<E>, Serializable {  
    ...  
    @Override  
    public boolean removeIf(Predicate<? super E> filter) {  
        synchronized (mutex) {return c.removeIf(filter);}  
    }  
    ...  
}
```

Collection 인터페이스 구현 클래스
removeIf 재정의 & 메서드 사용 전 동기화

목차

1

인터페이스 메서드 추가

2

Default 메서드란?

3

Default 메서드 문제점

4

Default 메서드 해결책

5

Default 메서드 주의점

default 메서드 주의점

**1. default 메서드 컴파일 성공해도
기존 구현체에 런타임 오류 유발 가능**

**2. default 메서드로 새 메서드 추가하는 일
꼭 필요한 경우 아니라면 피해야 함**

default 메서드 주의점

3. default 메서드는 인터페이스로부터 **메서드 제거**하거나
기존 메서드 시그니처 수정 용도 아님 명심

4. default 메서드 있어도 인터페이스 설계할 때는
여전히 세심한 **주의 기울여야 함**

Item 21 결론

인터페이스 설계 시 주의해야 함

default 메서드 사용 시 더 주의해야 함