Item 22.

인터페이스는 타입을 정의하는 용도로만 사용하라

INDEX

목차

01

인터페이스란?

움직인다()
먹는다()

푸들은 움직인다
푸들은 밥을 8시에 먹는다

```java
public interface Dog {

    void move();
    void eat();
}
```

```java
public class Poodle implements Dog {

    private static final int BREAKFAST_TIME = 8;

    @Override
    public void move() {
        System.out.println("푸들이 움직인다");
    }

    @Override
    public void eat() {
        System.out.println("푸들이 밥을 " + BREAKFAST_TIME + "시에 먹는다");

    }
}
```

```
푸들이 움직인다
푸들이 밥을 8시에 먹는다
```


인터페이스

- 인터페이스는 오직 자신의 인스턴스로 무엇을 할 수 있는지를 클라이언트에 얘기해주는 용도
- 인터페이스의 **타입**은 주로 **객체의 행위(메서드)**

상수는 타입인가요?

상수는 타입이 아닌가?

상수는 이미 값이 설정된 고정된 값
상수 자체는 **타입이 아닌, 특정 타입의 값**

```java
public class ConstantTypeEx1 {
    private static final int A = 100; // 상수, 고정된 특정 타입(int)의 값
    int b = 100; // 변수, 타입(int) 값
}
```

02

상수 인터페이스란?

움직인다()
먹는다()

아침 8시

푸들은 움직인다
푸들은 밥을 8시에 먹는다

상수 인터페이스

1. **상수**로 가득 찬 인터페이스
2. **정규화된 이름** 쓰는 것을 피하고자 만든 인터페이스

정규화된 이름? 상수의 클래스 명과 함께 명시된 형태

```java
import static java.lang.Integer.MAX_VALUE;

public class ConstantEx1 {
    Integer num1 = Integer.MIN_VALUE; // 정규화 이름
    Integer num2 = MAX_VALUE; // static import 사용으로 간소화
}
```

상수 인터페이스

상수의 정규화된 이름 사용을 피하고자 인터페이스를 사용하는 경우가 있다.

```java
public class Poodle implements Dog {

    private static final int BREAKFAST_TIME = 8;
```

```java
public interface DogConstant {
    int BREAKFAST_TIME = 8;
}
```

```java
public class PoodleEx1 implements Dog, DogConstant {

    @Override
    public void move() { System.out.println("푸들이 움직인다"); }

    @Override
    public void eat() {
        System.out.println("푸들이 밥을 " + BREAKFAST_TIME + "시에 먹는다");

    }
}
```

03

상수 인터페이스 안티패턴

움직인다()
먹는다()

아침 8시

푸들은 움직인다
푸들은 밥을 8시에 먹는다

상수 인터페이스 잘못된 사용

- 상수 인터페이스의 구현은 내부 구현인 **상수를 노출** 시키는 행위
  - 클라이언트에게 아무런 의미도 없고, 오히려 **혼란을 야기**

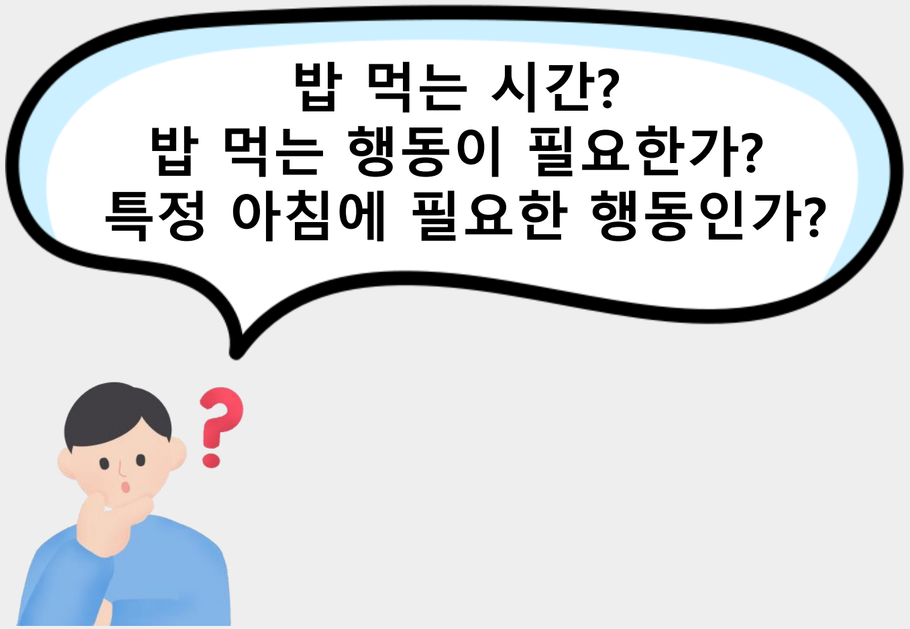
```
public interface ObjectStreamConstants {

    Magic number that is written to the stream header.
    static final short STREAM_MAGIC = (short) 0xaced;

    Version number that is written to the stream header.
    static final short STREAM_VERSION = 5;

    /* Each item in the stream is preceded by a tag
     */

    First tag value.
    static final byte TC_BASE = 0x70;

    Null object reference.
    static final byte TC_NULL = (byte) 0x70;

    Reference to an object already written into the stream.
    static final byte TC_REFERENCE = (byte) 0x71;
```

```
public interface SwingConstants {

    The central position in an area. Used for bo
    (NORTH, etc.) and box-orientation constan
    public static final int CENTER = 0;

    //
    // Box-orientation constant used to
    //

    Box-orientation constant used to specify th
    public static final int TOP = 1;

    Box-orientation constant used to specify th
    public static final int LEFT = 2;

    Box-orientation constant used to specify th
    public static final int BOTTOM = 3;

    Box-orientation constant used to specify th
    public static final int RIGHT = 4;
```

인터페이스 올바른 사용


```java
public interface Dog {

    void move();
    void eat();
}
```

상수 인터페이스 잘못된 사용




```java
public interface DogConstant {
    int BREAKFAST_TIME = 8;
    int LUNCH_TIME = 12;
    int DINNER_TIME = 18;

    String POODLE_REGISTER_NUMBER = "POODLE-1234-5678";
    String BULLDOG_REGISTER_NUMBER = "BULLDOG-8765-4321";
    String BEAGLE_REGISTER_NUMBER = "BEAGLE-5678-1234";
}
```

상수 인터페이스 잘못된 사용

- 상수 인터페이스에 종속되어 의존도가 높은 문제 발생

```java
public interface DogConstant {
    int BREAKFAST_TIME = 8;
    int LUNCH_TIME = 12;
    int DINNER_TIME = 18;
```

```java
public interface DogConstant {
    int BREAKFAST_TIME = 11;
    int LUNCH_TIME = 12;
    int DINNER_TIME = 18;
```

```
푸들이 움직인다
푸들이 밥을 8시에 먹는다
```

```
푸들이 움직인다
푸들이 밥을 11시에 먹는다
```

상수 인터페이스 잘못된 사용

- 상속된 모든 하위 클래스는 모두 상수 인터페이스로 이름 공간 오염 발생

이름 공간

추상적인 개념, 실제로 메모리에 따로 보관 형태 아닌
변수, 함수, 클래스 이름을 관리하는 공간

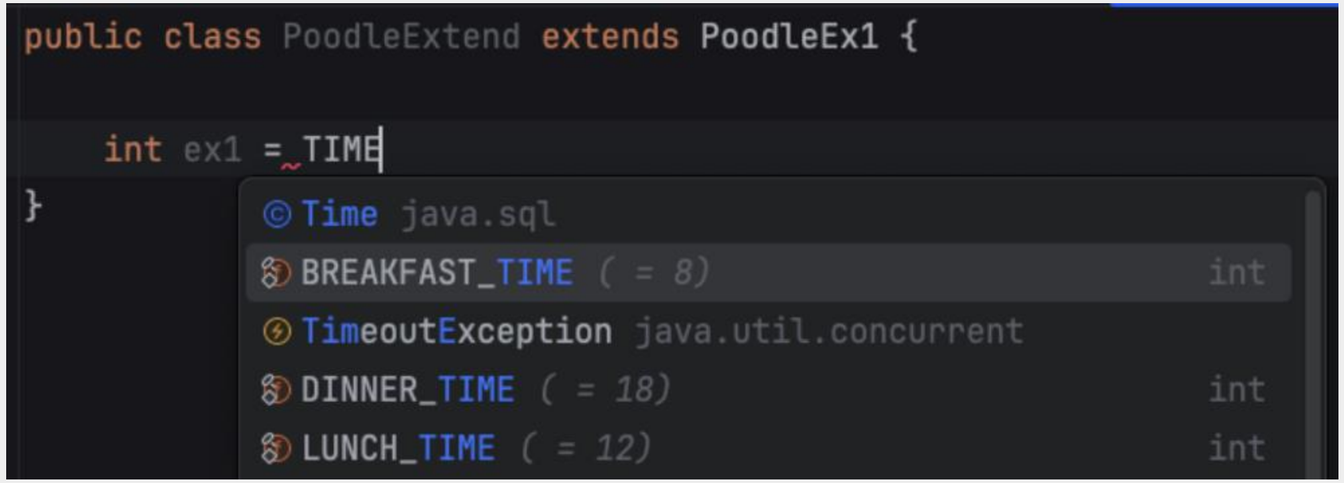
```
Integer.
    parseInt(String s)
    MAX_VALUE ( = 0x7fffffff)
    MIN_VALUE ( = 0x80000000)
```

```
int num1 = 1;
int num2 = 2;
int num3 = 3;

num
num3                int
num1                int
num2                int
```

상수 인터페이스 잘못된 사용

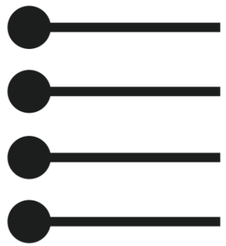
- 상속된 모든 하위 클래스는 모두 상수 인터페이스로 이름 공간 오염 발생

```
public class PoodleEx1 implements Dog, DogConstant {
```



04

상수 인터페이스 대안



열거형(Enum)

유틸형(Util)

```java
public enum ConstantEx2 {
    BREAKFAST_TIME( value: 8),
    LUNCH_TIME( value: 12),
    DINNER_TIME( value: 18)
    ;

    private final int value;

    ConstantEx2(final int value) {
        this.value = value;
    }

    public int getValue() {
        return value;
    }
}
```

presentation_photo

열거형(Enum)

열거 타입으로 정의해 공개

- 사용시 클라이언트는 클래스 이름 명시

```java
int breakfastTime = ConstantEx2.BREAKFAST_TIME.getValue();
```

- 단, 빈번한 사용인 경우 **정적 선언** 고려 (Static Import)

```java
public class ConstantEx3 {

    private ConstantEx3() {
    }

    public static final int BREAKFAST_TIME = 8;
    public static final int LUNCH_TIME = 12;
    public static final int DINNER_TIME = 18;
}
```

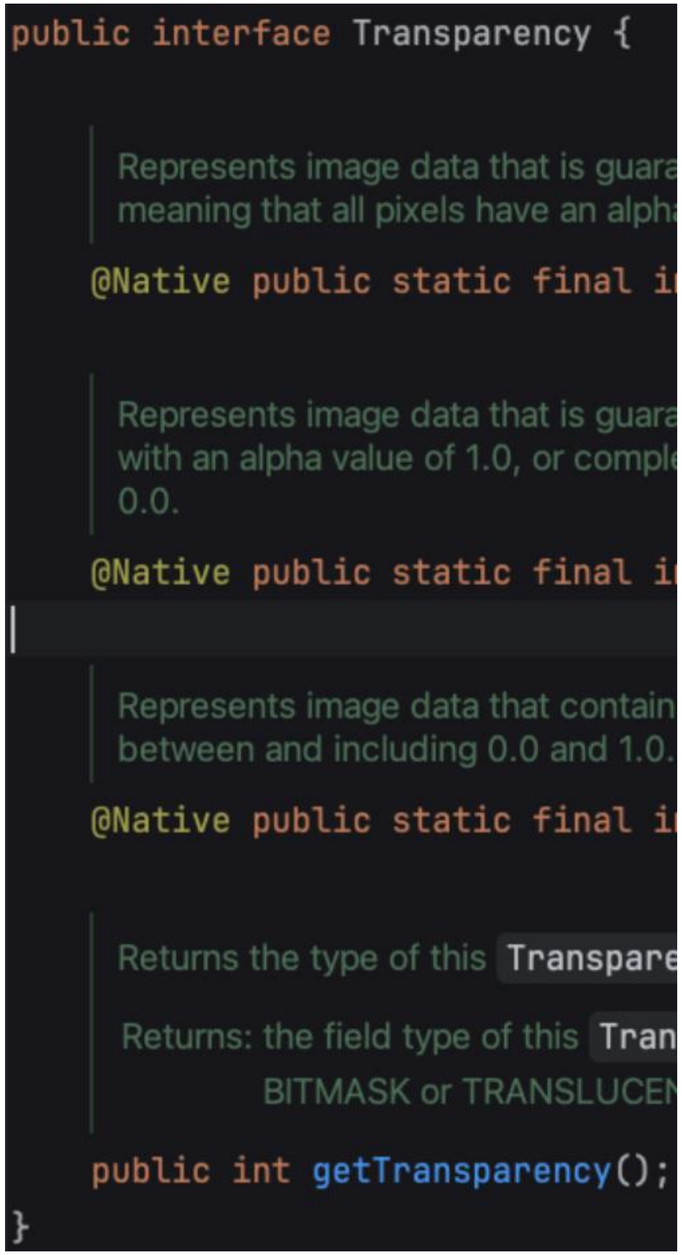

유틸형(Util)

유틸리티로 설계해 공개

- 인스턴스화 할 수 없도록 설계
- 사용시 클라이언트는 클래스 이름 명시

```java
int breakfastTime = ConstantEx3.BREAKFAST_TIME;
```

- 단, 빈번한 사용인 경우 **정적 선언** 고려 (Static Import)



상수 인터페이스 대안

인터페이스 상수 + 행위(메서드) 동시에 선언

- 자바의 오래된 라이브러리
- 일반적으로 상수와 메서드 함께 정의는 삼가
- 안티패턴

Item 22.

인터페이스는 타입을 정의하는 용도로만 사용하라