Java 기초

열거형(Enum)

열거형(Enum)

- Enum이란?
  - 서로 연관된 상수들의 집합을 의미
  - 기존에 상수를 정의하는 방법이였던 final static string 과 같이 문자열이나 숫자들을 나타내는 기본자료형의 값을 enum을 이용해서 같은 효과를 낼 수 있다.

```
public enum Week {
    SUNDAY,
    MONDAY,
    TUESDAY,
    WEDNESDAY,
    THURSDAY,
    FRIDAY,
    SATURDAY
}
```

열거형(Enum)

- Enum의 장점
  - 코드가 단순해지며, 가독성이 좋다.
  - 인스턴스 생성과 상속을 방지하여 상수값의 타입안정성이 보장됩니다.
  - enum class를 사용해 새로운 상수들의 타입을 정의함으로 정의한 타입이외의 타입을 가진 데이터값을 컴파일시 체크한다.
  - 키워드 enum을 사용하기 때문에 구현의 의도가 열거임을 분명하게 알 수 있다.

열거형(Enum)

- Enum 예제

```java
public enum Type {
    WALKING, RUNNING, TRACKING, HIKING
}
```

```java
public class Shoes {
    public String name;
    public int size;
    public Type type;

    public static void main(String[] qrgs) {
        Shoes shoes = new Shoes();

        shoes.name = "나이키";
        shoes.size = 230;
        shoes.type = Type.RUNNING;

        System.out.println("신발 이름 = " + shoes.name);
        System.out.println("신발 사이즈 = " + shoes.size);
        System.out.println("신발 종류 = " + shoes.type);
    }
}
```

열거형(Enum)

- 열거타입은 메소드 영역에 객체가 자리하므로 참조하는 객체들은 같은 객체를 참조하게 된다.

```java
public class Shoes {
    public String name;
    public int size;
    public Type type;

    public static void main(String[] qrgs) {
        Shoes sh1 = new Shoes();
        Shoes sh2 = new Shoes();

        sh1.type = Type.RUNNING;
        sh2.type = Type.RUNNING;

        System.out.println(sh1.type == sh2.type);
    }
```

```
true
```

열거형(Enum)

- 열거 타입에 맞춰 데이터를 넣을 수 있다.
- 단 열거 타입에 값을 넣어주기 위해서는 아래에 해당 값의 필드를 선언하여야 하며 해당 데이터를 받는 생성자를 따로 선언해 주어야 한다.

```
public enum Type {
    WALKING("워킹화"), RUNNING("러닝화"), TRACKING("트래킹화"), HIKING("등산화");

    final private String name;

    private Type(String name) { //enum에서 생성자 같은 역할
        this.name = name;
    }
}
```

열거 객체와 메서드

- 열거 객체에서 사용되는 메서드는 아래와 같다.

리턴타입	메소드(매개변수)	설명
String	name()	열거 객체의 문자열을 리턴
int	ordinal()	열거 객체의 순번(0부터 시작)을 리턴
int	compareTo()	열거 객체를 비교해서 순번 차이를 리턴
열거 타입	valueOf(String name)	주어진 문자열의 열거 객체를 리턴
열거 배열	values()	모든 열거 객체들을 배열로 리턴

열거 객체와 메서드

- 예제

```java
public enum Type {
    WALKING("워킹화"), RUNNING("러닝화"), TRACKING("트래킹화"), HIKING("등산화");

    final private String name;

    private Type(String name) { //enum에서 생성자 같은 역할
        this.name = name;
    }

    public String getName() { // 문자를 받아오는 함수
        return name;
    }
}
```

```java
public class Shoes {
    public static void main(String[] qrgs) {
        for(Type type : Type.values()){
            System.out.println(type.getName());
        }
    }
}
```

```
워킹화
러닝화
트래킹화
등산화
```