1. 열거형(enum)

1. 열거형이란

1. 관련된 상수들을 모아서 클래스처럼 만들어 주는 것.
2. 열거형의 상수들은 모두 값의 변경이 불가능.
3. 클래스와 인터페이스 중간단계의 형태를 가지고 있다.
4. 열거형은 객체처럼 선언하고 사용할 수 있다.
5. 열거형을 사용하지 않을 때 관련있는 상수들을 모두 정의하고 사용할 때도 값의 역할만 해줬는데 열거형에서는 관련있는 상수들을 집합처럼 모아서 선언할 수 있고 값도 여러 개를 할당할 수 있으며 기능인 메소드도 선언을 하고 사용할 수 있다.

2. 열거형의 선언과 사용

1. 열거형의 선언
    - enum 상수들의 공통명칭 {상수1, 상수2, 상수3 ....}
    - 상수들은 대문자로 선언
    - enum 내부에는 상수와 메소드를 선언하고 정의할 수 있다.
2. 열거형의 사용
    - enum의 상수에 접근할 때는 공통명칭.상수명
    - 공통명칭.상수명은 enum의 객체형태로 사용할 수 있다.
    - 공통명칭 변수명 = 공통명칭.상수명; => 해당 상수에 대한 enum 객체가 하나 생성
    - enum의 상수에 값을 지정하지 않으면 순서대로 0 값이 지정된다.

3. 열거형의 메소드

1. values() : 열거형에 선언되어 있는 상수를 배열로 리턴.([열거형.상수, ...]) enum의 객체가 담겨있는 배열을 리턴.
2. valueOf(String 상수명) : 해당 상수명의 열거형 객체를 리턴. 상수의 값은 메소드를 정의하고 해당 메소드의 호출을 통해서 가져온다.
3. ordinal() : 해당 상수의 인덱스 리턴. 몇 번째로 선언된 상수인지 확인할 수 있다.
4. name() : 상수명을 리턴.

4. 열거형의 name, value

1. name
    - name은 enum 안에 정의된 상수의 명칭
    - 관례적으로 대문자로 작성
    - name은 enum 안에서 유일해야 한다.
2. value
    - name에 지정된 상수의 값
    - enum의 선언되어 있는 특정 상수에 접근할 때 생성자에서 상수 값을 담을 변수에 해당 상수 값을 대입할 수 있다.

5. 상수 값의 할당

1. 상수 값은 선언된 모든 상수가 동일한 타입으로 가져야 한다. 첫 번째 상수의 값이 String으로 지정되면 나머지 상수 값들도 String으로 지정돼야 한다.
2. 지정된 상수 값의 타입으로 private 지정된 타입 변수명;으로 변수를 하나 생성한다.
3. 기본 생성자를 하나 만들어서 2번에서 만든 변수에 호출된 상수의 값을 대입해서 사용한다. 변수가 private으로 선언되었으므로 getter/setter를 이용해서 사용한다.
4. 값의 할당은 상수1(동일한 타입의 값), 상수2(동일한 타입의 값), 상수3(동일한 타입의 값).....

6. 상수값 여러개 할당

1. 상수의 값이 여러개 할당되는 첫 번째 상수와 나머지 상수들의 상수 값의 개수를 일치시켜야됨.
2. 상수값이 선언된 위치에 따라 타입을 다르게 할 수 있는데 같은 위치의 상수 값들은 모두 동일한 타입.

```
상수1(int, String, int), 상수2(int, String, int), 상수3(in
```

3. 할당된 값의 개수만큼 변수를 만들어 준다. 생성자에서 초기화.

7. 열거형의 메소드

1. enum은 일반 메소드와 추상메소드를 선언할 수 있다.
2. enum안의 추상메소드는 자식한테 상속돼서 구현되지 않고 enum안에서 선언하고 구현까지 해야된다.
3. enum안의 추상메소드의 선언은 멤버변수와 메소드가 있는 부분에 하되 구현은 상수별로 각각 이뤄져야 된다.

4. 추상메소드 선언과 구현

```
abstract void printRgbValue();
RED("red", 255, 0, 0) {
    void printRgbValue() {
        syso(red);
    }
}
```