

(3) Math 클래스에서 주로 사용하는 메소드들

1) 상수: 전부 long형이므로 결과를 정수형으로 바꾸려면 (int)를 붙여야 한다.

<ex>

```
- 제곱: System.out.println("2의 3승 : "+(int)Math.pow(2, 3));    => 1024
- 절대값: System.out.println("-9.9의 절대값:"+Math.abs(-9.9));    => 9.9
- 제곱근: System.out.println("16의 제곱근 : "+Math.sqrt(16));      => 4.0
- 최대값: System.out.println("4와 7중에 작은값:"+Math.min(4, 7)); => 4.0
```

2) 반올림, 버림, 내림

- 기본적으로 소수 첫째 자리에서 처리한다.
- round는 int형, ceil과 floor는 long형이다.

<ex>

```
올림: System.out.println("9.12의 올림 : "+Math.ceil(9.12));      =>10.0
반올림: System.out.println("9.69의 반올림 : "+Math.round(9.69));  =>10
내림: System.out.println("9.69 버림 : "+Math.floor(9.69));       =>9.0
```

3) Random : 무작위로 해당 범위의 수들 중 하나의 수가 출력되도록 하는 것.

<ex>

```
Random random = new Random();
System.out.println("int 난수: "+random.nextInt());
    => 무작위로 정수들 중 하나의 수가 출력된다.
```

(4) Object Class에서 자주 사용하는 메소드들 (필요에 따라 오버라이딩 가능)

- 1) public String toString(): 객체의 문자열 반환
- 2) public Boolean equals(Object obj)
- 3) public int hashCode(): 객체를 식별하는 정수 값인 해시코드 반환

<ex> - '카드의 모양을 출력하기 위한 toString 오버라이드'

```
@Override
public String toString() {
    return "카드 모양은 "+kind+" "+num;
```

- '카드 정보가 중복되는 경우를 equals로 오버라이드'

```
@Override
public boolean equals(Object obj) {

    if(obj!=null && obj instanceof Card) {
        boolean kindchk = this.kind == ((Card)obj).kind;
        boolean numchk = this.num == ((Card)obj).num;
        return kindchk && numchk;
    }
    return false;
```