

# **프로그래밍 언어 활용 강의안**

## **(Math / Random Class /Format/Arrays)**

## Math, Random 클래스

### Math 클래스

수학 계산에 사용할 수 있는 정적 메소드 제공

메소드	설명	예제 코드	리턴값
int abs(int a) double abs(double a)	절대값	int v1 = Math.abs(-5); double v2 = Math.abs(-3.14);	v1 = 5 v2 = 3.14
double ceil(double a)	올림값	double v3 = Math.ceil(5.3); double v4 = Math.ceil(-5.3);	v3 = 6.0 v4 = -5.0
double floor(double a)	버림값	double v5 = Math.floor(5.3); double v6 = Math.floor(-5.3);	v5 = 5.0 v6 = -6.0
int max(int a, int b) double max(double a, double b)	최대값	int v7 = Math.max(5, 9); double v8 = Math.max(5.3, 2.5);	v7 = 9 v8 = 5.3
int min(int a, int b) double min(double a, double b)	최소값	int v9 = Math.min(5, 9); double v10 = Math.min(5.3, 2.5);	v9 = 5 v10 = 2.5
double random()	랜덤값	double v11 = Math.random();	0.0 ≤ v11 < 1.0
double rint(double a)	가까운 정수의 실수값	double v12 = Math.rint(5.3); double v13 = Math.rint(5.7);	v12 = 5.0 v13 = 6.0
long round(double a)	반올림값	long v14 = Math.round(5.3); long v15 = Math.round(5.7);	v14 = 5 v15 = 6

Math, Random 클래스

## Random 클래스

boolean, int, long, float, double 난수 입수 가능

난수를 만드는 알고리즘에 사용되는 종자값(seed) 설정 가능

종자값이 같으면 같은 난수

Random 클래스로 부터 Random객체 생성하는 방법

생성자	설명
Random()	호출시 마다 다른 종자값(현재시간 이용)이 자동 설정된다.
Random(long seed)	매개값으로 주어진 종자값이 설정된다.

Random 클래스가 제공하는 메소드

리턴값	메소드(매개변수)	설명
boolean	nextBoolean()	boolean 타입의 난수를 리턴
double	nextDouble()	double 타입의 난수를 리턴( $0.0 \leq \sim < 1.0$ )
int	nextInt()	int 타입의 난수를 리턴( $-2^{32} \leq \sim \leq 2^{32}-1$ );
int	nextInt(int n)	int 타입의 난수를 리턴( $0 \leq \sim < n$ )

Format 클래스

## 형식(Format) 클래스

숫자와 날짜를 원하는 형식의 문자열로 변환  
종류

숫자 형식: DecimalFormat

날짜 형식: SimpleDateFormat

매개변수화 된 문자열 형식: MessageFormat

## 숫자 형식 클래스(DecimalFormat)

적용할 패턴 선택해 생성자 매개값으로 지정 후 객체 생성

```
DecimalFormat df = new DecimalFormat("#,###.0");  
String result = df.format(1234567.89);
```

## Format 클래스

### 날짜 형식 클래스(SimpleDateFormat)

```
SimpleDateFormat sdf = new SimpleDateFormat("yyyy 년 MM 월 dd 일");  
String strDate = sdf.format(new Date());
```

패턴 문자	의미	패턴 문자	의미
y	년	H	시(0~23)
M	월	h	시(1~12)
d	일	K	시(0~11)
D	월 구분이 없는 일(1~365)	k	시(1~24)
E	요일	m	분
a	오전/오후	s	초
w	년의 몇번째 주	S	밀리세컨드(1/1000 초)
W	월의 몇번째 주		

### 매개변수화 된 문자열 형식 클래스(MessageFormat)

```
String message = "회원 ID: {0} \n 회원 이름: {1} \n 회원 전화: {2}";  
String result = MessageFormat.format(message, id, name, tel);
```

```
String text = "회원 ID: {0} \n 회원 이름: {1} \n 회원 전화: {2}";  
Object[] arguments = { id, name, tel };  
String result = MessageFormat.format(text, arguments);
```

## Arrays 클래스

### Arrays

배열 조작 기능을 가지고 있는 클래스 - 배열 복사, 항목 정렬, 항목 검색 제공하는 정적 메소드

리턴타입	메소드 이름	설명
int	binarySearch(배열, 찾는값)	전체 배열 항목에서 찾는값이 있는 인덱스 리턴
타겟배열	copyOf(원본배열, 복사할길이)	원본배열의 0 번 인덱스에서 복사할 길이만큼 복사한 배열 리턴, 복사할 길이는 원본배열의 길이보다 크도 되며, 타겟배열의 길이가 된다.
타겟배열	copyOfRange(원본배열, 시작인덱스, 끝인덱스)	원본배열의 시작인덱스에서 끝인덱스까지 복사한 배열 리턴
boolean	deepEquals(배열, 배열)	두 배열의 깊은 비교(중첩 배열의 항목까지 비교)
boolean	equals(배열, 배열)	얕은 비교(중첩 배열의 항목은 비교하지 않음)
void	fill(배열, 값)	전체 배열 항목에 동일한 값을 저장
void	fill(배열, 시작인덱스, 끝인덱스, 값)	시작인덱스부터 끝인덱스까지의 항목에만 동일한 값을 저장
void	sort(배열)	배열의 전체 항목을 올림차순으로 정렬

## Arrays 클래스

### 배열 복사

Arrays.copyOf(원본배열, 복사할 길이)  
0 ~ (복사할 길이-1)까지 항목 복사  
복사할 길이는 원본 배열의 길이보다 커도 됨.

```
char[] arr1 = {'J', 'A', 'V', 'A'};  
char[] arr2 = Arrays.copyOf(arr1, arr1.length);
```

copyOfRange(원본 배열, 시작 인덱스, 끝 인덱스)  
시작인덱스 ~ (끝 인덱스-1)까지 항목 복사

```
char[] arr1 = {'J', 'A', 'V', 'A'};  
char[] arr2 = Arrays.copyOfRange(arr1, 1, 3);
```

### System.arraycopy()

```
System.arraycopy(Object src, int srcPos, Object dest, int destPos, int length)  
원본배열    원본시작인덱스    타겟배열    타겟시작인덱스    복사개수
```

## Arrays 클래스

### 배열 항목 비교

`Arrays.equals(배열, 배열)` - 1차 항목의 값만 비교

`Arrays.deepEquals(배열, 배열)` - 중첩된 배열의 항목까지 비교

### 배열 항목 정렬

`Arrays.sort(배열)`- 항목 오름차 순으로 정렬

기본 타입이거나 String 배열 자동 정렬

사용자 정의 클래스 배열은 Comparable 인터페이스를 구현해야만 정렬

### 배열 항목 검색

특정 값 위치한 인덱스 얻는 것

`Arrays.sort(배열)`로 먼저 정렬

`Arrays.binarySearch(배열, 찾는 값)` 메소드로 항목을 찾아야 정확한 값이  
나옴