

# 기본 패키지

예제 소스 코드는 파일과 연결되어 있습니다.  
editplus(유료), notepad++(무료)와 같은 편집 도구를  
미리 설치하여 PPT를 슬라이드 쇼로 진행할 때 소스  
파일과 연결하여 보면 강의하실 때 편리합니다.

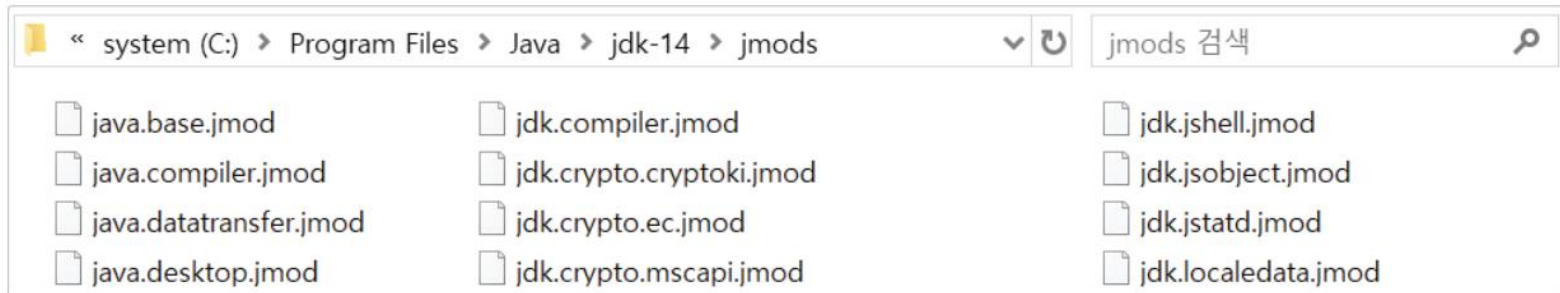
# 패키지와 API 문서

## ■ 자바 라이브러리

- 개발자가 편리하게 사용할 수 있도록 패키지 혹은 모듈을 압축한 파일

## ■ 패키지과 모듈

- 패키지 : 상호 관련 있는 클래스와 인터페이스를 한곳에 묶어 놓은 것
- 모듈 : 밀접한 관계가 있는 패키지와 리소스를 묶어 놓은 것. JDK를 설치하면 jmods 폴더에 jmod 파일을 제공하는데 jmod 파일이 모듈 파일



- JDK 8까지는 개발자가 편리하게 프로그래밍할 수 있도록 기본 패키지를 rt.jar 파일로 제공
- JDK 9부터는 jmod 파일을 통하여 필요한 패키지를 제공

# 패키지와 API 문서

## ■ 자바의 주요 패키지 및 모듈

패키지	설명
java.awt	그래픽을 처리하는 API
java.io	입출력을 스트림하는 API
java.lang	자바 프로그램의 필수 API
java.math	수학과 관련된 API
java.net	네트워크를 처리하는 API
java.text	날짜, 시간, 문자열 등 지역화를 처리하는 API
java.time	JDK 8이 지원하는 날짜 및 시간을 처리하는 API
java.util	날짜, 리스트, 벡터 등 유틸리티 API
javax.swing	스윙 컴포넌트 API

### External Libraries

< 14 > C:\Program Files\Java\jdk-14.0.1

- ▶ java.base library root
- ▶ java.compiler library root
- ▶ java.datatransfer library root
- ▶ java.desktop library root

인텔리J 아이디어에  
보이는 다수의 모듈 파일

# 패키지와 API 문서

## ■ API 문서

- <https://docs.oracle.com/en/java/javase/14/docs/api>

The screenshot shows the Oracle Java SE 14 & JDK 14 API Specification page. At the top, there is a navigation bar with tabs: OVERVIEW, MODULE, PACKAGE, CLASS, USE, TREE, DEPRECATED, INDEX, and HELP. The 'MODULE' tab is selected. To the right of the tabs is a search bar labeled 'SEARCH: Search' with a magnifying glass icon. Below the navigation bar, the title 'Java® Platform, Standard Edition & Java Development Kit Version 14 API Specification' is displayed. The main content area starts with the text 'This document is divided into two sections:'. Below this, there are two sections: 'Java SE' and 'JDK'. The 'Java SE' section states that its APIs define the core Java platform for general-purpose computing. The 'JDK' section states that its APIs are specific to the JDK and may not be available in all implementations. Below the text, there is a table with four tabs: 'All Modules', 'Java SE', 'JDK', and 'Other Modules'. The 'All Modules' tab is selected. The table has two columns: 'Module' and 'Description'. The first row shows 'java.base' with a description 'Defines the foundational APIs of the Java SE Platform.' The second row shows 'java.compiler' with a description 'Defines the Language Model, Annotation Processing, and Java Compiler APIs.'

Annotations on the screenshot:

- 1: Search bar
- 2: java.base module
- 3: OVERVIEW tab
- 4: MODULE tab
- 5: PACKAGE tab

Module	Description
java.base	Defines the foundational APIs of the Java SE Platform.
java.compiler	Defines the Language Model, Annotation Processing, and Java Compiler APIs.

# java.lang 패키지

## ■ 필수 패키지

- 자바 프로그램에서 가장 기본이 되는 클래스와 인터페이스를 보관
- import문 없이 사용
- Java.lang 패키지에 포함된 주요 클래스

클래스	설명
Class	실행 중에 클래스 정보를 제공한다.
Math	각종 수학 함수를 제공한다.
Object	최상위 클래스로 기본적인 메서드를 제공한다.
String, StringBuffer, StringBuilder	문자열을 처리하는 메서드를 제공한다.
System	시스템 정보나 입출력을 처리하는 메서드를 제공한다.
Thread	스레드를 처리하는 메서드를 제공한다.
포장 클래스	기초 타입 데이터를 객체로 처리하는 메서드를 제공한다.

# java.lang 패키지

## ■ Object 클래스

- 모든 클래스의 조상
- Object 클래스가 제공하는 주요 메서드

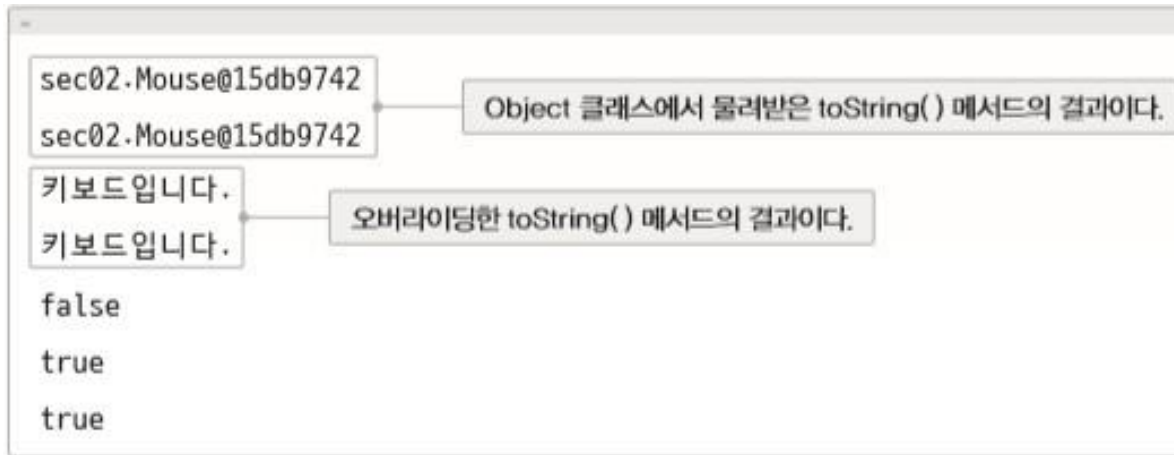
메서드	설명
<code>public String toString()</code>	객체의 문자 정보를 반환한다.
<code>public boolean equals(Object o)</code>	현재 객체와 동일한지 여부를 반환한다.
<code>public int hashCode()</code>	객체의 해시코드를 반환한다.
<code>protected Object clone()</code>	객체의 사본을 생성한다.
<code>protected void finalize()</code>	가비지 컬렉터가 객체를 수거할 때 호출한다.
<code>public final Class&lt;?&gt; getClass()</code>	객체 정보를 반환한다.

- 이외에도 final로 지정된 `wait()`, `notify()`, `notifyAll()` 메서드가 있음

# java.lang 패키지

## ■ Object 클래스

- 대다수 클래스는 Object 클래스가 제공하는 toString( )과 equals( ) 메서드를 오버라이딩해서 사용
- String은 이 두 메서드를 이미 오버라이딩한 클래스임
- 예제 : [sec02/Keyboard](#), [sec02/ObjectMethodDemo](#)



# java.lang 패키지

## ■ Class 클래스

- 실행 중인 자바 프로그램 내부에 포함된 클래스와 인터페이스 정보를 제공하려고 getName( ), getSimpleName( ) 등 다양한 메서드를 제공
- Class 클래스는 생성자가 없고 어떤 객체라도 생성하면 JVM이 대응하는 Class 객체를 자동으로 생성
- 실행 도중 객체 정보를 얻으려면 getClass( )의 결과인 Class 객체를 사용
- 예를 들어 주어진 obj 객체에서 Class 객체를 얻으려면

```
Class cls = obj.getClass();
```

임의의 객체이다.

Object 클래스에서 상속받은 메서드로 Class 객체를 반환한다.

- 예제 : [sec02/ClassDemo](#)

```
sec02.Keyboard
Keyboard
sec02.Keyboard
sec02
```



# java.lang 패키지

## ■ Math 클래스

- 모든 메서드가 static이기 때문에 객체를 생성하지 않고 메서드 호출
- Math 클래스가 제공하는 주요 double 타입의 메서드

- 예제 : [sec02/MathDemo](#)

```
Math.pow(2, 8) : 256.0
Math.random() : 0.27418513672415135
Math.sin(Math.PI) : 1.2246467991473532E-16
Math.min(10, 20) : 10
```

메서드	반환 값
static double abs(double a)	실수 a의 절댓값
static double cos(double a)	실수 a의 cosine 값
static double exp(double a)	$e^a$ 값
static double log(double a)	실수 a에 대한 자연 로그 값
static double log10(double a)	실수 a에 대한 10의 로그 값
static double max(double a, double b)	실수 a와 b 중 큰 값
static double min(double a, double b)	실수 a와 b 중 작은 값
static double pow(double a, double b)	$a^b$ 값
static double random()	0.0과 1.0 사이의 난수
static double sin(double a)	실수 a의 sine 값
static double sqrt(double a)	실수 a의 제곱근 값
static double tan(double a)	실수 a의 tangent 값

# java.lang 패키지

## ■ String 클래스

- 상수 문자열이기 때문에 String 객체에 포함된 문자열을 수정 불가.
- String 객체의 문자열을 수정하는 것은 내부적으로는 수정된 문자열을 포함하는 새로운 String 객체를 생성
- 따라서 문자열 내용을 자주 변경한다면 String 클래스를 사용하는 것은 좋지 않다.

## ■ StringBuffer 및 StringBuilder 클래스

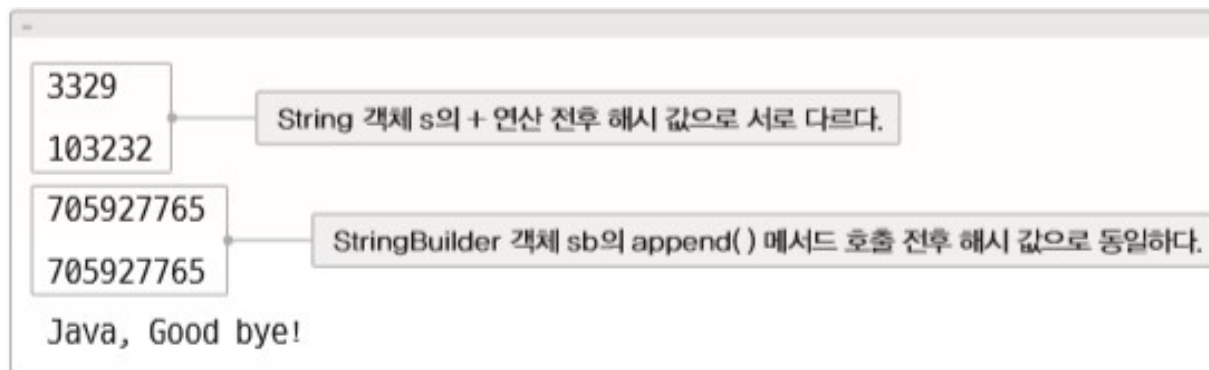
- 자바는 변경될 수 있는 문자열을 다룰 수 있도록 StringBuilder와 StringBuffer 클래스를 제공
- 다중 스레드 환경에서 StringBuffer 클래스가 안전하다는 점을 제외하면 StringBuffer 클래스는 StringBuilder 클래스와 거의 유사
- 다중 스레드 환경이 아니라면 StringBuilder 클래스를 사용하는 것이 효율적

# java.lang 패키지

## ■ StringBuilder 클래스가 제공하는 주요 메서드

메서드	설명
StringBuilder append(String s)	문자열 s를 버퍼에 덧붙인다.
int capacity( )	현재 버퍼의 크기를 반환한다.
StringBuilder delete(int start, int end)	문자열의 일부분을 버퍼에서 제거한다.
StringBuilder insert(int offset, String s)	문자열 s를 버퍼의 offset 위치에 삽입한다.
StringBuilder replace(int start, int end, String s)	문자열의 일부분을 문자열 s로 대체한다.
StringBuilder reverse( )	버퍼에 있는 문자열을 반대 순서로 변경한다.

### ● 예제 : [sec02/StringBuilderDemo](#)



# java.lang 패키지

## ■ System 클래스

- 표준 입출력을 비롯한 실행 시스템과 관련된 필드와 메서드를 static으로 제공
- System.out.println( )도 System 클래스가 제공하는 메서드
- System클래스의 세 가지 필드

필드	설명
static InputStream in	표준 입력 스트림이다.
static PrintStream out	표준 출력 스트림이다.
static PrintStream err	표준 오류 출력 스트림이다.

# java.lang 패키지

## ■ System 클래스

- System클래스가 제공하는 주요 메서드

메서드	설명
<code>static void arraycopy(Object src, int srcPos, Object dest, int destPos, int length)</code>	주어진 위치에서 주어진 길이만큼 배열 src를 배열 dest로 복사한다.
<code>static long currentTimeMillis( )</code>	현재 시각을 밀리초 단위로 반환한다.
<code>static void exit( )</code>	현재 실행 중인 JVM을 종료한다.
<code>static void gc( )</code>	가비지 컬렉터를 실행한다.
<code>static String getenv(String name)</code>	지정된 환경 변수 값을 반환한다.
<code>static String getProperty(String key)</code>	주어진 key 값에 해당하는 시스템 특성을 반환한다.
<code>static long nanoTime( )</code>	현재 시각을 나노초 단위로 반환한다.

- 예제 : [sec02/SystemDemo](#)

```
100 200 300 3 4 5 6
1457507710788
C:\Program Files\Java\jdk1.8.0_131
Windows 10
113787788160112
```

# java.lang 패키지

## ■ System 클래스

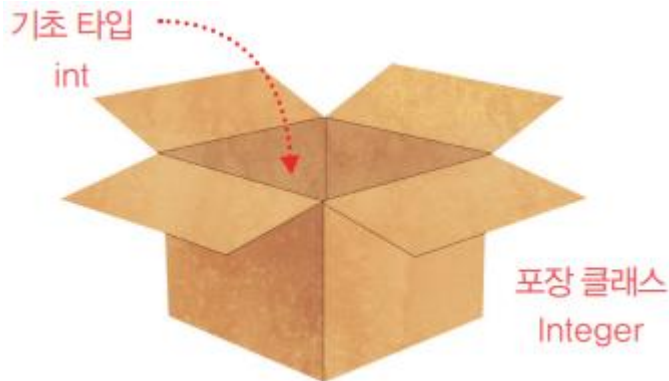
- 자바에서는 운영체제로부터 할당 받은 메모리를 JVM이 관리
  - JVM은 메모리가 부족하거나 주기적으로 가비지 컬렉터를 사용해 가비지를 수거
  - 가비지를 수거하는 순서는 객체의 생성 순서와는 무관
  - 프로그램에서 가비지 컬렉터를 직접 호출 불가. 대신에 `System.gc()`로 JVM에 가능하면 빨리 가비지 컬렉터를 실행하도록 요청 가능
- 
- 예제 : [sec02/GarbageDemo](#)



# java.lang 패키지

## ■ 포장 클래스

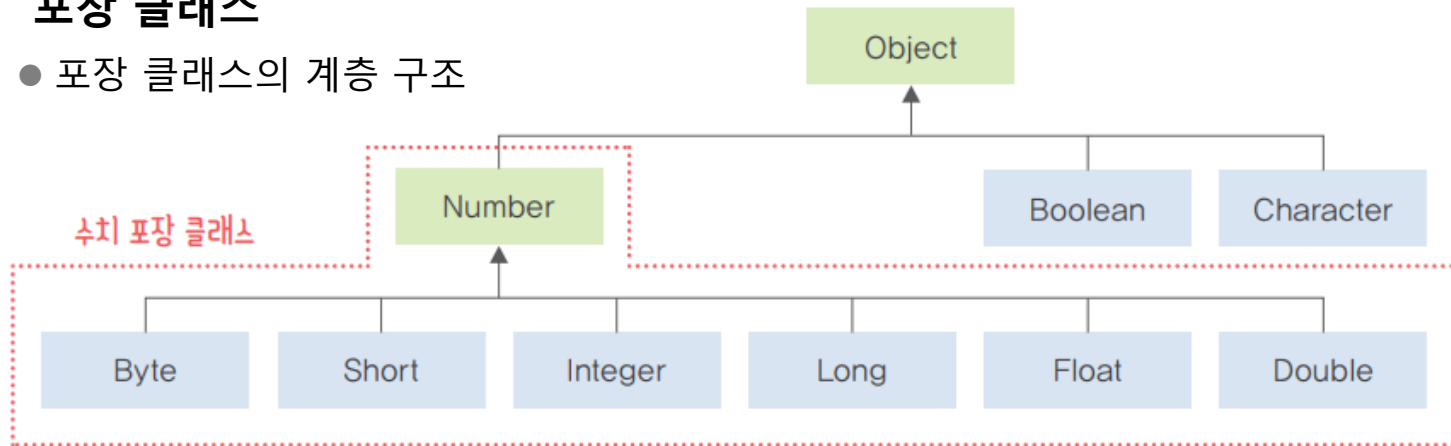
- 대부분의 기본 패키지가 제공하는 클래스의 메서드는 참조 타입을 매개변수로 사용하기 때문에 기초 타입을 사용하면 객체 지향 언어의 특징을 이용 불가
- 자바는 기초 타입을 포장해 클래스화 한 포장 클래스(wrapper class)를 제공해서 기초 타입 데이터도 기본 패키지를 활용토록 함



# java.lang 패키지

## ■ 포장 클래스

- 포장 클래스의 계층 구조



포장 클래스	생성자
Byte	Byte(byte value), Byte(String s)
Short	Short(short value), Short(String s)
Integer	Integer(int value), Integer(String s)
Long	Long(long value), Long(String s)
Float	Float(double value), Float(float value), Float(String s)
Double	Double(double value), Double(String s)
Character	Character(char value)
Boolean	Boolean(boolean value), Boolean(String s)



# java.lang 패키지

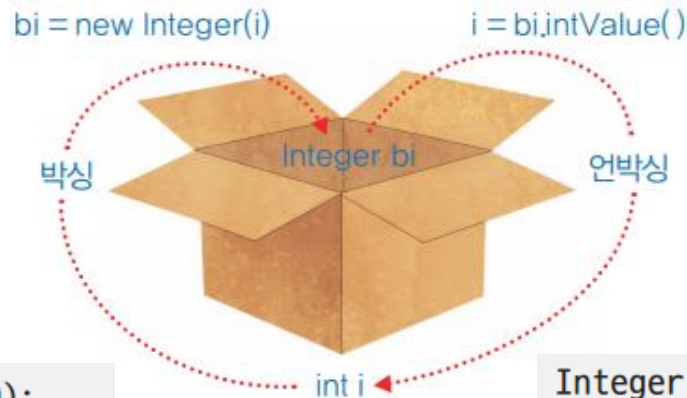
## ■ 포장 클래스

- Integer 클래스가 제공하는 주요 메서드

메서드	설명
<code>int intValue()</code>	int 타입으로 반환한다.
<code>double doubleValue()</code>	double 타입으로 반환한다.
<code>float floatValue()</code>	float 타입으로 반환한다.
<code>static int parseInt(String s)</code>	문자열을 int 타입으로 반환한다.
<code>static String toBinaryString(int i)</code>	int 타입을 2진수 문자열로 반환한다.
<code>static String toHexString(int i)</code>	int 타입을 16진수 문자열로 반환한다.
<code>String toString(int i)</code>	int 타입을 10진수 문자열로 반환한다.
<code>static Integer valueOf(String s)</code>	문자열을 Integer 객체로 반환한다.
<code>static Integer valueOf(String s, int radix)</code>	radix 진수의 문자열을 Integer 객체로 반환한다.

# java.lang 패키지

- 포장 클래스
  - 박싱과 언박싱

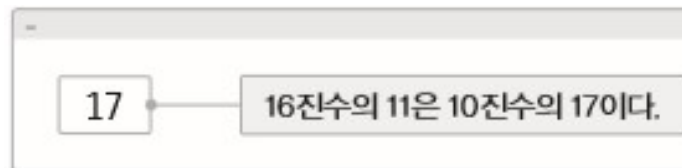


```
Integer bi = new Integer(10);  
Integer bi = Integer.valueOf(10);
```

```
Integer bi = 10;           // 자동 박싱  
int i1 = bi;               // 자동 언박싱  
int i2 = bi + 20;          // 자동 언박싱
```

```
Integer bi = new Integer(10);  
int i = bi.intValue();  
double d = bi.doubleValue();
```

- 예제 : [sec02/WrapperDemo](#)



# java.util 패키지

- 날짜, 시간, 리스트, 벡터, 해시 테이블, 컬렉션 등 다양한 유틸리티 클래스와 인터페이스를 제공
- java.util 패키지가 제공하는 주요 클래스

클래스	설명
Arrays	배열을 비교, 복사, 정렬 등 조작할 때 사용한다.
Calendar	날짜와 시간 정보가 필요할 때 사용한다.
Date	밀리초 단위의 현재 시각이 필요할 때 사용한다.
StringTokenizer	특정 문자로 구분된 문자열을 뽑아낼 때 사용한다.
Random	난수가 필요할 때 사용한다.

# java.util 패키지

## ■ Arrays 클래스

- Arrays 클래스가 제공하는 주요 정적 메서드

메서드	설명
List asList(배열)	배열을 리스트로 변환한다.
int binarySearch(배열, 키)	배열에서 키 값이 있는 인덱스를 반환한다.
배열 copyOf(배열, 길이)	원본 배열을 길이만큼 복사한다.
배열 copyOfRange(배열, 시작, 끝)	원본 배열을 지정한 영역만큼 복사한다.
boolean equals(배열, 배열)	두 배열의 동일 여부를 비교한다.
void fill(배열, 값)	배열을 지정된 값으로 저장한다.
void fill(배열, 시작, 끝, 값)	배열의 지정된 영역에 지정된 값을 저장한다.
void sort(배열)	배열을 오름차순으로 정렬한다.

- 예제 : [sec03/ArraysDemo](#)

```
Java
케이크 애플 도넛 바나나
도넛 바나나 애플 케이크
2
도넛 바나나 기타 기타
```

# java.util 패키지

## ■ Date 클래스

- Date 클래스는 현재 날짜와 시각 정보를 제공.
- 국제화에 맞지 않아 대부분의 메서드는 현재 폐기 중
- 주로 하위 호환성이나 간단한 날짜 정보를 원할 때만 사용

## ■ Calendar 클래스

- 지역이나 문화에 따라 달력을 표시하는 방식이 다르기 때문에 추상 클래스로 되어 있음
- 표준 달력을 사용한다면 다음과 같이 객체 생성

```
Calendar now = Calendar.getInstance();
```

# java.util 패키지

## ■ Calendar 클래스

- Calendar 클래스가 제공하는 정수 타입의 상수

필드 이름	의미
AM, AM_PM, PM	오전 및 오후
DATE	날짜
JANUARY, FEBRUARY, ...	1월, 2월 등
SUNDAY, MONDAY, ...	일요일, 월요일 등
MINUTE	분
HOUR	시간(0~11)
HOUR_OF_DAY	시간(0~23)
MONTH	월(0~11)
DAY_OF_MONTH	한 달 내에서의 날짜
WEEK_OF_YEAR	일 년 내에서의 몇 주차
YEAR	연도

- MONTH는 0~11사이의 정수

# java.util 패키지

## ■ Calendar 클래스

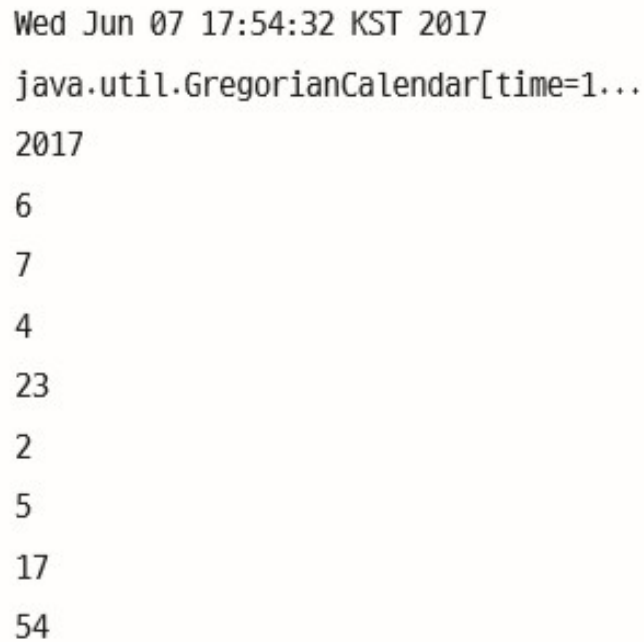
- Calendar 클래스가 제공하는 주요 메서드

메서드	설명
boolean after(Object when)	주어진 시간보다 뒤쪽이면 true를 반환한다.
boolean before(Object when)	주어진 시간보다 앞쪽이면 true를 반환한다.
void clear(int field)	지정된 필드를 미정의 상태로 변경한다.
int compareTo(Calendar anotherCalendar)	2개의 Calendar 객체를 비교한다.
int get(int fields)	주어진 필드 값을 반환한다.
int getFirstDayOfWeek( )	첫 날이 무슨 요일인지 반환한다.
Date getTime( )	Calendar 객체를 Date 객체로 변환한다.
void set(int field, int value)	주어진 필드를 주어진 값으로 변경한다.
void set(int year, int month, int date)	연, 월, 일 값을 변경한다.
void setTime(Date date)	Date 객체로 Calendar 객체를 설정한다.

# java.util 패키지

## ■ Calendar 클래스

- 예제 : [sec03/CalendarDemo](#)



```
Wed Jun 07 17:54:32 KST 2017
java.util.GregorianCalendar[time=1...
2017
6
7
4
23
2
5
17
54
```



# java.util 패키지

## ■ StringTokenizer 클래스

- 문자열을 토큰으로 분리하는 데 사용
- 토큰은 공백이나 줄 바꿈 등 구분자를 사용해 문자열을 분리
- StringTokenizer 클래스의 주요 생성자

생성자	설명
<code>StringTokenizer(String s)</code>	주어진 문자열을 기본 구분자로 파싱한 StringTokenizer 객체를 생성한다.
<code>StringTokenizer(String s, String delim)</code>	주어진 문자열을 delim 구분자로 파싱한 StringTokenizer 객체를 생성한다.

- 기본 구분자는 공백, 탭, 줄 바꿈, 복귀, 용지 먹임 문자

# java.util 패키지

## ■ StringTokenizer 클래스

- StringTokenizer 클래스가 제공하는 주요 메서드

메서드	설명
<code>int countTokens()</code>	남아 있는 토큰의 개수를 반환한다.
<code>boolean hasMoreTokens()</code>	남아 있는 토큰이 있는지 여부를 반환한다.
<code>String nextToken()</code>	다음 토큰을 꺼내 온다.

- 예제 : [sec03/StringTokenizerDemo](#)

```
9
[of] [the] [people] [by] [the] [people] [for] [the] [people]
```

# java.util 패키지

## ■ Random 클래스

생성자	설명
Random()	Random 객체를 생성한다.
Random(long seed)	주어진 시드를 사용하는 Random 객체를 생성한다.

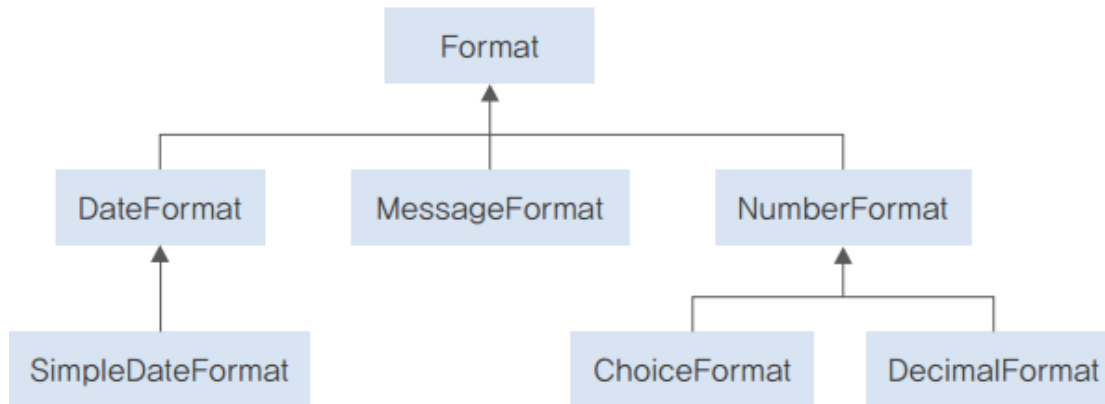
메서드	설명
boolean nextBoolean()	논리 타입 난수를 발생시킨다.
double nextDouble()	0.0~1.0 사이의 double 타입 난수를 발생시킨다.
float nextFloat()	0.0~1.0 사이의 float 타입 난수를 발생시킨다.
double nextGaussian()	평균, 표준편차가 0.0 및 1.0인 정규분포 난수를 발생시킨다.
int nextInt()	int 타입의 난수를 발생시킨다.
int nextInt(int bound)	0~n 사이의 int 타입 난수를 발생시킨다.
long nextLong() n)	long 타입의 난수를 발생시킨다.
void setSeed(long seed)	시드 값을 설정한다.

- 예제 : [sec03/RandomDemo](#)



# java.text 패키지

- 현지화가 필요한 데이터의 효율적 처리를 위한 패키지
- 특히 패키지의 Format 클래스는 지역에 민감한 데이터를 현장에 맞게 문자열로 표현하고 포맷할 수 있도록 지원
- Format 클래스의 계층 구조



# java.text 패키지

## ■ SimpleDateFormat 클래스

- 날짜 정보를 현지화하는 클래스로 날짜를 텍스트로 포맷하거나 텍스트를 날짜로 파싱
- SimpleDateFormat 클래스에서 사용할 수 있는 패턴 기호

패턴 기호	설명	패턴 기호	설명
y	연	h	시(1~12)
M	월	H	시(0~23)
w	월 구분 없는 주	k	시(1~24)
W	주	K	시(0~11)
d	일	m	분
D	월 구분 없는 일	s	초
E	요일	S	밀리초
a	오전과 오후	z	타임존

# java.text 패키지

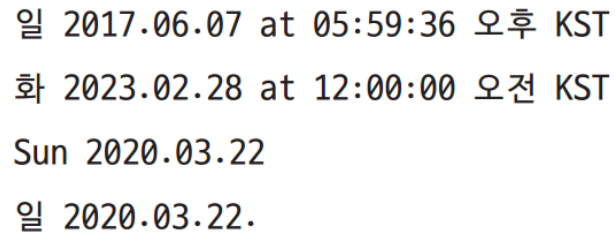
## ■ SimpleDateFormat 클래스

### ● 사용 방법

```
SimpleDateFormat f = new SimpleDateFormat("패턴");  
String s = f.format(new Date());
```

```
SimpleDateFormat f = new SimpleDateFormat("패턴");  
Date d = f.parse("날짜 문자열");
```

### ● 예제 : [sec04/SimpleDateFormatDemo](#)



```
일 2017.06.07 at 05:59:36 오후 KST  
화 2023.02.28 at 12:00:00 오전 KST  
Sun 2020.03.22  
일 2020.03.22.
```

# java.text 패키지

## ■ MessageFormat 클래스

- 문자열을 특정 포맷에 맞추어 깔끔하게 처리하는 클래스
- MessageFormat 객체 생성

```
MessageFormat(String pattern)
```

```
MessageFormat(String pattern, Locale locale)
```

```
static String format(String pattern, Object... arguments)
```

# java.text 패키지

## ■ MessageFormat 클래스

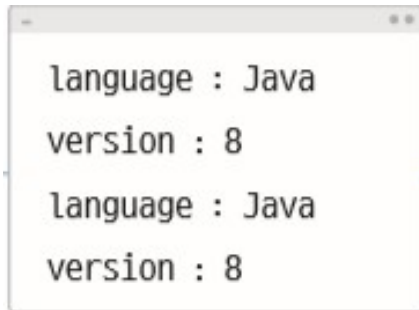
- 사용 방법

```
MessageFormat.format("{0} * {1} = {2}", 3, 4, 3 * 4);
```



```
MessageFormat.format("{0} * {1} = {2}", new Object[]{ 3, 4, 3 * 4 });
```

- 예제 : [sec04/MessageFormatDemo](#)



```
language : Java  
version : 8  
language : Java  
version : 8
```



# java.text 패키지

## ■ DecimalFormat 클래스

- 정수, 실수를 포함한 다양한 종류의 수를 과학적 표기, 퍼센트 표시, 화폐 표시 등으로 포맷 지원
- DecimalFormat 클래스에서 사용할 수 있는 패턴 기호

패턴 기호	설명	사용 예	1234567.890의 반환 값
#	10진수	#	12345678
0	선행 제로 10진수	0000000000.00	01234567.89
.	소수점	#,000	1234567.890
,	구분자	#,###,##	1,234,567.89
+ 또는 -	양수 또는 음수	-#,0	-1234567.9
E	지수	#,##E00	1.23E06
;	패턴 구분	+#,#;-#,#	+1234567.9
%	백분율	#,00%	123456789.00%

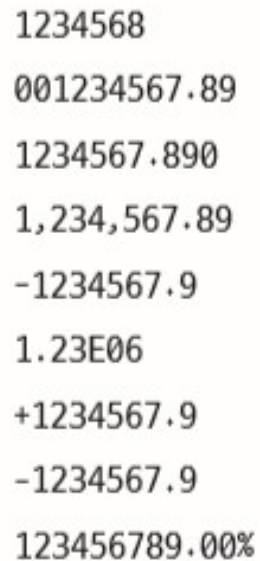
# java.text 패키지

## ■ DecimalFormat 클래스

- 사용 방법

```
DecimalFormat f = new DecimalFormat("패턴");  
String s = f.format(숫자);
```

- 예제 : [sec04/DecimalFormatDemo](#)



```
1234568  
001234567.89  
1234567.890  
1,234,567.89  
-1234567.9  
1.23E06  
+1234567.9  
-1234567.9  
123456789.00%
```