

기본 패키지

패키지

- 상호 관련 있는 클래스와 인터페이스를 한곳에 묶어 놓은 것
- 자바는 개발자가 편리하게 프로그래밍할 수 있도록 기본 패키지를 제공
- 패키지는 중첩되지 않음.
 - java.util 및 java.util.regex는 각각 독립된 패키지로서 java.util 패키지가 java.util.regex 패키지를 포함하지 않음
- JDK에 제공하는 기본 패키지는 java로 시작하며, 확장 패키지는 javax로 시작

패키지

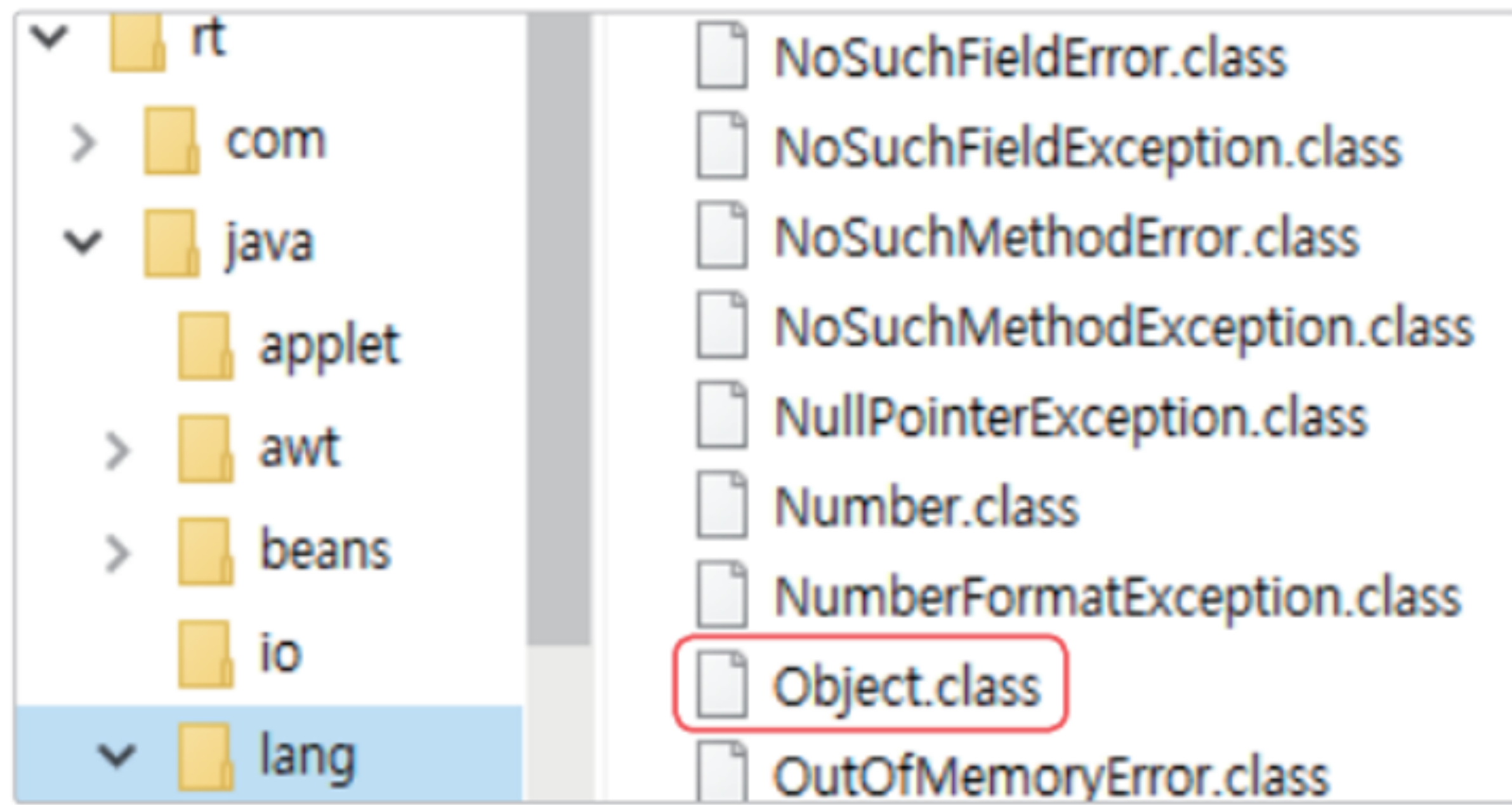
주요기본 패키지

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

●

패키지 확인

- 자바 기본 패키지를 압축 해제한 폴더와 클래스



패키지 확인

- 자바 API 문서(JDK 17)

Java® Platform, Standard Edition & Java Development Kit
Version 17 API Specification

This document is divided into two sections:

Java SE

The Java Platform, Standard Edition (Java SE) APIs define the core Java platform for general-purpose computing. These APIs are in modules whose names start with java.

JDK

The Java Development Kit (JDK) APIs are specific to the JDK and will not necessarily be available in all implementations of the Java SE Platform. These APIs are in modules whose names start with jdk.

All Modules	Java SE	JDK	Other Modules
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.datatransfer	Defines the API for transferring data between and within applications.		
java.desktop	Defines the AWT and Swing user interface toolkits, plus APIs for accessibility, audio, imaging, printing, and JavaBeans.		
java.instrument	Defines services that allow agents to instrument programs running on the JVM.		
java.logging	Defines the Java Logging API.		
java.management	Defines the Java Management Extensions (JMX) API.		
java.management.rmi	Defines the RMI connector for the Java Management Extensions (JMX) Remote API.		
java.naming	Defines the Java Naming and Directory Interface (JNDI) API.		

<https://docs.oracle.com/en/java/javase/17/docs/api/index.html>

패키지

자바 API 문서(JDK8)

The screenshot shows the Java Platform SE 8 API documentation page for the `System` class. The browser address bar shows `http://docs.oracle.com/javase/8/docs/api/`. The page has a dark blue header with navigation tabs: OVERVIEW, PACKAGE, CLASS (selected), USE, TREE, DEPRECATED, INDEX, and HELP. Below the header, there are links for PREVIOUS CLASS, NEXT CLASS, FRAMES, and NO FRAMES. The main content area displays the class `System` as a public final class that extends `Object`. It includes a description: "The System class contains several useful class fields and methods. It cannot be instantiated." and a "Since:" section indicating it was introduced in JDK1.0. A "Field Summary" section is visible at the bottom, showing a table with columns "Modifier and Type" and "Field and Description". The first entry is `static PrintStream err`. On the left side, there is a sidebar with a list of packages and classes. A blue box highlights the "자바 패키지" (Java Packages) section, and another blue box highlights the "선택한 패키지에 포함된 클래스들" (Classes included in the selected package) section, which lists various classes including `System`.

Java™ Platform Standard Ed. 8

System (Java Platform SE 8) ×

파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(T) 도움말(H)

OVERVIEW PACKAGE **CLASS** USE TREE DEPRECATED INDEX HELP

PREV CLASS NEXT CLASS FRAMES NO FRAMES

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

compact1, compact2, compact3
java.lang

Class System

java.lang.Object
java.lang.System

public final class **System** 선택된 클래스에 대한 설명
extends Object

The System class contains several useful class fields and methods. It cannot be instantiated.

Among the facilities provided by the System class are standard input, standard output, and error output streams; access to externally defined properties and environment variables; a means of loading files and libraries; and a utility method for quickly copying a portion of an array.

Since:
JDK1.0

Field Summary

Modifier and Type	Field and Description
static PrintStream	err

<http://docs.oracle.com/javase/8/docs/api>

패키지

자바 API

- java.lang.Object 클래스 확인하기
- <http://docs.oracle.com/javase/8/docs/api>

java.lang 패키지

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

java.lang 패키지

- Object 클래스
- 모든 클래스의 조상
- Object 클래스가 제공하는 주요 메서드

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

java.lang 패키지

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

```
1 package sec02;
2
3 class Mouse {
4     String name;
5
6     public Mouse(String name) {
7         this.name = name;
8     }
9 }
10
11 class Keyboard {
12     String name;
13
14     public Keyboard(String name) {
15         this.name = name;
16     }
17
18     public boolean equals(Object obj) {
19         if (obj instanceof Keyboard) {
20             Keyboard k = (Keyboard) obj;
21             if (name.equals(k.name))
22                 return true;
23         }
24         return false;
25     }
26
27     public String toString() {
28         return "I am keyboard";
29     }
30 }
```

```
1 package sec02;
2
3 public class ObjectMethodDemo {
4     public static void main(String[] args) {
5         Mouse m1 = new Mouse("Logitech");
6         Mouse m2 = new Mouse("Logitech");
7         Mouse m3 = m1;
8         Keyboard k1 = new Keyboard("Microsoft");
9         Keyboard k2 = new Keyboard("Microsoft");
10
11         System.out.println(m1.toString());
12         System.out.println(m1);
13
14         System.out.println(k1.toString());
15         System.out.println(k1);
16
17         System.out.println(m1.equals(m2));
18         System.out.println(m1.equals(m3));
19         System.out.println(k1.equals(k2));
20     }
21 }
```

Console Properties

<terminated> ObjectMethodDemo [Java Applicatio

```
sec02.Mouse@5f8ed237
sec02.Mouse@5f8ed237
I am keyboard
I am keyboard
false
true
true
```


java.lang 패키지

Class 클래스

- 실행 중인 자바 프로그램 내부에 포함된 클래스와 인터페이스 정보를 제공하기 위해 `getName()`, `getSimpleName()` 등 다양한 메서드를 제공
- Class클래스는 생성자가 없고 JVM이 대응하는 Class객체를 자동으로 생성
- 실행 도중 객체 정보를 얻으려면 `getClass()`의 결과인 Class 객체를 사용

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

임의의 객체이다.

Object 클래스에서 상속받은 메서드로 Class 객체를 반환한다.

java.lang 패키지

Class 클래스

- 객체를 사용하여 클래스 정보를 얻음

```
1 package sec02;
2
3 public class ClassDemo {
4     public static void main(String[] args) {
5         Keyboard k = new Keyboard("Logitech");
6
7         Class c = k.getClass();
8         System.out.println(c.getName());
9         System.out.println(c.getSimpleName());
10        System.out.println(c.getTypeName());
11        System.out.println(c.getPackage().getName());
12    }
13 }
```

<terminated> ClassDer
sec02.Keyboard
Keyboard
sec02.Keyboard
sec02

java.lang 패키지

Math 클래스

- 모든 메서드가 static이기 때문에 객체를 생성하지 않고 메서드 호출
- Math 클래스가 제공하는 메서드 예(double 타입 반환)

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

Math 클래스

- Math.PI를 사용하여 원의 넓이와 원의 둘레를 구하도록 Circle.java을 수정함

java.lang 패키지

String 클래스

- String 상수는 문자열이기 때문에 String 객체에 포함된 문자열을 수정 불가함
- String객체의 문자열을 수정하는 것은 내부적으로는 수정된 문자열을 포함하는 새로운 String객체를 생성
- 문자열 내용을 자주 변경한다면 String 클래스를 사용하는 것은 좋지 않음

java.lang 패키지

StringBuffer 및 StringBuilder 클래스

- 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

- 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.

java.lang 패키지

StringBuilder

- ObjectMethodDemo.java
 - String 클래스와 StringBuilder 차이점을 이해한다

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

- System 클래스가 제공하는 주요 메서드

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

java.lang 패키지

System 클래스

- 프로그램의 수행시간을 출력하도록 currentTimeMillis() 메소드를 사용하여 프로그램을 완성하시오

```
long start = System.currentTimeMillis();
```

```
// ...
```

```
long finish = System.currentTimeMillis();
```

```
long timeElapsed = finish - start;
```


java.lang 패키지

System 클래스

- 자바에서는 운영체제로부터 할당받은 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

java.lang 패키지

System 클래스

```
1 package sec02;
2
3 class Garbage {
4     public int no;
5
6     public Garbage(int no) {
7         this.no = no;
8         System.out.printf("Garbage(%d) create\n", no);
9     }
10
11     protected void finalize() {
12         System.out.printf("Garbage(%d) collect\n", no);
13     }
14 }
15
16 public class GarbageDemo {
17     public static void main(String[] args) {
18         for (int i = 0; i < 3; i++)
19             new Garbage(i);
20
21         System.gc();
22     }
23 }
```

<terminated> GarbageDei
Garbage(0) create
Garbage(1) create
Garbage(2) create
Garbage(2) collect
Garbage(1) collect
Garbage(0) collect

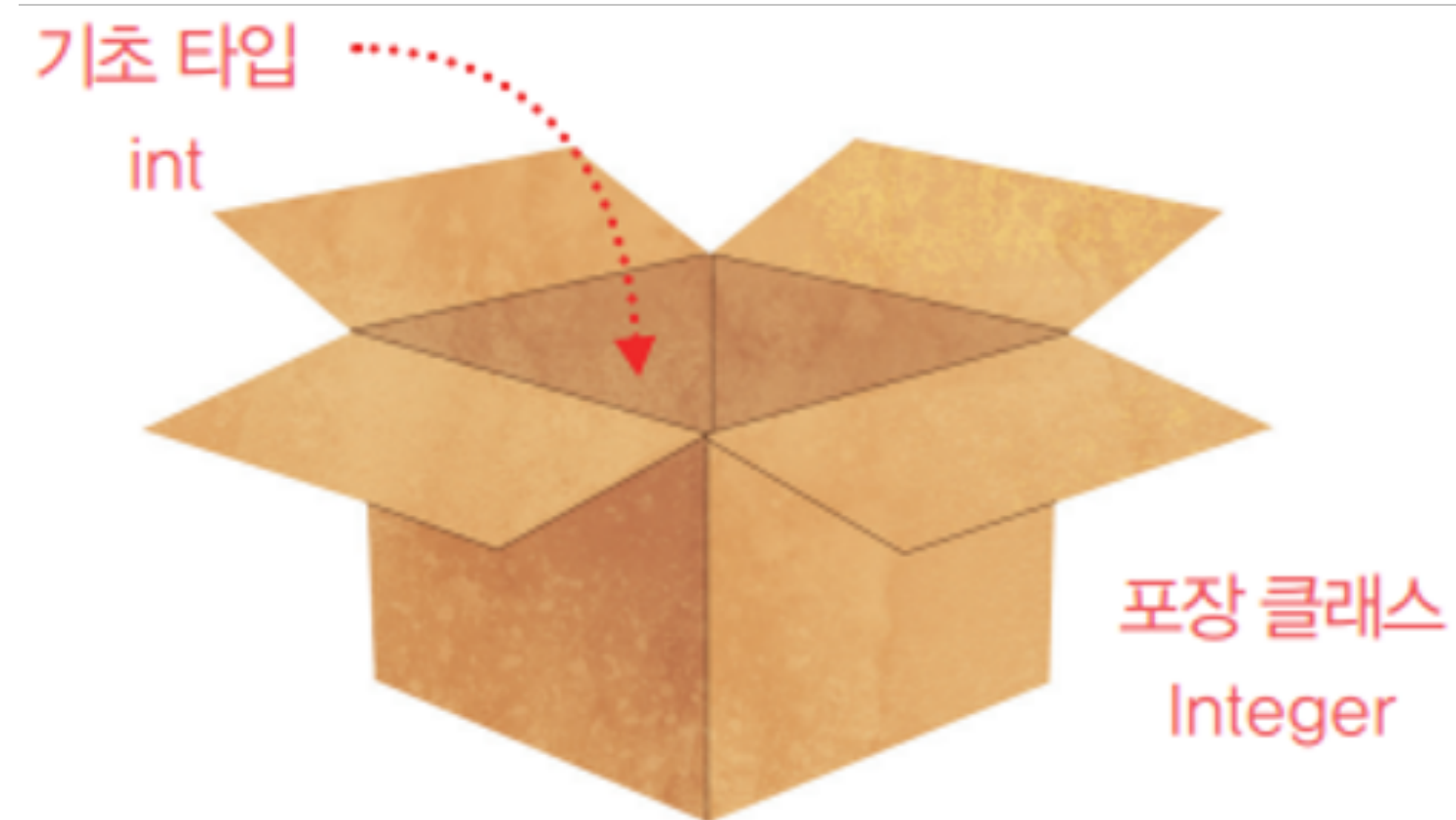
Outline

sec02
Garbage

java.lang 패키지

포장 클래스

- 자바는 기초 타입을 포장해 클래스화한 포장 클래스(wrapper class)를 제공해서 기초 타입 데이터도 기본 패키지를 활용토록 함

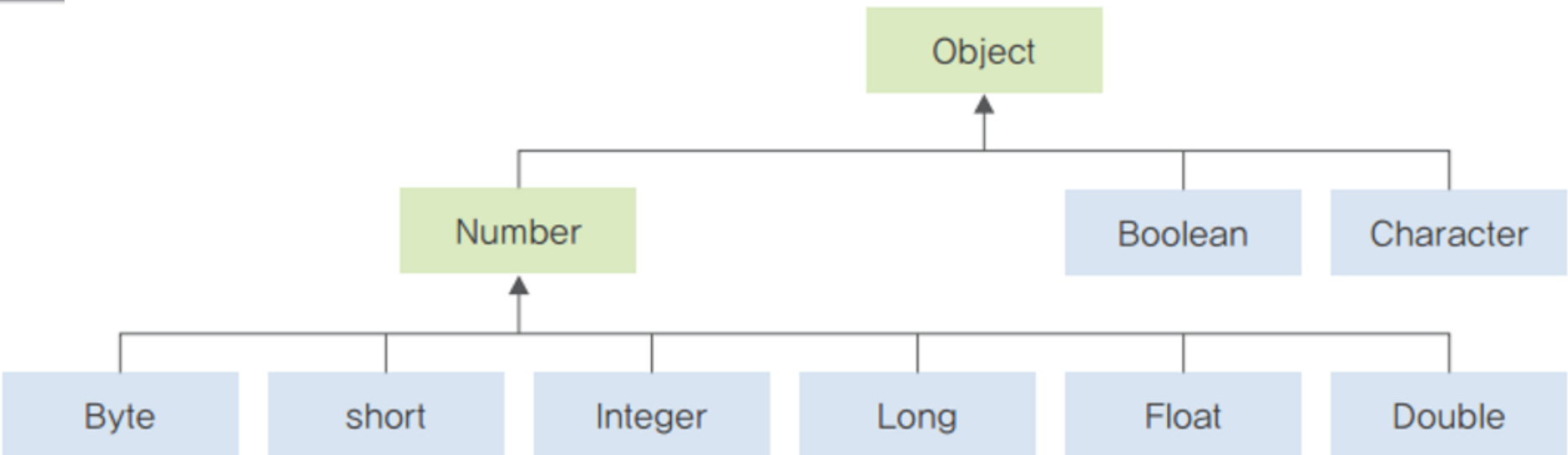


java.lang 패키지

포장 클래스

- 포장 클래스의 계층 구조

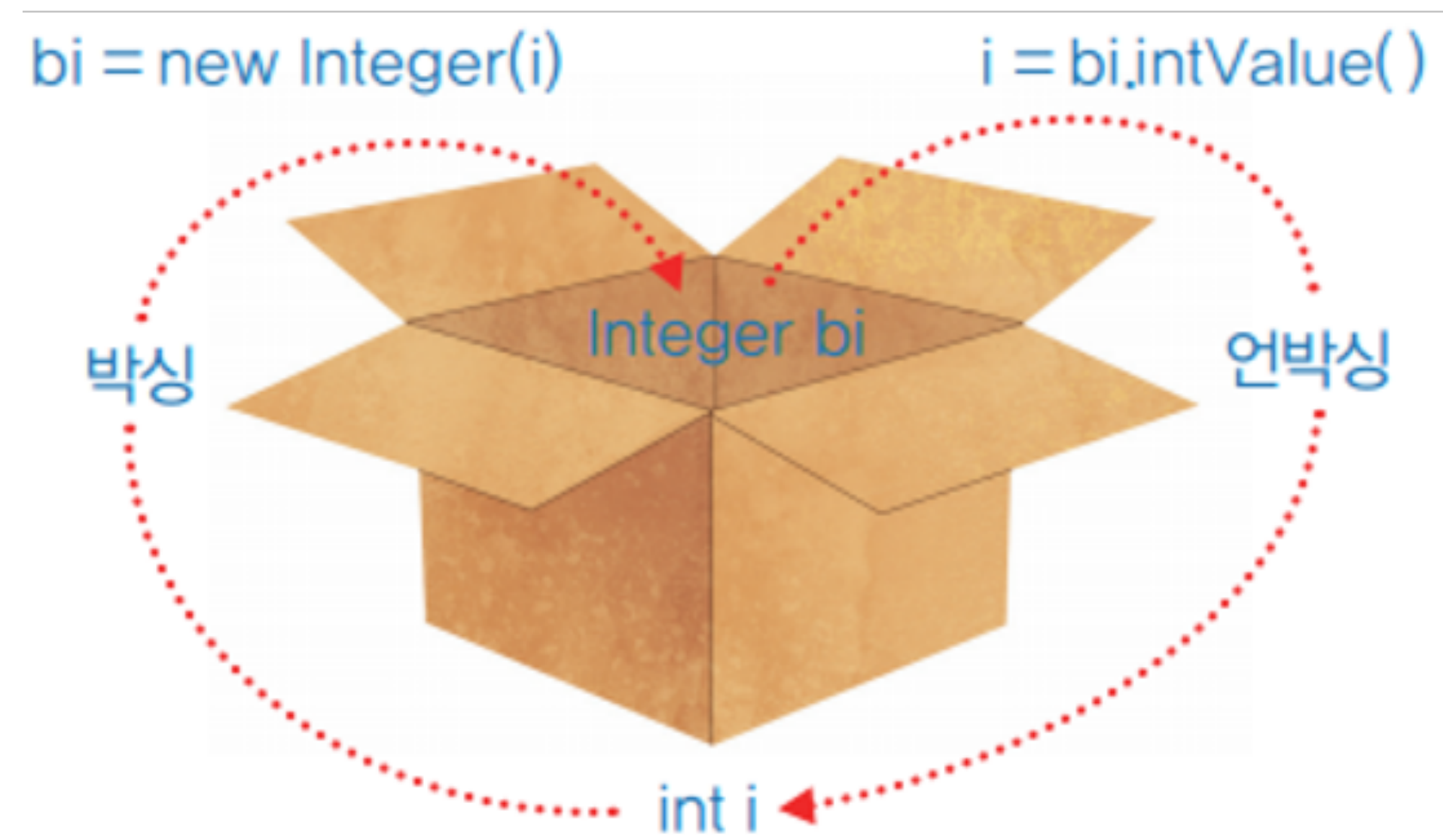
포장 클래스	생성자
Byte	Byte(byte value), Byte(String s)
Short	Short(short value), Short(String s)
Integer	Integer(int value), Integer(String s)
Long	Long(long value), Long(String s)
Float	Float(double value), Float(float value), Float(String s)
Double	Double(double value), Double(String s)
Character	Character(char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



java.lang 패키지

포장 클래스

- 박싱과 언박싱



java.lang 패키지

포장 클래스

- 박싱과 언박싱

```
1 package sec02;
2
3 public class WrapperDemo {
4     public static void main(String[] args) {
5         //Integer bi1 = new Integer(10);
6         Integer bi1 = Integer.valueOf(10);
7
8         int i1 = bi1.intValue();
9
10        double d = bi1.doubleValue();
11
12        Integer bi2 = 20;
13
14        int i2 = bi2 + 20;
15
16        String s1 = Double.toString(3.14);
17
18        Float pi = Float.parseFloat("3.14");
19
20        Integer bi3 = Integer.valueOf("11", 16);
21
22        System.out.println(bi3);
23    }
24 }
```

