

# Java



# 자바 클래스들

JDK 라이브러리에 대하여

문자열 관련 클래스들의 사용 방법

날짜/시간 관련 클래스들의 사용 방법

수학적 계산에 사용되는 클래스들의 사용 방법



- 01. JDK 라이브러리에 대하여
  - JDK 라이브러리란?

- 
- 이런 파일들 안에 JDK 라이브러리의 클래스와 인터페이스들이 들어있습니다.
- String 클래스와 System 클래스의 클래스 파일

# 기초적인 자바 클래스들

## 01. JDK 라이브러리에 대하여 JDK 라이브러리란? (버전별로 업그레이드 됨)

•• JDK 라이브러리에 대한 설명이 있는 API 규격서 (1)

The screenshot displays three web browser windows from the early 2000s, illustrating the process of finding Java API documentation.

- Left Window:** The "Java Technology" website (http://java.sun.com/). It features the Sun Developer Network logo and a navigation menu with links to "APIs" and "Downloads". A "What's New" section highlights "AJAX in Action" and "Using Headless Mode in the Java SE".
- Middle Window:** The "API Specifications" page (http://java.sun.com/reference/api/). It lists various API specifications, including "Standard Edition", "J2SE 1.5.0", "Non-Core APIs", "Enterprise Edition", and "Micro Edition".
- Right Window:** The "Overview (Java 2 Platform SE 5.0)" page (http://java.sun.com/j2se/1.5.0/docs/api/). This is the "Java™ 2 Platform Standard Edition 5.0 API Specification" document. It provides an overview of the API, including a table of contents and a list of packages.

The "Overview" page includes the following text:

Java™ 2 Platform Standard Edition 5.0  
API Specification

This document is the API specification for the Java 2 Platform Standard Edition 5.0.

See: [Description](#)

**Java 2 Platform Packages**

Package	Description
<a href="#">java.applet</a>	Provides the classes necessary to create an applet and the classes an applet uses to communicate with its applet context.
<a href="#">java.awt</a>	Contains all of the classes for creating user interfaces and for painting graphics and images.
<a href="#">java.awt.color</a>	Provides classes for color spaces.
<a href="#">java.awt.datatransfer</a>	Provides interfaces and classes for transferring data between and within applications.
<a href="#">java.awt.dnd</a>	Drag and Drop is a direct manipulation gesture found in many Graphical User Interface systems that provides a mechanism to transfer information between two entities logically associated with presentation elements in the GUI.
<a href="#">java.awt.event</a>	Provides interfaces and classes for dealing with different types of events fired by AWT components.
<a href="#">java.awt.font</a>	Provides classes and interface relating to fonts.
<a href="#">java.awt.geom</a>	Provides the Java 2D classes for defining and performing operations on objects related to two-dimensional geometry.
<a href="#">java.awt.im</a>	Provides classes and interfaces for the input method framework.
<a href="#">java.awt.im.spi</a>	Provides interfaces that enable the development of input methods that can be used with any Java runtime environment.
<a href="#">java.awt.image</a>	Provides classes for creating and modifying images.
<a href="#">java.awt.image.renderable</a>	Provides classes and interfaces for producing rendering-independent images.
<a href="#">java.awt.print</a>	Provides classes and interfaces for a general printing API.

# 기초적인 자바 클래스들

## 01. JDK 라이브러리에 대하여

### JDK 라이브러리란?

- JDK 라이브러리에 대한 설명이 있는 API 규격서 (2)

패키지 이름을 선택하면 그에 속하는 클래스와 인터페이스 이름들이 나열됩니다.

Overview (Java 2 Platform SE 5.0) - Microsoft Internet Explorer

주소(D) http://java.sun.com/j2se/1.5.0/docs/api/

패키지 이름: [java.security.cert](#), [java.security.interfaces](#), [java.security.spec](#), [java.sql](#), [java.text](#), [java.util](#), [java.util.concurrent](#), [java.util.concurrent.atomic](#)

Selected Package: [java.text](#)

Interfaces

- [AttributedCharacterIterator](#)
- [CharacterIterator](#)

Classes

- [Annotation](#)
- [AttributedString](#)
- [Bidi](#)
- [BreakIterator](#)

Overview Package Class Use Tree Deprecated Index Help

PREV NEXT FRAMES NO FRAMES

### Java™ 2 Platform Standard Edition 5.0 API Specification

This document is the API specification for the Java 2 Platform Standard Edition 5.0.

See: [Description](#)

Java 2 Platform Packages	
<a href="#">java.applet</a>	Provides the classes necessary to create an applet and the classes an applet uses to communicate with its applet context.
<a href="#">java.awt</a>	Contains all of the classes for creating user interfaces and for painting

# 기초적인 자바 클래스들

## 01. JDK 라이브러리에 대하여

### JDK 라이브러리란?

- JDK 라이브러리에 대한 설명이 있는 API 규격서 (3)

The screenshot shows the Java API documentation for the `DateFormat` class in the `java.text` package. The browser window title is "DateFormat (Java 2 Platform SE 5.0) - Microsoft Internet Explorer". The address bar shows the URL `http://java.sun.com/j2se/1.5.0/docs/api/`. The left sidebar contains a tree view of the API, with `DateFormat` selected under `java.text`. The main content area displays the class name `Class DateFormat`, its inheritance hierarchy (`java.lang.Object` → `java.text.Format` → `DateFormat`), and lists of implemented interfaces (`Serializable`, `Cloneable`) and subclasses (`SimpleDateFormat`). The class signature is `public abstract class DateFormat extends Format`. The description states: "DateFormat is an abstract class for date/time formatting subclasses which formats and parses dates or time in a language-independent manner. The date/time formatting subclass, such as SimpleDateFormat, allows for formatting (i.e., date → text), parsing (text → date), and normalization. The date is represented as a date object or as the milliseconds since January 1, 1970, 00:00:00 GMT."

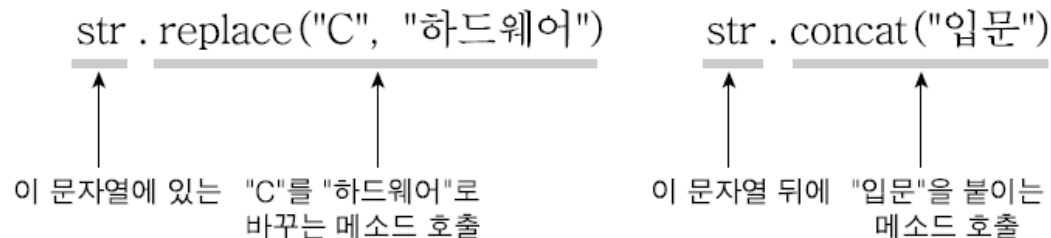
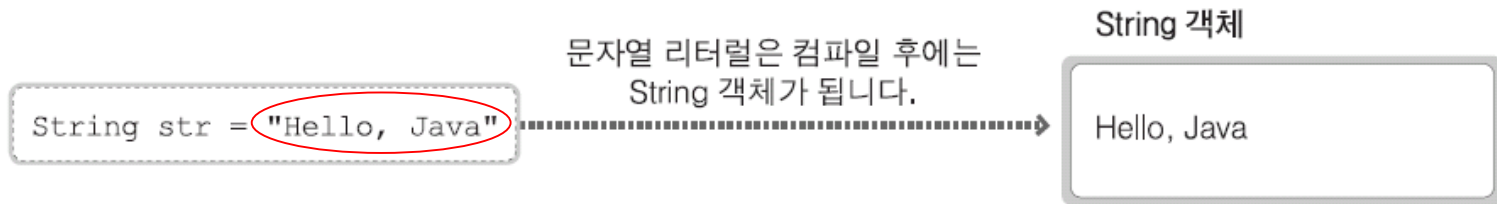
클래스 이름을 선택하면  
그에 대한 API 규격서를  
볼 수 있습니다.

# 기초적인 자바 클래스들

## 02. 문자열에 관련된 클래스들

### 문자열 관련 클래스들

- 자바 프로그램에 있는 문자열은 모두 `String` 클래스의 객체로 표현됨





# 기초적인 자바 클래스들

## 02. 문자열에 관련된 클래스들

### 문자열 관련 클래스들

- String 클래스는 문자열 조작에 적합하지 않음
  - 너무 많은 String 객체를 만들기 때문
- 문자열 조작에 적합한 클래스 : `StringBuilder` 클래스, `StringBuffer` 클래스
  - 객체를 많이 만들지 않고 문자열 조작이 가능
- 문자열로부터 작은 단위 문자열을 추출하는 클래스 : `StringTokenizer` 클래스
  - "사과 배 복숭아" 라는 문자열로부터 "사과" , "배" , "복숭아" 를 추출



# 기초적인 자바 클래스들

## 02. 문자열에 관련된 클래스들

### String 클래스

•• String 클래스의 API 규격서

The screenshot shows the Java API documentation for the `String` class. The left sidebar lists various packages and classes. The main content area is divided into sections: **Field Summary** (showing `CASE_INSENSITIVE_ORDER`), **Constructor Summary** (listing various constructors for creating `String` objects), and **Method Summary** (listing methods like `charAt`, `codePointAt`, etc.). A mouse cursor is pointing at the `charAt` method in the Method Summary section.

필드, 생성자, 메소드 이름을 클릭하면  
자세한 설명이 있는 부분으로 이동

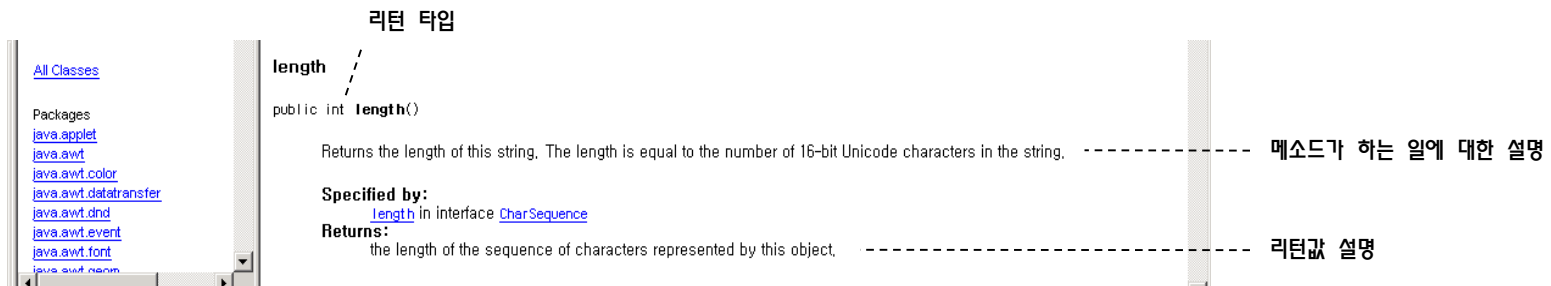
This screenshot shows the detailed API specification for the `charAt` method. It includes the method signature `public char charAt(int index)`, a description of its return value, and a list of exceptions it throws. The `Throws` section lists `IndexOutOfBoundsException` with a note that it is thrown if the `index` argument is negative or not less than the length of the string.

# 기초적인 자바 클래스들

## 02. 문자열에 관련된 클래스들

### String 클래스

- 문자열 길이를 가져오는 `length` 메소드



The image shows a screenshot of the Java API documentation for the `length` method of the `String` class. On the left, there is a sidebar with a list of packages including `java.applet`, `java.awt`, `java.awt.color`, `java.awt.datatransfer`, `java.awt.dnd`, `java.awt.event`, `java.awt.font`, and `java.awt.geom`. The main content area displays the method signature `length` with a return type of `public int`. Below the signature, a description states: "Returns the length of this string. The length is equal to the number of 16-bit Unicode characters in the string." To the right of this description, a dashed line points to the text "메소드가 하는 일에 대한 설명". Below the description, the "Specified by:" section indicates that `length` is specified in the `CharSequence` interface. The "Returns:" section states that it returns "the length of the sequence of characters represented by this object," with a dashed line pointing to the text "리턴값 설명". Above the method signature, the text "리턴 타입" is written with a dashed line pointing to the `length` property.

리턴 타입

`length`

`public int length()`

Returns the length of this string. The length is equal to the number of 16-bit Unicode characters in the string.

메소드가 하는 일에 대한 설명

Specified by:  
`length` in interface `CharSequence`

Returns:  
the length of the sequence of characters represented by this object.

리턴값 설명

# 기초적인 자바 클래스들

## 02. 문자열에 관련된 클래스들

### String 클래스

• 특징 위치의 문자를 가져오는 `charAt` 메소드

[StartTlsResponse](#)  
[State](#)  
[StateEdit](#)  
[StateEditable](#)  
[StateFactory](#)  
[Statement](#)  
[Statement](#)  
[Streamable](#)  
[StreamableValue](#)  
[StreamCorruptedException](#)  
[StreamHandler](#)  
[StreamPrintService](#)  
[StreamPrintServiceFactory](#)  
[StreamResult](#)  
[StreamSource](#)  
[StreamTokenizer](#)  
[StrictMath](#)  
[String](#)  
[StringBuffer](#)  
[StringBufferInputStream](#)  
[StringBuilder](#)

리턴 타입      파라미터

`charAt`

`public char charAt(int index)`

Returns the char value at the specified index. An index ranges from 0 to `length() - 1`. The first char value of the sequence is at index 0, the next at index 1, and so on, as for array indexing.

If the char value specified by the index is a [surrogate](#), the surrogate value is returned.

**Specified by:**

[charAt](#) in interface [CharSequence](#)

**Parameters:**

`index` - the index of the char value.

**Returns:**

the char value at the specified index of this string. The first char value is at index 0.

**Throws:**

[IndexOutOfBoundsException](#) - if the `index` argument is negative or not less than the length of this string.

} 메소드가 하는 일에 대한 설명

파라미터 설명

리턴값 설명

예외선 설명

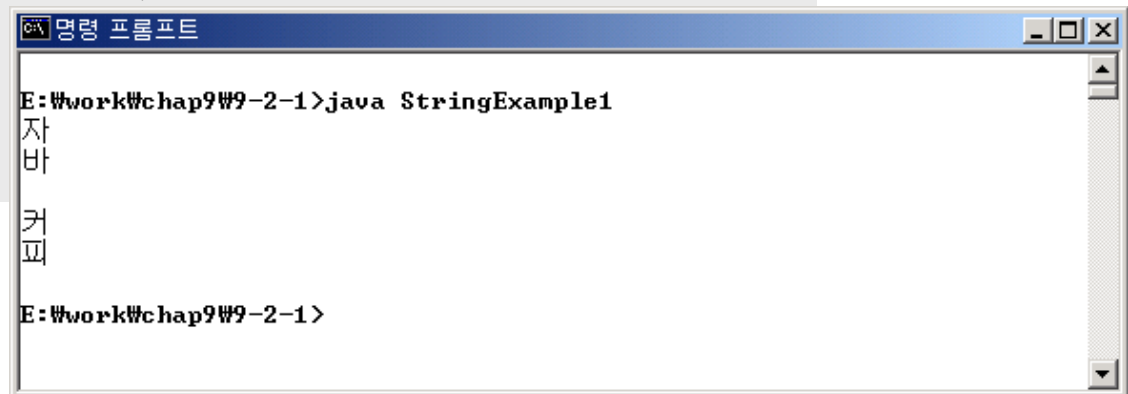
# 기초적인 자바 클래스들

## 02. 문자열에 관련된 클래스들

### String 클래스

- [예제 9-1] String 클래스의 `length`, `charAt` 메소드 사용 예

```
1  class StringExample1 {  
2      public static void main(String args[]) {  
3          String str = "자바 커피";  
4          int len = str.length();           // length 메소드 호출  
5          for (int cnt = 0; cnt < len; cnt++) {  
6              char ch = str.charAt(cnt);     // charAt 메소드 호출  
7              System.out.println(ch);  
8          }  
9      }  
10 }
```



```
명령 프롬프트  
E:\work\chap9\9-2-1>java StringExample1  
자  
바  
커피  
E:\work\chap9\9-2-1>
```

# 기초적인 자바 클래스들

## 02. 문자열에 관련된 클래스들

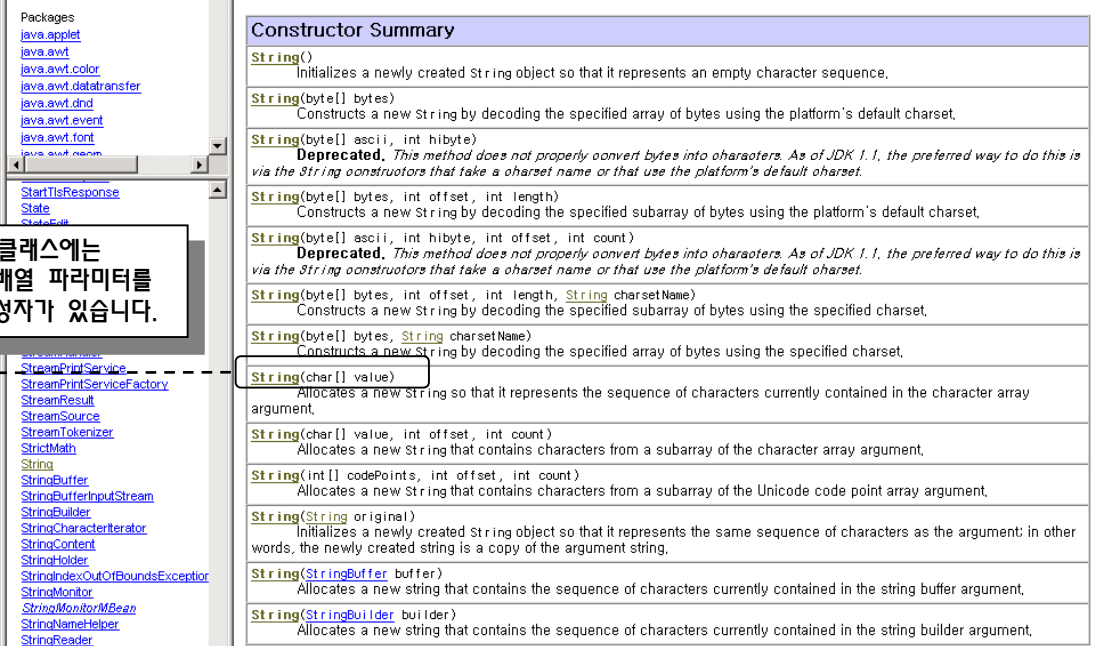
### String 클래스

• String 클래스의 생성자 (1)

String 클래스에는 char 배열 파라미터를 받는 생성자가 있습니다.

```
String str = new String(arr);
```

그 생성자를 이용하여 String 객체를 생성합니다.



Constructor Summary	
<b>String()</b>	Initializes a newly created String object so that it represents an empty character sequence.
<b>String(byte[] bytes)</b>	Constructs a new String by decoding the specified array of bytes using the platform's default charset.
<b>String(byte[] ascii, int hibyte)</b>	<b>Deprecated.</b> This method does not properly convert bytes into characters. As of JDK 1.1, the preferred way to do this is via the String constructors that take a charset name or that use the platform's default charset.
<b>String(byte[] bytes, int offset, int length)</b>	Constructs a new String by decoding the specified subarray of bytes using the platform's default charset.
<b>String(byte[] ascii, int hibyte, int offset, int count)</b>	<b>Deprecated.</b> This method does not properly convert bytes into characters. As of JDK 1.1, the preferred way to do this is via the String constructors that take a charset name or that use the platform's default charset.
<b>String(byte[] bytes, int offset, int length, String charsetName)</b>	Constructs a new String by decoding the specified subarray of bytes using the specified charset.
<b>String(byte[] bytes, String charsetName)</b>	Constructs a new String by decoding the specified array of bytes using the specified charset.
<b>String(char[] value)</b>	Allocates a new String so that it represents the sequence of characters currently contained in the character array argument.
<b>String(char[] value, int offset, int count)</b>	Allocates a new String that contains characters from a subarray of the character array argument.
<b>String(int[] codePoints, int offset, int count)</b>	Allocates a new String that contains characters from a subarray of the Unicode code point array argument.
<b>String(String original)</b>	Initializes a newly created String object so that it represents the same sequence of characters as the argument; in other words, the newly created string is a copy of the argument string.
<b>String(StringBuffer buffer)</b>	Allocates a new string that contains the sequence of characters currently contained in the string buffer argument.
<b>String(StringBuilder builder)</b>	Allocates a new string that contains the sequence of characters currently contained in the string builder argument.

## 기초적인 자바 클래스들

## 02. 문자열에 관련된 클래스들

## String 클래스

## •• String 클래스의 생성자 (1)

String 클래스에는 String 타입의 파라미터를 받는 생성자도 있습니다.

Constructor Summary	
<b>String()</b>	Initializes a newly created String object so that it represents an empty character sequence.
<b>String(byte[] bytes)</b>	Constructs a new String by decoding the specified array of bytes using the platform's default charset.
<b>String(byte[] ascii, int hibyte)</b>	<b>Deprecated.</b> This method does not properly convert bytes into characters. As of JDK 1.1, the preferred way to do this is via the String constructors that take a charset name or that use the platform's default charset.
<b>String(byte[] bytes, int offset, int length)</b>	Constructs a new String by decoding the specified subarray of bytes using the platform's default charset.
<b>String(byte[] ascii, int hibyte, int offset, int count)</b>	<b>Deprecated.</b> This method does not properly convert bytes into characters. As of JDK 1.1, the preferred way to do this is via the String constructors that take a charset name or that use the platform's default charset.
<b>String(byte[] bytes, int offset, int length, String charsetName)</b>	Constructs a new String by decoding the specified subarray of bytes using the specified charset.
<b>String(byte[] bytes, String charsetName)</b>	Constructs a new String by decoding the specified array of bytes using the specified charset.
<b>String(char[] value)</b>	Allocates a new String so that it represents the sequence of characters currently contained in the character array argument.
<b>String(char[] value, int offset, int count)</b>	Allocates a new String that contains characters from a subarray of the character array argument.
<b>String(int[] codePoints, int offset, int count)</b>	Allocates a new String that contains characters from a subarray of the Unicode code point array argument.
<b>String(String original)</b>	Initializes a newly created String object so that it represents the same sequence of characters as the argument; in other words, the newly created string is a copy of the argument string.
<b>String(StringBuffer buffer)</b>	Allocates a new string that contains the sequence of characters currently contained in the string buffer argument.
<b>String(StringBuilder builder)</b>	Allocates a new string that contains the sequence of characters currently contained in the string builder argument.

String str = new String("Hello, Java");

-----

그 생성자를 이용하여 String 객체를 생성합니다.

# 기초적인 자바 클래스들

## 02. 문자열에 관련된 클래스들

### String 클래스

•• 다음 명령문들을 실행하면 몇 개의 String 객체가 생길까요?

```
str1 = new String("Hello, Java");  
str2 = new String("Hello, Java");  
str3 = new String("Hello, Java");
```

정답 : 4개

왜 6개가 아니라 4개일까요?

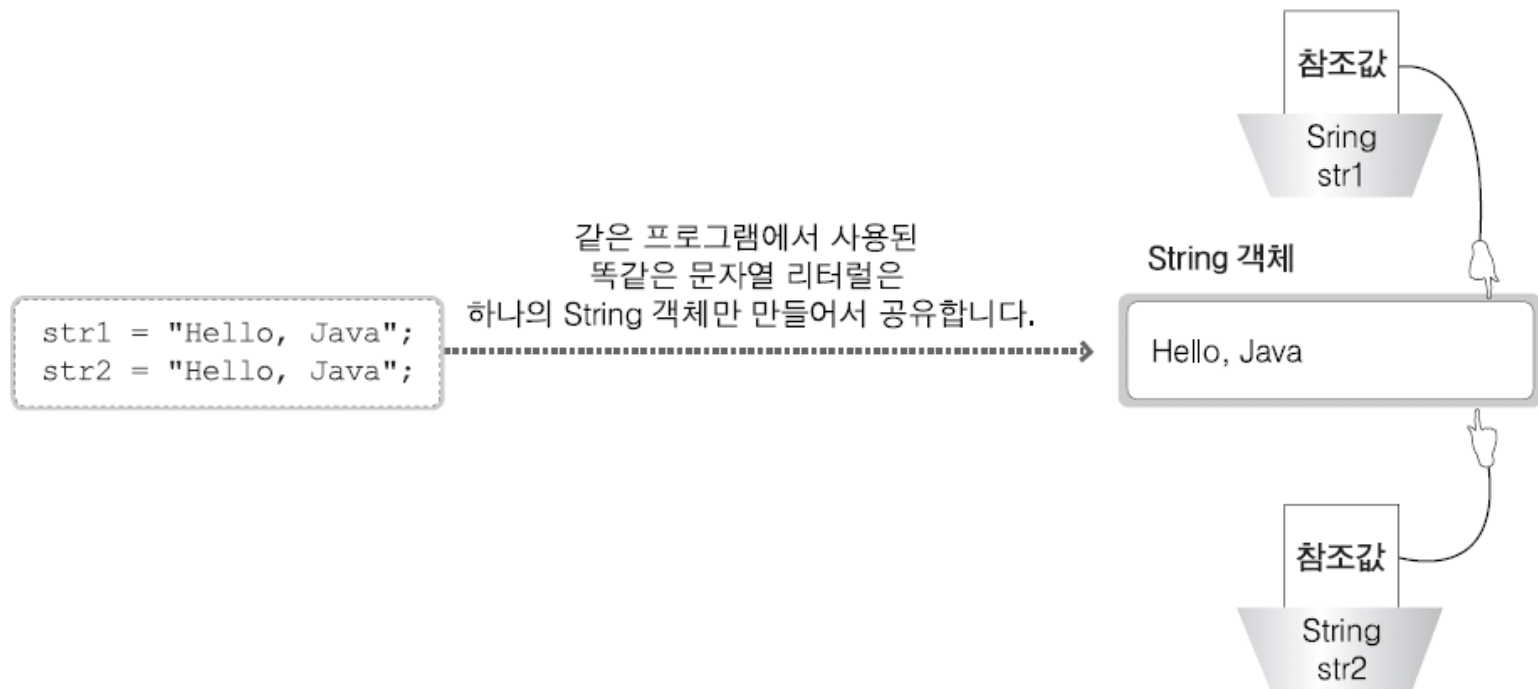


# 기초적인 자바 클래스들

## 02. 문자열에 관련된 클래스들

### String 클래스

• 문자열 리터럴이 String 객체로 만들어지는 방법



# 기초적인 자바 클래스들

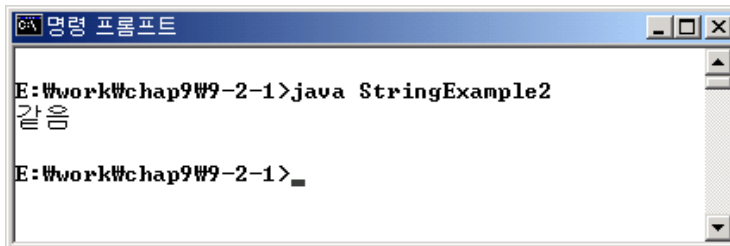
## 02. 문자열에 관련된 클래스들

### String 클래스 : 문자열 비교

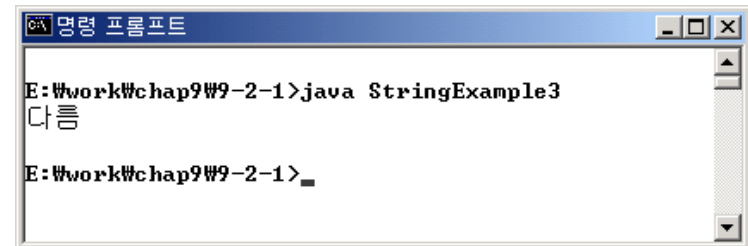
- [예제 9-2] 동등 연산자를 이용한 문자열 비교 프로그램

```
class StringExample2 {  
    public static void main(String args[]) {  
        String str1 = "자바";  
        String str2 = "자바";  
        if (str1 == str2)  
            System.out.println("같음");  
        else  
            System.out.println("다름");  
    }  
}
```

```
class StringExample3 {  
    public static void main(String args[]) {  
        String str1 = new String("자바");  
        String str2 = new String("자바");  
        if (str1 == str2)  
            System.out.println("같음");  
        else  
            System.out.println("다름");  
    }  
}
```



```
명령 프롬프트  
E:\work\chap9\9-2-1>java StringExample2  
같음  
E:\work\chap9\9-2-1>
```



```
명령 프롬프트  
E:\work\chap9\9-2-1>java StringExample3  
다름  
E:\work\chap9\9-2-1>
```

# 기초적인 자바 클래스들

## 02. 문자열에 관련된 클래스들

### String 클래스 : 문자열 비교

- 문자열 내용을 비교하는 `equals` 메소드

`str1.equals( str2 )`

이 문자열과 이 문자열을 비교합니다.

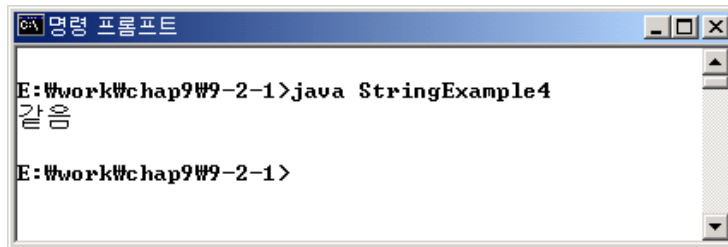
# 기초적인 자바 클래스들

## 02. 문자열에 관련된 클래스들

### String 클래스 : 문자열 비교

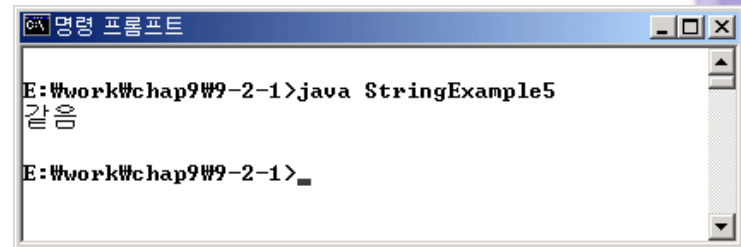
- [예제 9-3] `equals` 메소드를 이용한 문자열 비교 프로그램

```
class StringExample4 {  
    public static void main(String args[]) {  
        String str1 = "자바";  
        String str2 = "자바";  
        if (str1.equals(str2))  
            System.out.println("같음");  
        else  
            System.out.println("다름");  
    }  
}
```



```
명령 프롬프트  
E:\work\chap9\9-2-1>java StringExample4  
같음  
E:\work\chap9\9-2-1>
```

```
class StringExample5 {  
    public static void main(String args[]) {  
        String str1 = new String("자바");  
        String str2 = new String("자바");  
        if (str1.equals(str2))  
            System.out.println("같음");  
        else  
            System.out.println("다름");  
    }  
}
```



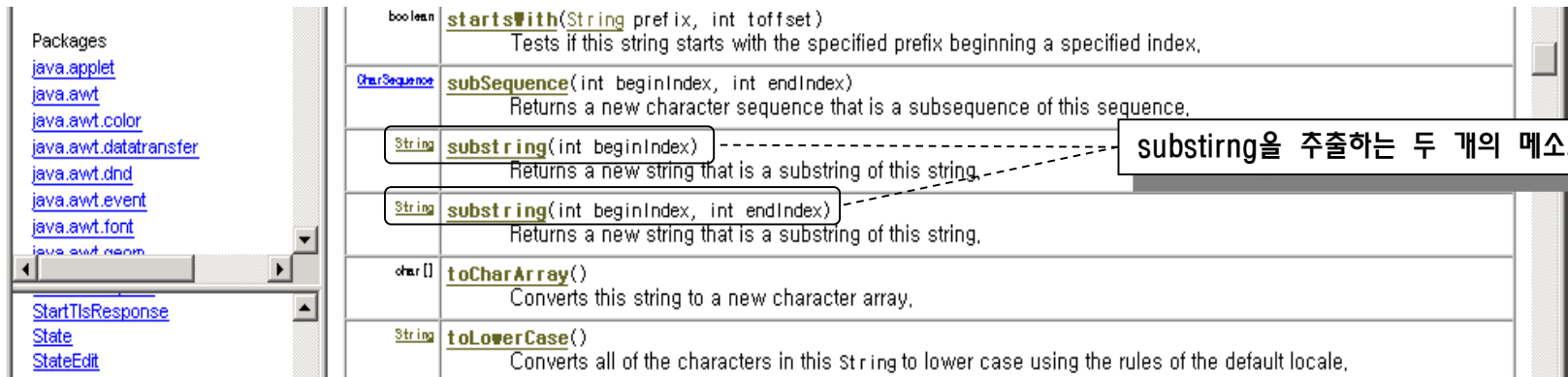
```
명령 프롬프트  
E:\work\chap9\9-2-1>java StringExample5  
같음  
E:\work\chap9\9-2-1>
```

# 기초적인 자바 클래스들

## 02. 문자열에 관련된 클래스들

### String 클래스 : 문자열 추출

- 부분 문자열(substring)을 추출하는 substring 메소드



The screenshot shows the Java API documentation for the `String` class. On the left, a 'Packages' list includes `java.applet`, `java.awt`, `java.awt.color`, `java.awt.datatransfer`, `java.awt.dnd`, `java.awt.event`, `java.awt.font`, and `java.awt.geom`. The main area displays the following methods:

- `boolean startsWith(String prefix, int toffset)`: Tests if this string starts with the specified prefix beginning a specified index.
- `CharSequence subSequence(int beginIndex, int endIndex)`: Returns a new character sequence that is a subsequence of this sequence.
- `String substring(int beginIndex)`: Returns a new string that is a substring of this string.
- `String substring(int beginIndex, int endIndex)`: Returns a new string that is a substring of this string.
- `char[] toCharArray()`: Converts this string to a new character array.
- `String toLowerCase()`: Converts all of the characters in this string to lower case using the rules of the default locale.

A callout box with dashed lines pointing to the two `substring` methods contains the text: "substirng을 추출하는 두 개의 메소드".

# 기초적인 자바 클래스들

## 02. 문자열에 관련된 클래스들

### String 클래스 : 문자열 추출

- [예제 9-4] String 클래스의 substring 메소드 사용 예

```
1  class StringExample6 {  
2      public static void main(String args[]) {  
3          String str = "뇌를 자극하는 자바";  
4          System.out.println(str.substring(3));  
5          System.out.println(str.substring(3, 7));  
6      }  
7  }
```



```
명령 프롬프트  
E:\work\chap9\9-2-1>java StringExample6  
자극하는 자바  
자극하는  
E:\work\chap9\9-2-1>
```

# 기초적인 자바 클래스들

## 02. 문자열에 관련된 클래스들

### String 클래스 : 문자열 조작

•• String 클래스의 문자열 조작 메소드

메소드	기능
String concat(Strin str)	문자열 뒤에 str을 연결
String trim()	문자열의 앞뒤 공백을 제거
String toUpperCase()	문자열의 모든 소문자를 대문자로 교체
String toLowerCase()	문자열의 모든 대문자를 소문자로 교체
String replace(char oldChar, char newChar)	문자열에 포함된 oldChar를 모두 newChar로 교체

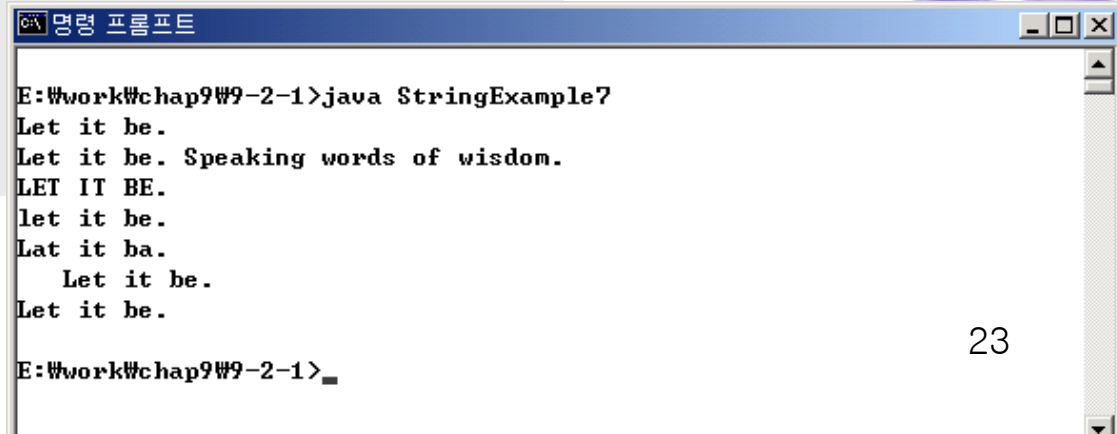


# 기초적인 자바 클래스들

## ● String 클래스 : 문자열 조작

- [예제 9-5] String 클래스의 문자열 조작 메소드 사용 예

```
1  class StringExample7 {  
2      public static void main(String args[]) {  
3          String str1 = "  Let it be.  ";  
4          String str2 = str1.trim();  
5          System.out.println(str2);  
6          System.out.println(str2.concat(" Speaking words of wisdom."));  
7          System.out.println(str2.toUpperCase());  
8          System.out.println(str2.toLowerCase());  
9          System.out.println(str2.replace('e', 'a'));  
10         System.out.println(str1);  
11         System.out.println(str2);  
12     }  
13 }
```



```
E:\work\chap9\9-2-1>java StringExample7  
Let it be.  
Let it be. Speaking words of wisdom.  
LET IT BE.  
let it be.  
Lat it ba.  
Let it be.  
Let it be.  
  
E:\work\chap9\9-2-1>
```

# 기초적인 자바 클래스들

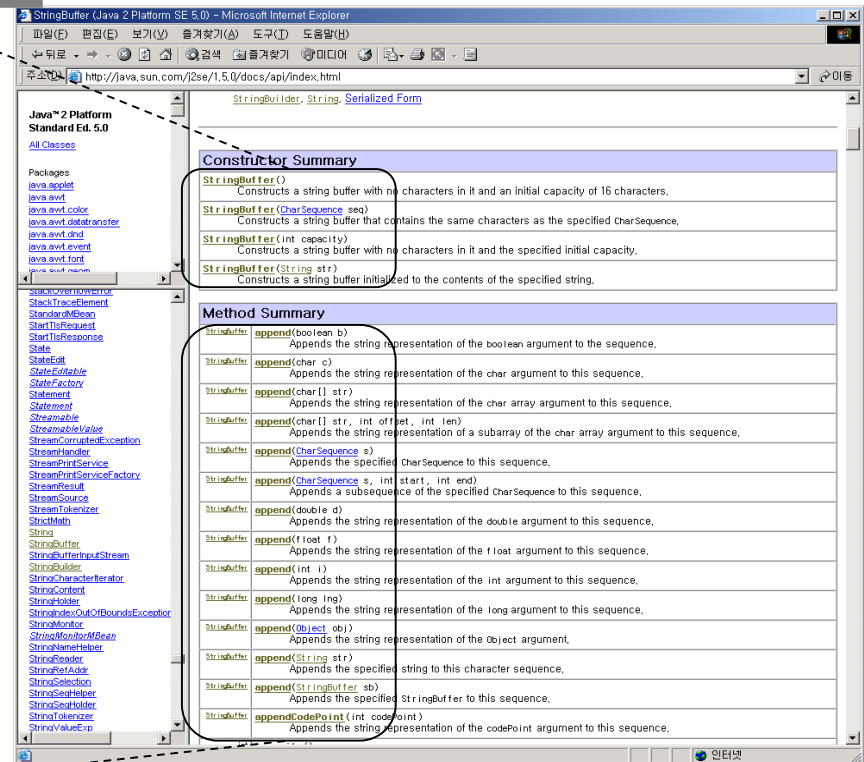
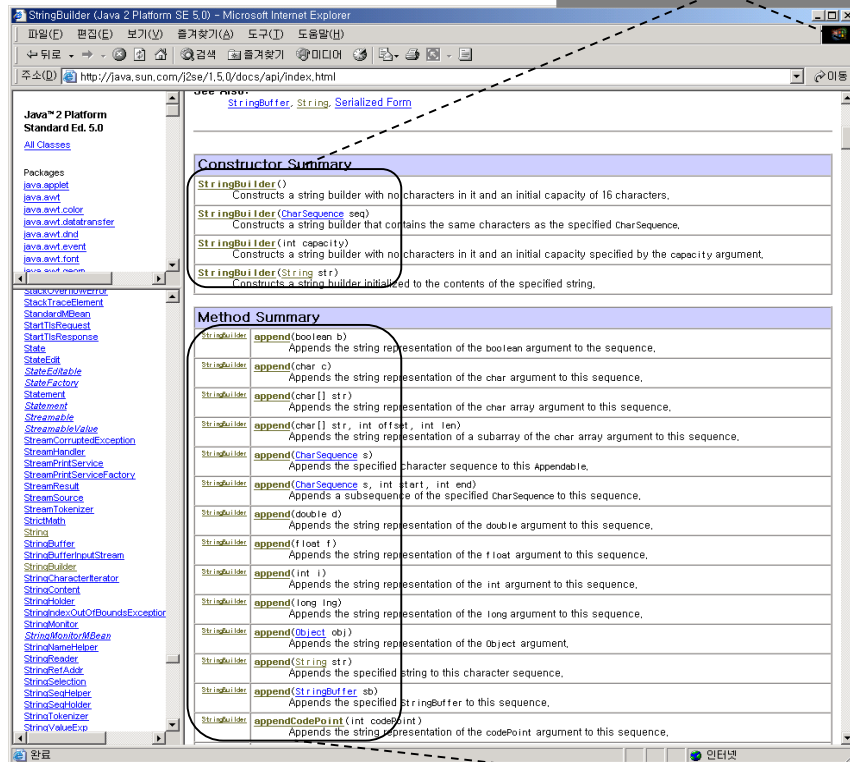
## StringBuilder 클래스와 StringBuffer 클래스

•• StringBuffer 클래스와 StringBuilder 클래스의 유사성

[StringBuilder 클래스의 API 규격서]

생성자의 사용 방법이 같습니다.

[StringBuffer 클래스의 API 규격서]



메소드의 기능과 사용 방법도 같습니다.

# 기초적인 자바 클래스들

## 02. 문자열에 관련된 클래스들

### StringBuilder 클래스

•• StringBuilder 클래스의 생성자

```
sb1 = new StringBuilder("Hello");           // 크기가 11인 버퍼 생성  
sb2 = new StringBuilder(100);               // 크기가 100인 버퍼 생성  
sb3 = new StringBuilder();                  // 크기가 16인 버퍼 생성
```

# 기초적인 자바 클래스들

## 02. 문자열에 관련된 클래스들

### StringBuilder 클래스

•• StringBuilder 클래스의 문자열 조작 메소드

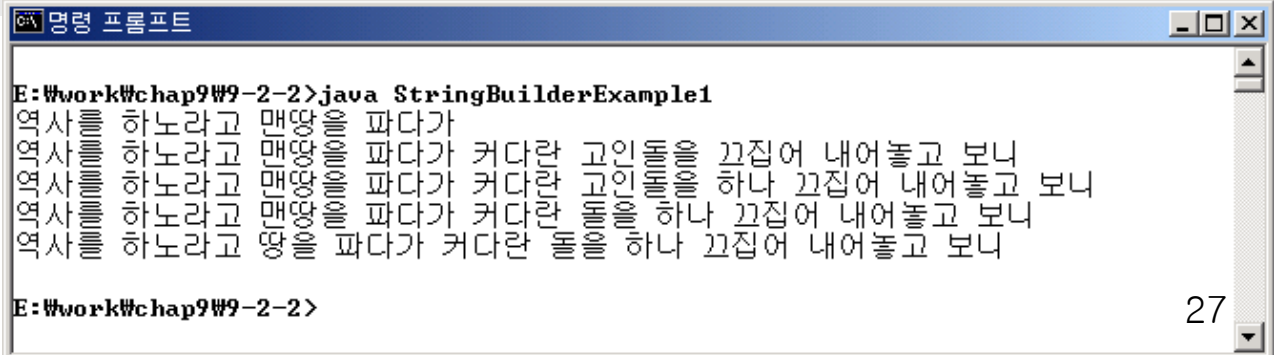
메소드	기능
StringBuilder append(String str)	문자열 뒤에 str을 덧붙임
StringBuilder insert(int offset, String str)	문자열의 offset 위치에 str을 삽입
StringBuilder delete(int start, int end)	start부터 end - 1까지의 부분 문자열을 삭제
StringBuilder deleteCharAt(int index)	index 위치에 있는 하나의 문자를 삭제

# 기초적인 자바 클래스들

## ● StringBuilder 클래스

- [예제 9-6] StringBuilder 클래스를 이용하여 문자열을 조작하는 프로그램

```
1  class StringBuilderExample1 {  
2      public static void main(String args[]) {  
3          StringBuilder sb = new StringBuilder("역사를 하노라고 맨땅을 파다가 ");  
4          System.out.println(sb);  
5          System.out.println(sb.append("커다란 고인돌을 끄집어 내어놓고 보니"));  
6          System.out.println(sb.insert(26, "하나 "));  
7          System.out.println(sb.delete(21, 23));  
8          System.out.println(sb.deleteCharAt(9));  
9      }  
10 }
```



```
명령 프롬프트  
E:\work\chap9\9-2>java StringBuilderExample1  
역사를 하노라고 맨땅을 파다가  
역사를 하노라고 맨땅을 파다가 커다란 고인돌을 끄집어 내어놓고 보니  
역사를 하노라고 맨땅을 파다가 커다란 고인돌을 하나 끄집어 내어놓고 보니  
역사를 하노라고 맨땅을 파다가 커다란 돌을 하나 끄집어 내어놓고 보니  
역사를 하노라고 땅을 파다가 커다란 돌을 하나 끄집어 내어놓고 보니  
E:\work\chap9\9-2>
```

# 기초적인 자바 클래스들

## 02. 문자열에 관련된 클래스들

### StringBuilder 클래스

- 버퍼 크기를 리턴하는 `capacity` 메소드

```
int bufSize = sb.capacity(); // sb의 버퍼 크기를 리턴하는 메소드
```

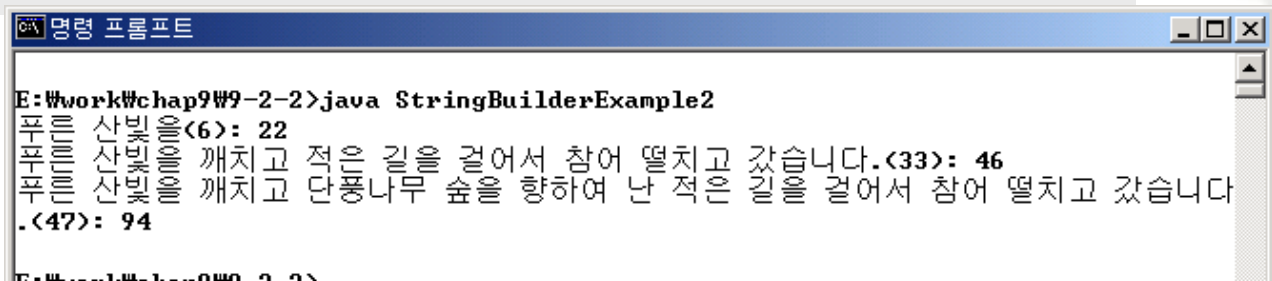
# 기초적인 자바 클래스들

## StringBuilder 클래스

- [예제 9-7] StringBuilder 객체의 버퍼 크기를 출력하는 프로그램

```
1 class StringBuilderExample2 {
2     public static void main(String args[]) {
3         StringBuilder sb = new StringBuilder("푸른 산빛을");
4         printStringBuilder(sb);
5         sb.append(" 깨치고 적은 길을 걸어서 참어 떨치고 갔습니다.");
6         printStringBuilder(sb);
7         sb.insert(10, " 단풍나무 숲을 향하여 난");
8         printStringBuilder(sb);
9     }
10    static void printStringBuilder(StringBuilder sb) {
11        String str = sb.toString(); ----- 버퍼의 문자열을 String 객체로 만들어서 리턴하는 메소드
12        int len = sb.length(); ----- 버퍼의 문자열 길이를 리턴하는 메소드
13        int bufSize = sb.capacity();
14        System.out.printf("%s(%d): %d %n", str, len, bufSize); -----
15    }
16 }
```

“문자열(길이): 버퍼크기”  
포맷으로 출력



```
E:\work\chap9\9-2-2>java StringBuilderExample2
푸른 산빛을(6): 22
푸른 산빛을 깨치고 적은 길을 걸어서 참어 떨치고 갔습니다.(33): 46
푸른 산빛을 깨치고 단풍나무 숲을 향하여 난 적은 길을 걸어서 참어 떨치고 갔습니다.
(47): 94
E:\work\chap9\9-2-2>
```



# 기초적인 자바 클래스들

## 02. 문자열에 관련된 클래스들

### StringBuilder 클래스

- 버퍼 크기를 늘리는 `ensureCapacity` 메소드

```
sb. ensureCapacity(100);
```

↑  
버퍼의 크기를 100 또는  
그보다 더 크게 늘립니다.

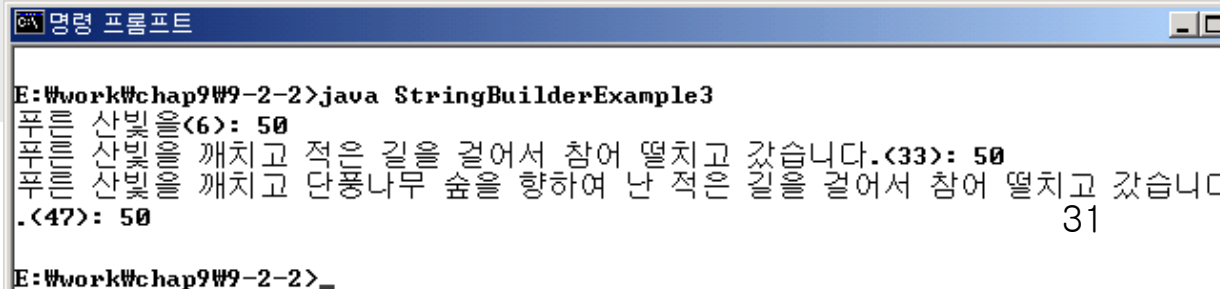
파라미터가 ((원래의 버퍼크기 X 2) + 2)보다 작으면;  
((원래의 버퍼크기 X 2) + 2)로 버퍼의 크기를 늘임

# 기초적인 자바 클래스들

## StringBuilder 클래스

- [예제 9-8] StringBuilder 객체의 버퍼 크기를 인위적으로 늘리는 프로그램

```
1  class StringBuilderExample3 {
2      public static void main(String args[]) {
3          StringBuilder sb = new StringBuilder("푸른 산빛을");
4          sb.ensureCapacity(50);
5          printStringBuilder(sb);
6          sb.append(" 깨치고 적은 길을 걸어서 참어 떨치고 갔습니다.");
7          printStringBuilder(sb);
8          sb.insert(10, " 단풍나무 숲을 향하여 난");
9          printStringBuilder(sb);
10     }
11     static void printStringBuilder(StringBuilder sb) {
12         String str = sb.toString();
13         int len = sb.length();           // length는 문자열의 길이를 리턴하는 메소드
14         int bufSize = sb.capacity();
15         System.out.printf("%s(%d): %d %n", str, len, bufSize);
16     }
17 }
```



```
E:\work\chap9\9-2-2>java StringBuilderExample3
푸른 산빛을(6): 50
푸른 산빛을 깨치고 적은 길을 걸어서 참어 떨치고 갔습니다.(33): 50
푸른 산빛을 깨치고 단풍나무 숲을 향하여 난 적은 길을 걸어서 참어 떨치고 갔습니다.(47): 50
E:\work\chap9\9-2-2>
```

# 기초적인 자바 클래스들

## 02. 문자열에 관련된 클래스들

### StringBuilder 클래스

- [예제 9-9] StringBuilder 객체의 버퍼 크기를 문자열에 맞게 줄이는 프로그램

```
1  class StringBuilderExample4 {  
2      public static void main(String args[]) {  
3          StringBuilder sb = new StringBuilder(100);  
4          sb.append("자바");  
5          System.out.println(sb + ": " + sb.capacity());  
6          sb.trimToSize(); ----- 과도한 버퍼 크기를 적당하게 줄이는 메소드  
7          System.out.println(sb + ": " + sb.capacity());  
8      }  
9  }
```

```
명령 프롬프트  
E:\work\chap9\9-2-2>java StringBuilderExample4  
자바: 100  
자바: 2  
E:\work\chap9\9-2-2>
```

# 기초적인 자바 클래스들

## 02. 문자열에 관련된 클래스들

### StringTokenizer 클래스

- 문자열로부터 토큰(token)을 추출하는 기능이 있는 클래스



문자열을 구성하는 단위

- [예] “토끼 강아지 고양이” 로부터 “토끼” , “강아지” , “고양이” 를 추출

# 기초적인 자바 클래스들

## 02. 문자열에 관련된 클래스들

### StringTokenizer 클래스

#### •• 사용 방법

- 1) StringTokenizer 객체를 생성

```
StringTokenizer stok = new StringTokenizer("사과 배 복숭아");
```

↑  
토큰을 추출할 문자열을 가지고  
StringTokenizer 객체를 생성합니다.

- 2) nextToken 메소드를 이용하여 토큰 추출

```
str1 = stok.nextToken(); // 첫 번째 토큰인 "사과"를 리턴  
str2 = stok.nextToken(); // 두 번째 토큰인 "배"를 리턴  
str3 = stok.nextToken(); // 세 번째 토큰인 "복숭아"를 리턴
```

더 이상 추출할 토큰이 없으면 NoSuchElementException 발생

# 기초적인 자바 클래스들

## 02. 문자열에 관련된 클래스들

### StringTokenizer 클래스

•• NoSuchElementException이 발생하지 않게 하려면;

```
while (stok.hasMoreTokens()) { // 토큰이 있는 동안만 while 문 안에서
    str = stok.nextToken();      // 토큰을 추출하여
    System.out.println(str);     // 처리합니다.
}
```

# 기초적인 자바 클래스들

## 02. 문자열에 관련된 클래스들

### StringTokenizer 클래스

- [예제 9-10] StringTokenizer를 이용해서 문자열로부터 토큰을 분리하는 프로그램

```
1  import java.util.StringTokenizer;
2  class StringTokenizerExample1 {
3      public static void main(String args[]) {
4          StringTokenizer stok = new StringTokenizer("사과 배 복숭아");
5          while (stok.hasMoreTokens()) {
6              String str = stok.nextToken();
7              System.out.println(str);
8          }
9      }
10 }
```



```
명령 프롬프트
E:\work\chap9\9-2-3>java StringTokenizerExample1
사과
배
복숭아
E:\work\chap9\9-2-3>
```



# 기초적인 자바 클래스들

## 02. 문자열에 관련된 클래스들

### StringTokenizer 클래스

- 구획 문자(delimiter) 지정하기

```
stok = new StringTokenizer("사과,배,복숭아", ",");
```

이 문자열에 있는 콤마(,)를 가지고  
토큰을 추출하는 StringTokenizer 객체를 생성합니다.


# 기초적인 자바 클래스들

## 02. 문자열에 관련된 클래스들

### StringTokenizer 클래스

- [예제 9-11] 공백문자가 아닌 구획문자를 이용하여 토큰을 추출하는 예

```
1  import java.util.StringTokenizer;
2  class StringTokenizerExample2 {
3      public static void main(String args[]) {
4          StringTokenizer stok = new StringTokenizer("사과,배,복숭아", ",");
5          while (stok.hasMoreTokens()) {
6              String str = stok.nextToken();
7              System.out.println(str);
8          }
9      }
10 }
```



```
명령 프롬프트
E:\work\chap9\9-2-3>java StringTokenizerExample2
사과
배
복숭아
E:\work\chap9\9-2-3>
```

# 기초적인 자바 클래스들

## 02. 문자열에 관련된 클래스들

### StringTokenizer 클래스

- 여러 개의 구획 문자(delimiter) 지정하기

```
stok = new StringTokenizer("사과,배|복숭아", ",|");
```

이 문자열에 있는 콤마(,)와 수직선(|)을 가지고  
토큰을 추출하는 StringTokenizer 객체를 생성합니다.

# 기초적인 자바 클래스들

## 02. 문자열에 관련된 클래스들

### StringTokenizer 클래스

- 구획 문자(delimiter)도 토큰으로 추출하기

```
stok = new StringTokenizer("사과=10| 초콜렛=3| 샴페인=1", "=| ", true);
```

↑  
토큰을 추출할  
문자열

↑  
구획문자

↑  
구획문자도 토큰으로  
추출하도록 만드는 true 파라미터

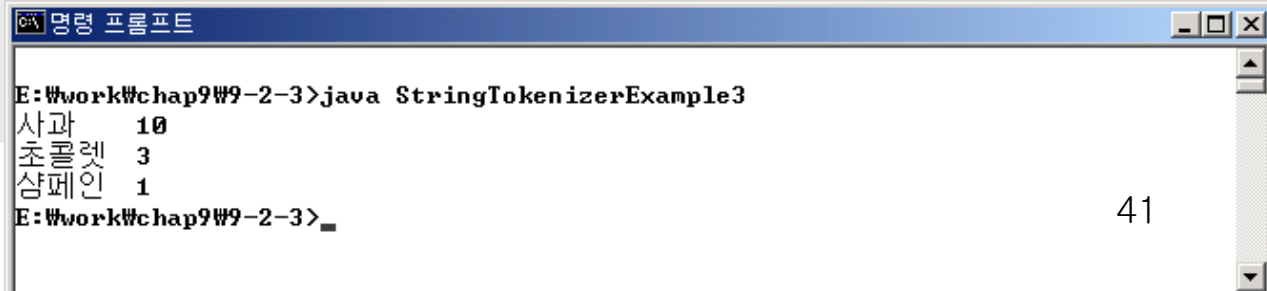
# 기초적인 자바 클래스들

## StringTokenizer 클래스

- [예제 9-12] 구획문자를 토큰으로 추출하는 예

```
1  import java.util.*;
2  class StringTokenizerExample3 {
3      public static void main(String args[]) {
4          StringTokenizer stok = new StringTokenizer("사과=10|초콜렛=3|삼페인=1", "=", true);
5          while (stok.hasMoreTokens()) {
6              String token = stok.nextToken();
7              if (token.equals("="))
8                  System.out.print("Wt");
9              else if (token.equals("|"))
10                 System.out.print("Wn");
11             else
12                 System.out.print(token);
13         }
14     }
15 }
```

토큰이 "="이면 탭 문자,  
"|"이면 줄 바꿈 문자,  
그 밖의 문자열이면 토큰을 출력합니다.



```
E:\work\chap9\9-2-3>java StringTokenizerExample3
사과      10
초콜렛    3
삼페인    1
E:\work\chap9\9-2-3>
```

# Calendar class

## ● Calendar

- ▶ Calendar class는 1970년 1월 1일부터 특정 값으로 진보해 오면서 날짜와 시각에 대한 조작을 수행할 수 있도록 제공되는 abstract class이다.
- ▶ Object 생성 법은 다음과 같다.

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

### ※ Calendar 객체의 주요 메서드

반환형	메서드명	설명
boolean	after (Object when)	현재 Calendar 객체가 인자로 전달된 when 객체의 날짜보다 후의 시각이라면 true, 그렇지 않으면 false를 반환한다.
	before (Object when)	현재 Calendar 객체가 인자로 전달된 when 객체의 날짜보다 전의 시각이라면 true, 그렇지 않으면 false를 반환한다.
int	get(int field)	인자로 전달된 field(년, 월, 일, 시, 분, 초)의 값을 반환한다.
static Calendar	getInstance( )	default TimeZone과 Locale을 사용해 Calendar 객체를 반환한다.
void	set(int year, int month, int date)	현 Calendar 객체의 필드 중 년도와 월, 그리고 일(DAY_OF_MONTH)을 인자로 전달된 값을 설정한다.
	setTimeInMillis (long millis)	Calendar의 현재 시각을 인자로 전달된 long형의 값으로 설정한다.

# Calendar class

- `get()` method를 통해 현재 `Calendar Object`가 기억하고 있는 `Field`(년, 월, 일, 시, 분, 초)의 값을 얻어낼 수가 있다. 얻고자 하는 `Field`의 `Constant`를 정확하게 알고 있어야 하는데 API문서의 `Field Summary`에서 확인이 가능하다.

## ※ Calendar 객체 주요 상수들

상수	설명
AM_PM	HOUR가 정오보다 전인지, 후인지를 나타내주는 상수이며, 만약 정오 전일 경우(AM)에는 0을 반환하고 정오이후일 경우(PM)는 1을 반환한다.
DAY_OF_MONTH	한달 중의 날짜 수를 의미하는 상수(1~31)
DAY_OF_WEEK	SUNDAY에서 SATURDAY까지의 요일을 나타내는 상수(1~7)
DAY_OF_YEAR	한해 중 몇 일째인지를 나타내는 상수
DAY_OF_WEEK_IN_MONTH	한달 중 해당 요일이 몇 번째인지 나타내는 상수
HOUR	오전 또는 오후의 의미가 없이 몇 시인지를 나타내는 상수(0~11)
HOUR_OF_DAY	하루 중 시각을 나타내는 상수(0~23)
MONTH	달을 나타내는 상수이며 1월이 0이다(0~11).
WEEK_OF_MONTH	현재 달에서 몇 주째인지를 나타내는 상수
WEEK_OF_YEAR	현재 해에서 몇 주째인지를 나타내는 상수

# Calendar class

```
01 import java.util.Calendar;
02 import static java.lang.System.out;
03 public class CalendarEx1 {
04
05     public static void main(String[] args) {
06         StringBuffer sb = new StringBuffer("년중 ");
07         Calendar now = Calendar.getInstance();
08
09         int week_yy = now.get(now.WEEK_OF_YEAR);
10         int yy = now.get(now.YEAR);
11         int mm = now.get(now.MONTH)+1; //1월이 0을 기억한다.
12         int dd = now.get(now.DAY_OF_MONTH);
13
14         sb.append(week_yy);
15         sb.append("주째인 ");
16         sb.append(yy);
17         sb.append("년 ");
18         sb.append(mm);
```



# Calendar class

```
19         sb.append("월 ");
20         sb.append(dd);
21         sb.append("일");
22
23         out.println(sb.toString());
24     }
25 }
```

- Calendar Object에서의 월은 1월을 0으로 기억하고 있으며 우리가 인식하고 있는 월의 값을 구하기 위해 1을 더해야 함.
- 반복문이 아니므로 바로 출력해도 상관 없겠지만 복습하는 차원에서 memory 공간 낭비에 대한 고려로 StringBuffer Object에 변수 week\_yy값과 yy, mm, dd값들과 몇 개의 문자열들을 중간에 넣어 추가하였다.

# 기초적인 자바 클래스들

## 03. 날짜와 시간에 관련된 클래스들

### 시스템 시계와 날짜/시간 관련 클래스

#### • 시스템 시계(system clock)

- 컴퓨터에 내장되어 있는 현재 시각을 가리키는 시계

#### • Calendar 클래스와 GregorianCalendar 클래스

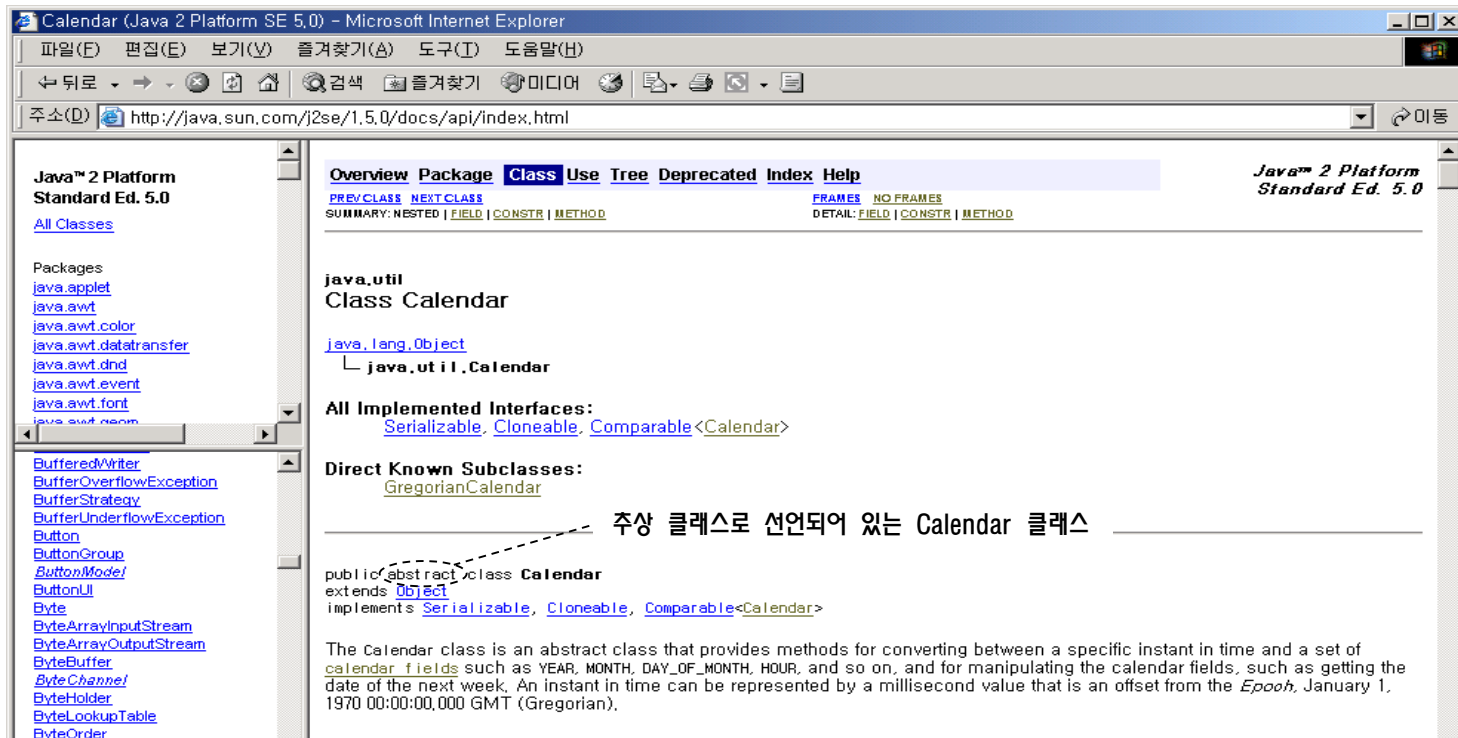
- 시스템 시계로부터 현재 시각을 읽어오는 기능의 클래스

# 기초적인 자바 클래스들

## 03. 날짜와 시간에 관련된 클래스들

### Calendar 클래스

• Calendar 클래스 : 날짜와 시간 계산에 필요한 일반적인 기능만을 제공



The screenshot shows the Java 2 Platform Standard Ed. 5.0 API documentation for the `Calendar` class. The browser window title is "Calendar (Java 2 Platform SE 5.0) - Microsoft Internet Explorer". The address bar shows the URL "http://java.sun.com/j2se/1.5.0/docs/api/index.html". The left sidebar lists various Java packages and classes, including `java.applet`, `java.awt`, `java.awt.color`, `java.awt.datatransfer`, `java.awt.dnd`, `java.awt.event`, `java.awt.font`, `java.awt.geom`, `BufferedReader`, `BufferedWriter`, `BufferOverflowException`, `BufferStrategy`, `BufferUnderflowException`, `Button`, `ButtonGroup`, `ButtonModel`, `ButtonUI`, `Byte`, `ByteArrayInputStream`, `ByteArrayOutputStream`, `ByteBuffer`, `ByteChannel`, `ByteHolder`, `ByteLookupTable`, and `ByteOrder`. The main content area shows the `Calendar` class page. The page title is "Overview Package Class Use Tree Deprecated Index Help". The class is `java.util.Calendar`, which extends `java.lang.Object` and implements `Serializable`, `Cloneable`, and `Comparable<Calendar>`. The page also lists "All Implemented Interfaces" and "Direct Known Subclasses". A note indicates that `Calendar` is an abstract class. The description states: "The `Calendar` class is an abstract class that provides methods for converting between a specific instant in time and a set of `calendar fields` such as `YEAR`, `MONTH`, `DAY_OF_MONTH`, `HOURL`, and so on, and for manipulating the calendar fields, such as getting the date of the next week. An instant in time can be represented by a millisecond value that is an offset from the *Epoch*, January 1, 1970 00:00:00 GMT (Gregorian)."

# 기초적인 자바 클래스들

## 03. 날짜와 시간에 관련된 클래스들

### GregorianCalendar 클래스

- GregorianCalendar 클래스 : 널리 사용되는 양력을 계산하는 클래스

The screenshot shows the Java 2 Platform Standard Ed. 5.0 API documentation for the `GregorianCalendar` class. The browser window title is "GregorianCalendar (Java 2 Platform SE 5.0) - Microsoft Internet Explorer". The address bar shows the URL "http://java.sun.com/j2se/1.5.0/docs/api/index.html". The left sidebar lists various Java packages and classes, including `java.applet`, `java.awt`, `java.awt.color`, `java.awt.datatransfer`, `java.awt.dnd`, `java.awt.event`, `java.awt.font`, and `java.awt.geom`. The main content area displays the class hierarchy: `java.util` package, `Calendar` class, and `GregorianCalendar` class. It also lists the implemented interfaces: `Serializable`, `Cloneable`, and `Comparable<Calendar>`. The class definition shows `public class GregorianCalendar` extending `Calendar`. A description states that `GregorianCalendar` is a concrete subclass of `Calendar` and provides the standard calendar system used by most of the world. It also mentions that `GregorianCalendar` is a hybrid calendar that supports both the Julian and Gregorian calendar systems with the support of a single discontinuity, which corresponds by default to the Gregorian date when the Gregorian calendar was instituted (October 15, 1582 in some countries, later in others). The cutover date may be changed by the caller by calling `setGregorianChange()`.

# 기초적인 자바 클래스들

## 03. 날짜와 시간에 관련된 클래스들

### 시스템 시계에 대하여

- 배터리로 작동 : 항상 현재 시각을 가리킴
- 다음 시각을 기점으로 하는 밀리세컨드 단위로 시각 표시

[시스템 클럭의 기점이 되는 시간]

1970년 1월 1일 00:00:00 GMT

GMT 2007년 1월 1일 0시    ➡    1167609600000

# 기초적인 자바 클래스들

## 03. 날짜와 시간에 관련된 클래스들

### GregorianCalendar 클래스

#### •• 사용 방법

- 1) 시스템 시계로부터 현재 시각 읽어오기

```
GregorianCalendar calendar = new GregorianCalendar();
```

- 2) GregorianCalendar 객체로부터 년월일시분초 얻어내기

```
int year = calendar.get(Calendar.YEAR);           // 연도를 리턴
int month= calendar.get(Calendar.MONTH);          // 월을 리턴
int date = calendar.get(Calendar.DATE);           // 일을 리턴
int amPm = calendar.get(Calendar.AM_PM);          // 오전/오후 구분을 리턴
int hour = calendar.get(Calendar.HOUR);           // 시를 리턴
int min  = calendar.get(Calendar.MINUTE);         // 분을 리턴
int sec  = calendar.get(Calendar.SECOND);         // 초를 리턴
```

# 기초적인 자바 클래스들

## 03. 날짜와 시간에 관련된 클래스들

### Calendar 클래스

• Calendar 클래스의 년월일시분초 관련 상수들

Field Summary

static int	<b>AM</b>	Value of the <code>AM_PM</code> field indicating the period of the day from midnight to just before noon.
static int	<b>AM_PM</b>	Field number for get and set indicating whether the HOUR is before or after noon.
static int	<b>APRIL</b>	Value of the <code>MONTH</code> field indicating the fourth month of the year.
protected boolean	<b>areFieldsSet</b>	True if <code>fields[]</code> are in sync with the currently set time.
static int	<b>AUGUST</b>	Value of the <code>MONTH</code> field indicating the eighth month of the year.
static int	<b>DATE</b>	Field number for get and set indicating the day of the month.
static int	<b>DAY_OF_MONTH</b>	Field number for get and set indicating the day of the month.
static int	<b>DAY_OF_WEEK</b>	Field number for get and set indicating the day of the week.

Calendar 클래스에는 get 메소드에 사용할 수 있는 이런 상수들이 선언되어 있습니다.

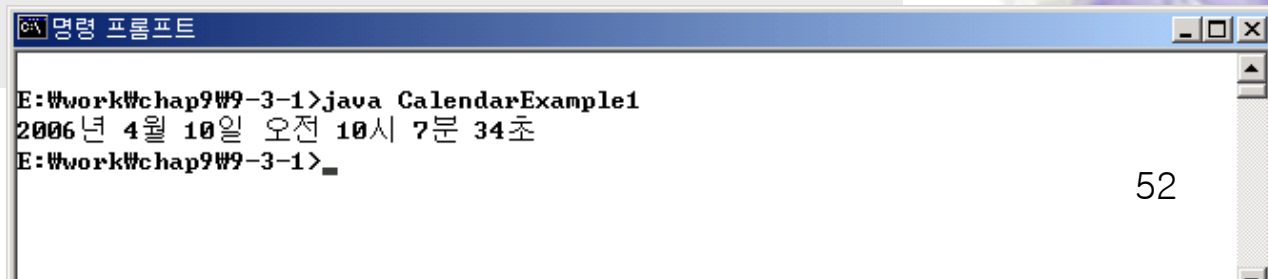
get 메소드의 리턴 값도 일부 상수로 선언되어 있습니다.

# 기초적인 자바 클래스들

## Calendar 클래스와 GregorianCalendar 클래스

- [예제 9-13] 현재의 날짜와 시간을 출력하는 프로그램

```
1  import java.util.GregorianCalendar;
2  import java.util.Calendar;
3  class CalendarExample1 {
4      public static void main(String args[]) {
5          GregorianCalendar calendar = new GregorianCalendar();
6          int year = calendar.get(Calendar.YEAR);
7          int month = calendar.get(Calendar.MONTH) + 1; ----- 월(月)은 0 ~ 11로 리턴되기 때문에 1을 더했습니다.
8          int date = calendar.get(Calendar.DATE);
9          int amPm = calendar.get(Calendar.AM_PM);
10         int hour = calendar.get(Calendar.HOUR);
11         int min = calendar.get(Calendar.MINUTE);
12         int sec = calendar.get(Calendar.SECOND);
13         String sAmPm = amPm == Calendar.AM ? "오전" : "오후";
14         System.out.printf("%d년 %d월 %d일 %s %d시 %d분 %d초",
15                             year, month, date, sAmPm, hour, min, sec); } ----- 연월일 시분초를
16                                     }                                     주어진 포맷으로 출력합니다.
17     }
```



```
E:\work\chap9\9-3-1>java CalendarExample1
2006년 4월 10일 오전 10시 7분 34초
E:\work\chap9\9-3-1>
```

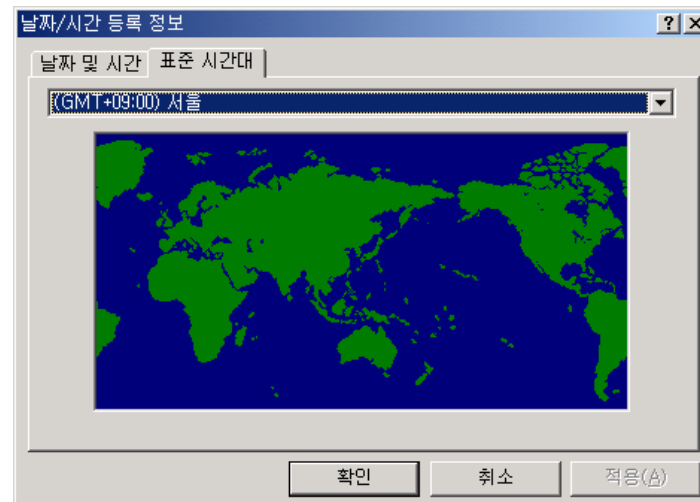


# 기초적인 자바 클래스들

## 03. 날짜와 시간에 관련된 클래스들

### 시간대(time zone)에 대하여

- 시간대 : 동일 시각을 사용하는 지구상의 구역
- 컴퓨터가 사용하는 시간대는 운영체제에서 설정할 수 있음
  - 윈도우즈 운영체제의 시간대를 설정하는 윈도우



# 기초적인 자바 클래스들

## 03. 날짜와 시간에 관련된 클래스들

### 시간대 설정하기

- GregorianCalendar 클래스의 setTimeZone 메소드 호출

```
calendar.setTimeZone(timeZone);
```

GregorianCalendar 객체

TimeZone 객체

# 기초적인 자바 클래스들

## 03. 날짜와 시간에 관련된 클래스들

### 시간대 설정하기

- TimeZone 객체를 얻는 방법

```
TimeZone timeZone = TimeZone.getTimeZone("Europe/London");
```

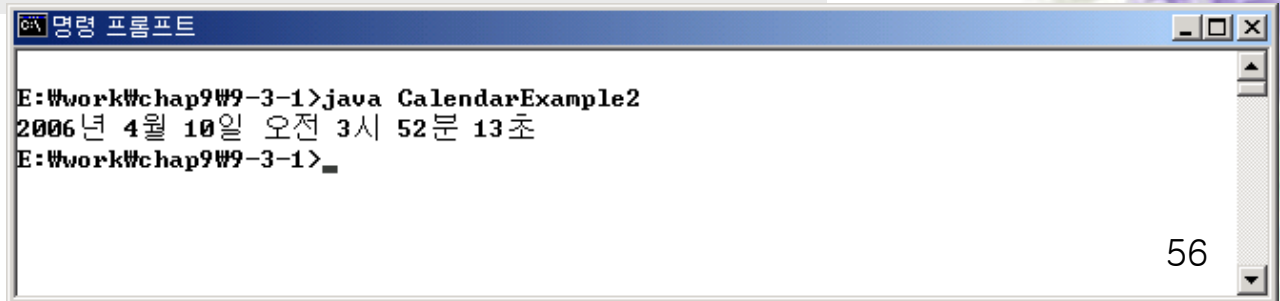
↑                      ↑  
이 문자열에 해당하는  
TimeZone 객체를 찾아서 리턴하는 메서드

# 기초적인 자바 클래스들

## ● 시간대 설정하기

- [예제 9-14] 런던의 현재 날짜와 시간을 출력하는 프로그램

```
1  import java.util.*;
2  class CalendarExample2 {
3      public static void main(String args[]) {
4          Calendar calendar = new GregorianCalendar();
5          TimeZone timeZone = TimeZone.getTimeZone("Europe/London");
6          calendar.setTimeZone(timeZone);
7          int year = calendar.get(Calendar.YEAR);
8          int month = calendar.get(Calendar.MONTH) + 1;
9          int date = calendar.get(Calendar.DATE);
10         int amPm = calendar.get(Calendar.AM_PM);
11         int hour = calendar.get(Calendar.HOUR);
12         int min = calendar.get(Calendar.MINUTE);
13         int sec = calendar.get(Calendar.SECOND);
14         String sAmPm = amPm == Calendar.AM ? "오전" : "오후";
15         System.out.printf("%d년 %d월 %d일 %s %d시 %d분 %d초",
16                             year, month, date, sAmPm, hour, min, sec);
17     }
18 }
```



```
명령 프롬프트
E:\work\chap9\9-3-1>java CalendarExample2
2006년 4월 10일 오전 3시 52분 13초
E:\work\chap9\9-3-1>
```

# 기초적인 자바 클래스들

## 03. 날짜와 시간에 관련된 클래스들

### 특정 날짜와 시간 표현하기

• GregorianCalendar 클래스의 생성자 파라미터를 이용

```
calendar1 = new GregorianCalendar(2007, 7, 1);  
calendar2 = new GregorianCalendar(2007, 7, 1, 14, 30);  
calendar3 = new GregorianCalendar(2007, 7, 1, 14, 30, 15);
```

파라미터로 주어진 시각 정보를 갖는  
GregorianCalendar 객체를 생성합니다.

# 기초적인 자바 클래스들

## 03. 날짜와 시간에 관련된 클래스들

### 날짜와 시간을 포맷하는 클래스

#### •• DateFormat 클래스

- 날짜와 시각의 포맷 기능을 제공하는 클래스
- 추상 클래스이므로 직접 사용은 불가능

#### •• SimpleDateFormat 클래스

- DateFormat 클래스의 서브클래스
- 추상 클래스가 아니므로 직접 사용 가능

# 기초적인 자바 클래스들

## 03. 날짜와 시간에 관련된 클래스들

### SimpleDateFormat 클래스

#### •• 사용 방법

- 1) 생성자 파라미터로 날짜와 시간의 포맷을 넘겨줍니다.

```
dateFormat = new SimpleDateFormat("yyyy년 MM월 dd일 aa hh시 mm분 ss초");
```

↑  
날짜와 시간의 포맷에 사용되는 문자열 패턴

패턴 문자	의미
y	년
M	월
d	일
D	일 (1 ~ 365)
E	요일
a	오전/오후

패턴 문자	의미
H	시 (0 ~ 23)
h	시 (1 ~ 12)
K	시 (0 ~ 11)
k	시 (1 ~ 24)
m	분
s	초
S	1/1000초

# 기초적인 자바 클래스들

## 03. 날짜와 시간에 관련된 클래스들

### SimpleDateFormat 클래스

•• 사용 방법

- 2) `format` 메소드를 호출합니다.

```
String str = dateFormat.format(date);
```

↑  
SimpleDateFormat 객체

↑  
날짜와 시간 정보를  
담고 있는 객체

```
Date date = calendar.getTime();
```

↑                      ↑  
이 객체가 가지고 있는 시각정보와  
똑같은 날짜와 시간을 갖는 Date 객체를 만들어서 리턴하는 메소드



# 기초적인 자바 클래스들

## 03. 날짜와 시간에 관련된 클래스들

### SimpleDateFormat 클래스

- [예제 9-15] 날짜와 시간을 포맷하는 프로그램

```
1  import java.util.*;
2  import java.text.*;    // SimpleDateFormat 클래스가 속하는 패키지
3  class DateFormatExample1 {
4      public static void main(String args[]) {
5          GregorianCalendar calendar = new GregorianCalendar();
6          SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy년 MM월 dd일 aa hh시 mm분 ss초");
7          String str = dateFormat.format(calendar.getTime());
8          System.out.println(str);
9      }
10 }
```

GregorianCalendar 객체를 Date 객체로  
만들어서 format 메소드를 호출합니다.



```
명령 프롬프트
E:\work\chap9\9-3-2>java DateFormatExample1
2006년 11월 06일 오후 04시 05분 23초
E:\work\chap9\9-3-2>
```

# 기초적인 자바 클래스들

## 03. 날짜와 시간에 관련된 클래스들

### SimpleDateFormat 클래스

- 날짜/시간을 특정 시간대로 포맷하는 방법

- format 메소드를 호출하기 전에 setTimeZone 메소드를 호출하면 됩니다.

```
dateFormat.setTimeZone(timeZone);
```

↑  
TimeZone 클래스의 객체

# 기초적인 자바 클래스들

## 03. 날짜와 시간에 관련된 클래스들

### SimpleDateFormat 클래스

- [예제 9-16] 날짜/시간을 특정 시간대로 포맷하는 예

```
1  import java.util.*;
2  import java.text.*;
3  class DateFormatExample2 {
4      public static void main(String args[]) {
5          GregorianCalendar calendar = new GregorianCalendar();
6          SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy년 MM월 dd일 aa hh시 mm분 ss초");
7          dateFormat.setTimeZone(TimeZone.getTimeZone("America/New_York"));
8          String str = dateFormat.format(calendar.getTime());
9          System.out.println(str);
10     }
11 }
```

---- 뉴욕에 해당하는 TimeZone 객체를 가지고 setTimeZone 메소드를 호출합니다.



```
명령 프롬프트
E:\work\chap9\9-3-2>java DateFormatExample2
2006년 11월 06일 오전 02시 05분 46초
E:\work\chap9\9-3-2>
```

## 기초적인 자바 클래스들

## 04. 수학적 계산에 사용되는 클래스들

- Math 클래스

- `Math` 클래스 : 수학에서 많이 사용되는 함수를 제공하는 클래스

The screenshot shows the Java™ 2 Platform Standard Ed. 5.0 API documentation for the `java.lang.Math` class. The left sidebar lists the package hierarchy, with `java.lang.Math` selected. The main content area is divided into two sections: "Field Summary" and "Method Summary".

**Field Summary:**

- `E`: The double value that is closer than any other to  $e$ , the base of the natural logarithms.
- `PI`: The double value that is closer than any other to  $\pi$ , the ratio of the circumference of a circle to its diameter.

**Method Summary:**

- `abs(double a)`: Returns the absolute value of a double value.
- `abs(float a)`: Returns the absolute value of a float value.
- `abs(int a)`: Returns the absolute value of an int value.
- `abs(long a)`: Returns the absolute value of a long value.
- `acos(double a)`: Returns the arc cosine of an angle, in the range of 0.0 through  $\pi$ .
- `asin(double a)`: Returns the arc sine of an angle, in the range of  $-\pi/2$  through  $\pi/2$ .
- `atan(double a)`: Returns the arc tangent of an angle, in the range of  $-\pi/2$  through  $\pi/2$ .
- `atan2(double y, double x)`: Converts rectangular coordinates ( $x$ ,  $y$ ) to polar ( $r$ ,  $\theta$ ).
- `cbrt(double a)`: Returns the cube root of a double value.
- `ceil(double a)`: Returns the smallest (closest to negative infinity) double value that is greater than or equal to the argument and is equal to a mathematical integer.
- `cos(double a)`: Returns the trigonometric cosine of an angle.
- `cosh(double a)`: Returns the hyperbolic cosine of a double value.
- `exp(double d)`: Returns Euler's number  $e$  raised to the power of a double value.
- `expm1(double x)`: Returns  $e^x - 1$ .
- `floor(double a)`: Returns the largest (closest to positive infinity) double value that is less than or equal to the argument and is equal to a mathematical integer.
- `hypot(double x, double y)`: Returns  $\sqrt{x^2 + y^2}$  without intermediate overflow or underflow.
- `IEEEremainder(double f1, double f2)`: Computes the remainder operation on two arguments as prescribed by the IEEE 754 standard.
- `log(double a)`: ...

# 기초적인 자바 클래스들

## 04. 수학적 계산에 사용되는 클래스들

### Math 클래스

•• Math 클래스가 제공하는 정적 메소드

메소드	설명
double pow(double a, double b)	a의 b승을 리턴
double sqrt(double a)	a의 제곱근을 리턴
double cbrt(double a)	a의 세제곱근을 리턴
double log(double a)	a의 자연로그를 리턴
double log10(double a)	a의 상용로그를 리턴
double sin(double a) double cos(double a) double tan(double a)	a의 sine, cosine, tangent 값을 리턴
double asin(double a) double acos(double a) double atan(double a)	a의 arc sine, arc cosine, arc tangent 값을 리턴
int abs(int a) long abs(long a) float abs(float a) double abs(double a)	a의 절대치를 리턴
int max(int a, int b) long max(long a, long b) float max(float a, float b) double max(double a, double b)	a와 b 중 큰 값을 리턴
int min(int a, int b) long min(long a, long b) float min(float a, float b) double min(double a, double b)	a와 b 중 작은 값을 리턴

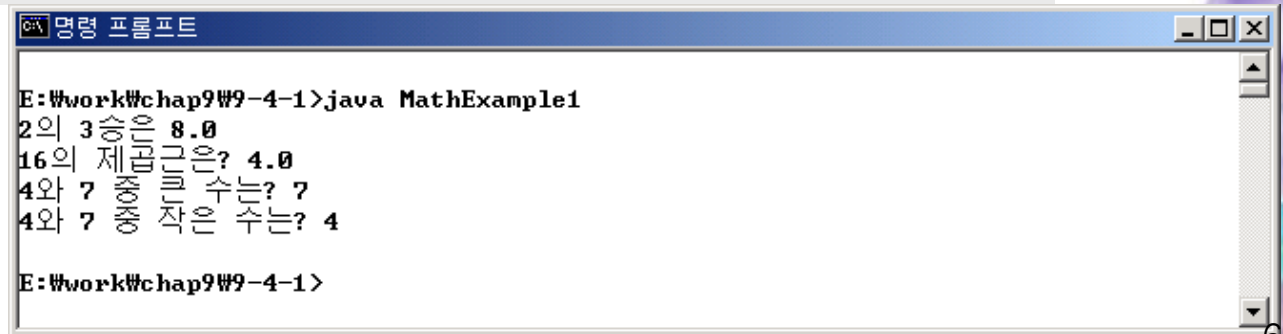
# 기초적인 자바 클래스들

## 04. 수학적 계산에 사용되는 클래스들

### Math 클래스

- [예제 9-17] Math 클래스의 메소드 사용 예를 보여주는 프로그램

```
1  class MathExample1 {  
2      public static void main(String args[]) {  
3          System.out.println("2의 3승은 " + Math.pow(2, 3));  
4          System.out.println("16의 제곱근은? " + Math.sqrt(16));  
5          System.out.println("4와 7 중 큰 수는? " + Math.max(4, 7));  
6          System.out.println("4와 7 중 작은 수는? " + Math.min(4, 7));  
7      }  
8  }
```



```
명령 프롬프트  
E:\work\chap9\9-4-1>java MathExample1  
2의 3승은 8.0  
16의 제곱근은? 4.0  
4와 7 중 큰 수는? 7  
4와 7 중 작은 수는? 4  
E:\work\chap9\9-4-1>
```

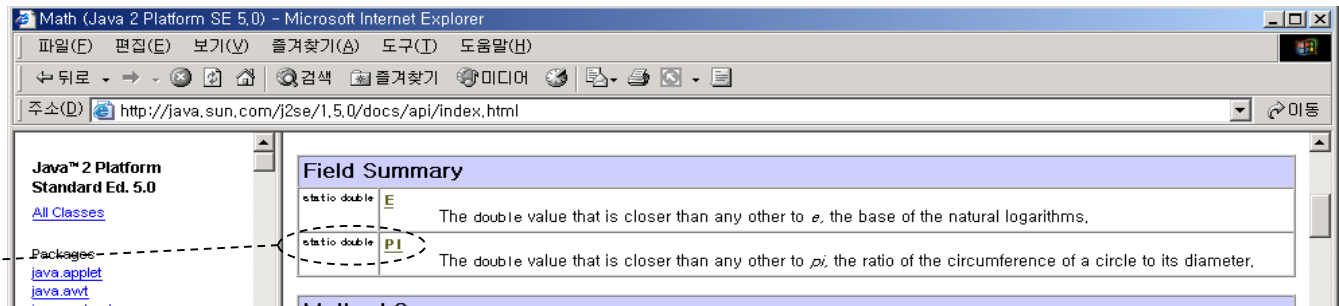
# 기초적인 자바 클래스들

## 04. 수학적 계산에 사용되는 클래스들

### Math 클래스

- 삼각함수 계산에 사용되는 상수

Math 클래스에 선언되어 있는  $\pi$  값을 표현하는 상수



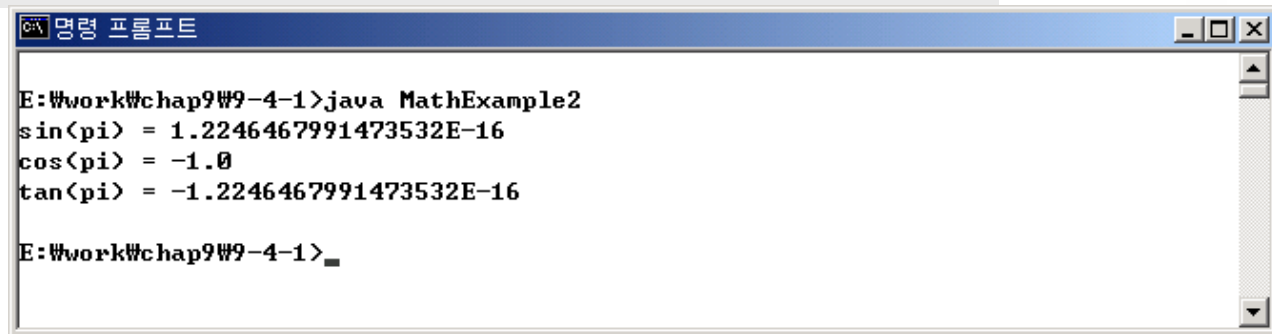
# 기초적인 자바 클래스들

## 04. 수학적 계산에 사용되는 클래스들

### Math 클래스

- [예제 9-18] Math 클래스의 삼각함수 메소드를 사용하는 프로그램

```
1  class MathExample2 {  
2      public static void main(String args[]) {  
3          System.out.println("sin(pi) = " + Math.sin(Math.PI));  
4          System.out.println("cos(pi) = " + Math.cos(Math.PI));  
5          System.out.println("tan(pi) = " + Math.tan(Math.PI));  
6      }  
7  }
```



```
명령 프롬프트  
E:\work\chap9\9-4-1>java MathExample2  
sin(pi) = 1.2246467991473532E-16  
cos(pi) = -1.0  
tan(pi) = -1.2246467991473532E-16  
E:\work\chap9\9-4-1>
```



# 기초적인 자바 클래스들

## 04. 수학적 계산에 사용되는 클래스들

### Math 클래스

- [예제 9-19] `Math.random` 메소드를 이용하여 난수를 발생하는 프로그램

```
1 class MathExample3 {  
2     public static void main(String args[]) {  
3         System.out.println(Math.random());  
4         System.out.println(Math.random());  
5         System.out.println(Math.random());  
6     }  
7 }
```

```
E:\work\chap9\9-3-1>java MathExample3  
0.32570713863110434  
0.3364767788648283  
0.6299921317758209  
  
E:\work\chap9\9-3-1>java MathExample3  
0.1464262312704253  
0.8398496462988356  
0.9723189857664793  
  
E:\work\chap9\9-3-1>java MathExample3  
0.5872845452546221  
0.7610581656478624  
0.7148175700969296  
  
E:\work\chap9\9-3-1>
```

프로그램을 실행할 때마다 다른 결과가 나옵니다.

# 기초적인 자바 클래스들

## 04. 수학적 계산에 사용되는 클래스들

### Math 클래스

- random 메소드의 활용 방법

```
int num = (int) (Math.random() * 10);
```

↑  
0부터 1까지의 소수 난수를  
0부터 9까지의 정수 난수로 만드는 식

# Random class

## ❖ Random

- Random Object는 일련의 난수를 생성한다. 이렇게 생성된 Random Object는 int type, float type 등의 난수가 발생 가능하며 정수형 난수 발생은 특정 범위가 없다. 하지만 부동소수점을 가지는 실수형들의 난수는 0.0에서 1.0사이의 값을 받는다.
- RandomObject를 생성할 수 있는 Constructor

### ※ Random 객체의 생성자

생성자	설명
Random( )	새로운 Random 객체를 생성하고 초기화한다.

# Random class

## ※ Random 객체에 자주 쓰이는 메서드

반환형	메서드명	설명
double	nextDouble( )	double형 자료에 따른 자료를 반환하게 되며 0.0~1.0 사이의 값이 반환
float	nextFloat( )	float형 자료에 따른 자료를 반환하게 되며 0.0~1.0 사이의 값이 반환
double	nextGaussian( )	평균 0.0, 표준 편차 1.0의 Gauss 분포의 double형의 난수를 반환
int	nextInt( )	int형의 범위 전체에서 난수를 발생하여 반환
	nextInt(int n)	0부터 인자로 전달된 값의 전까지를 범위로 하여 난수를 발생하여 반환
long	nextLong( )	long형의 범위 전체에서 난수를 발생하여 반환
void	setSeed(long seed)	인자로 전달된 long형인 seed를 난수 발생기의 시작 seed로 재설정

- 위의 Constructor와 method들을 이용하여 간단한 예제를 작성한다. 예제는 Array를 미리 준비해 두고 요소를 접근할 수 있는 index값을 난수를 발생하여 얻어낸다. 그렇게 되면 Array의 요소들 중 어떤 것이 선택이 될지 예측할 수 없으며 매번 실행할 때마다 선택되는 Array의 요소가 달라질 가능성이 매우 크므로 Lotto나 맞고 와 같은 게임에 응용이 되는 예제다.

# Random class

```
01 import java.util.Random;
02 import static java.lang.System.out;
03 public class RandomEx1 {
04
05     public static void main(String[] args) {
06         String[] lesson = {"Java Beg", "JSP", "XML&Java", "EJB",};
07         Random r1 = new Random();
08
09         int index = r1.nextInt(4);
10         out.println("선택과목 :"+lesson[index]);
11     }
12 }
```

- 9번 행에서 난수가 발생하며 이는 `nextInt(4)`로 인해 0부터 3까지의 정수 중 난수를 발생하여 `index` 변수에 대입한다, 이것으로 `nextInt(4)`에서 쓰이는 4라는 정수는 바로 범위를 의미,

# 기초적인 자바 클래스들

## 04. 수학적 계산에 사용되는 클래스들

### Random 클래스

- Random 클래스 : 난수 발생 기능만을 제공하는 클래스

- 사용 방법

- 1) Random 클래스의 객체를 생성합니다.

```
Random random = new Random();
```

↑  
파라미터를 받지 않는 생성자

```
Random random = new Random(51909091L);
```

↑  
long 타입의 파라미터를  
받는 생성자

- 2) 생성된 객체에 대해 난수 발생 메소드를 호출합니다.

# 기초적인 자바 클래스들

## 04. 수학적 계산에 사용되는 클래스들

### Random 클래스

• Random 클래스의 난수 발생 메소드

메소드	설명
int nextInt()	int 타입 난수를 리턴
long nextLong()	long 타입 난수를 리턴
float nextFloat()	0 이상 1 미만의 float 타입 난수를 리턴
double nextDouble()	0 이상 1 미만의 double 타입 난수를 리턴
boolean nextBoolean()	true와 false 중 한 값을 랜덤하게 리턴
int nextInt(int n)	0 이상 n 미만의 int 타입 난수를 리턴
void nextBytes(byte[] arr)	파라미터로 주어진 배열을 난수로 채움
double nextGaussian()	평균 0.0, 표준편차 1.0으로 정규분포된 난수를 리턴

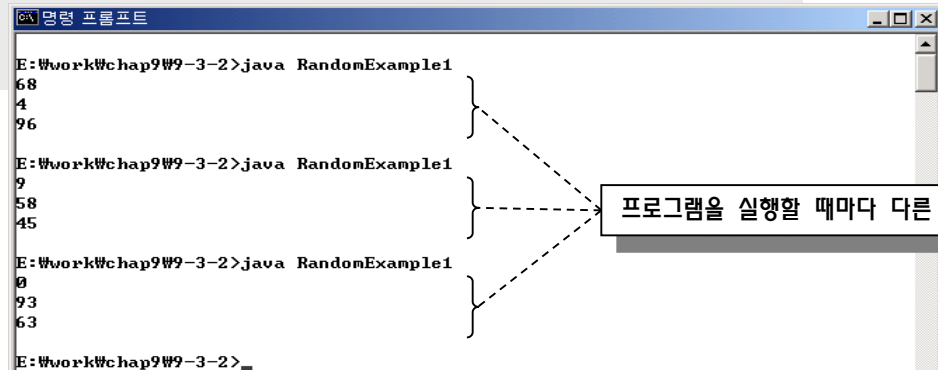
# 기초적인 자바 클래스들

## 04. 수학적 계산에 사용되는 클래스들

### Random 클래스

- [예제 9-20] Random 클래스를 이용하여 난수를 발생하는 프로그램 (1)

```
1  import java.util.*;
2  class RandomExample1 {
3      public static void main(String args[]) {
4          Random random = new Random();
5          System.out.println(random.nextInt(100));
6          System.out.println(random.nextInt(100));
7          System.out.println(random.nextInt(100));
8      }
9  }
```



The screenshot shows a Windows command prompt window titled "명령 프롬프트". It displays the execution of the Java program `RandomExample1` three times. Each execution produces three random integers between 0 and 99. The results are: 68, 4, 96; 9, 58, 45; and 0, 93, 63. Dashed lines connect the three groups of results to a text box on the right.

```
E:\work\chap9\9-3-2>java RandomExample1
68
4
96

E:\work\chap9\9-3-2>java RandomExample1
9
58
45

E:\work\chap9\9-3-2>java RandomExample1
0
93
63

E:\work\chap9\9-3-2>_
```

프로그램을 실행할 때마다 다른 결과가 나옵니다.



# 기초적인 자바 클래스들

## Random 클래스

- [예제 9-21] Random 클래스를 이용하여 난수를 발생하는 프로그램 (2)

```
1  import java.util.*;
2  class RandomExample2 {
3      public static void main(String args[]) {
4          Random random = new Random(349239L);
5          System.out.println(random.nextInt(100));
6          System.out.println(random.nextInt(100));
7          System.out.println(random.nextInt(100));
8      }
9  }
```

```
명령 프롬프트
E:\work\chap9\9-3-2>java RandomExample2
2
23
1
E:\work\chap9\9-3-2>java RandomExample2
2
23
1
E:\work\chap9\9-3-2>java RandomExample2
2
23
1
E:\work\chap9\9-3-2>_
```

몇번을 실행해도 같은 결과가 나옵니다.