



JAVA 프로그래밍

Chapter 12 기본 API 클래스

목차

- ❖ 자바 API 도큐먼트
- ❖ java.lang 패키지
- ❖ Object 클래스
- ❖ System 클래스
- ❖ Class 클래스
- ❖ String, StringBuffer, StringTokenizer 클래스
- ❖ Wrapper 클래스
- ❖ java.util 패키지
- ❖ Date, Calendar 클래스
- ❖ Format 클래스

자바 API 도큐먼트

- ❖ 자바 API (Application Programming Interface)
- ❖ API는 라이브러리(library) 라고 부르며, 프로그램 개발에 자주 사용되는 클래스 및 인터페이스의 모음을 말한다.
- ❖ `${JAVA_HOME}\jre\lib\rt.jar`라는 압축 파일

java.lang 패키지

- ❖ java.lang 패키지는 자바 프로그램의 기본적인 클래스들의 집합이다.
- ❖ java.lang 패키지의 클래스들은 import 없이 사용할 수 있다.

클래스		용도
Object		자바 클래스의 최상위 클래스
System		표준 입력 장치(키보드)로 부터 데이터를 입력. 표준 출력 장치(모니터)로 출력. 자바 가상 기계 종료. 현재 시스템 시간 조회. 시스템 프로퍼티 및 환경변수 조회.
Class		클래스의 정보를 담고 있는 클래스
String		문자열을 저장하고 문자열 관련 정보를 제공
StringBuffer, StringBuilder		문자열을 저장하고 내부 문자열 조작 기능을 제공
Math		수학 함수를 제공
Wrapper	Byte, Short, Character, Integer, Float, Double, Boolean, Long	기본 타입의 데이터를 갖는 객체를 만들 때 사용 문자열을 기본 타입으로 변환 입력값 검사에 사용

java.util 패키지

- ❖ java.util 패키지는 자바 프로그램의 Utility 클래스들의 집합이다.
- ❖ java.util 패키지는 컬렉션 클래스들이 주로 존재

클래스	용도
Arrays	배열을 조작(비교, 복사, 정렬, 찾기) 등의 기능 제공
Calendar	시스템의 날짜와 시간 정보를 제공
Date	날짜와 시간 정보를 저장하는 클래스
Objects	객체 비교, 널(null) 여부 등을 조사
StringTokenizer	특정 문자로 구분된 문자열을 추출하는 기능 제공
Random	난수를 생성

Object 클래스

❖ Object 클래스

- 자바의 모든 클래스는 직접적이든 간접적이든 Object 클래스를 상속받는다.
- 모든 클래스는 묵시적으로 자동 상속된다. (extends Object 하지 않아도 자동 상속)
- 모든 클래스의 최상위 클래스이기 때문에 참조값에 연결된 메모리와 직접적인 관련이 있는 메서드들로 이루어져 있다.

Object 클래스의 메서드	설 명
Object clone()	객체 자신의 복사본을 반환한다.
boolean equals(Object obj)	객체 자신과 객체 obj가 같은 객체인지 알려준다.
void finalize()	객체가 소멸될 때 가비지 컬렉터에 의해 자동으로 호출된다. 이때 수행되어야 하는 코드가 있을 때 오버라이딩 한다.
Class getClass()	객체 자신의 클래스 정보를 담고 있는 Class 인스턴스를 반환한다.
int hashCode()	객체 자신의 해시코드를 반환한다.
String toString()	객체 자신의 정보를 문자열로 반환한다.
void notify()	객체 자신을 사용하려고 기다리는 스레드를 하나만 깨운다.
void notifyAll()	객체 자신을 사용하려고 기다리는 모든 스레드를 깨운다.
void wait()	다른 스레드가 notify(), notifyAll()을 호출할 때까지 현재 스레드를 무한히 또는 지정된 시간 동안 기다리게 한다.

System 클래스

❖ System 클래스

- System 클래스를 이용하면 운영체제의 일부 기능을 이용할 수 있다.
- 프로그램 종료, 키보드 입력, 모니터 출력, 메모리 정리, 현재 시간 읽기, 시스템 프로퍼티 읽기, 환경 변수 읽기 등이 가능하다.
- 모든 필드와 메서드는 정적(static) 필드와 메서드로 구성되어 있다.

메서드	설명
exit()	프로세스 강제 종료. System.exit(0);
gc()	쓰레기 수집기 실행. 메모리 정리
currentTimeMillis()	시스템의 현재 시각을 밀리세컨드(1/1000초) 단위로 읽기.
nanoTime()	시스템의 현재 시각을 나노세컨드(1/10 ⁹ 초) 단위로 읽기.
getProperty(String key)	시스템 프로퍼티 읽기
getenv(String name)	환경 변수 읽기

Class 클래스

❖ Class 클래스

- 클래스의 등록정보 클래스
- 바이트 코드의 등록정보 클래스

❖ 리플렉션(Reflection)이란

- 객체를 통해 클래스의 정보를 분석해 내는 프로그램 기법
- 클래스의 생성자, 필드, 메서드 정보를 알아낼 수 있다.
- `getMethods()` : 클래스 내에 포함된 메서드 알아내기
- `getFields()` : 클래스 내에 포함된 필드 알아내기
- `getConstructors()` : 클래스 내에 존재하는 생성자 알아내기
- `getInterfaces()` : 클래스 내에 존재하는 인터페이스 알아내기
- `getSuperclass()` : 클래스 내에 존재하는 상위클래스 알아내기

Class 클래스

❖ Class 클래스를 얻는 방법

- `class Data {}`
- `Class c = Data.class or obj.getClass()`

❖ Class c로부터 클래스의 이름 얻기

- `String name = c.getName();`

❖ 등록정보(Class) 클래스로 할 수 있는 일

- 가상머신이 `Data.class`를 로딩한 뒤에 할 수 있는 대부분의 작업을 프로그래머가 직접 할 수 있다.
- `Data.class`를 로딩한 뒤에 가상머신이 하는 일을 `Data` 클래스의 등록정보 클래스를 이용해서 할 수 있다.

❖ Class 클래스를 이용한 객체 생성

- `Class c = Data.class;`
- `Object obj = c.newInstance();`

String 클래스

- ❖ 자바의 문자열은 String클래스의 인스턴스로 관리된다.
- ❖ 문자열 리터럴은 String 객체로 자동 생성되지만, 다양한 생성자를 통해서 직접 생성도 가능하다.

메소드		기능
int	length()	문자열의 길이를 리턴
char	charAt(int index)	주어진 index위치의 문자를 리턴
boolean	equals(Object anObject)	문자열의 내용 비교, 같으면 true, 다를 때 false
String	substring(int beginIndex, int endIndex)	beginIndex부터 endIndex-1까지의 문자열을 리턴
String	concat(String 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

StringBuffer & StringBuilder 클래스

- ❖ String 클래스의 인스턴스는 그 값을 변경할 수 없는 불변객체(immutable object)
- ❖ StringBuffer 클래스는 버퍼(buffer) 공간을 가지고 문자열을 변경, 조작할 수 있는 기능을 제공한다.
- ❖ StringBuilder
 - StringBuffer와 동일한 기능을 제공하며, 단지 StringBuilder는 동기화를 제공하지 않는다.
 - Java 1.5 추가

메소드		기능
StringBuffer	append(String str)	문자열 뒤에 str을 덧붙임
StringBuffer	insert(int offset, String str)	문자열의 offset 위치에 str을 삽입
StringBuffer	delete(int start, int end)	Start부터 end - 1까지의 부분 문자열을 삭제
StringBuffer	deleteCharAt(int index)	Index 위치에 있는 하나의 문자를 삭제
int	capacity()	버퍼 크기를 리턴
void	ensureCapacity(int miniumCapacity)	버퍼 크기를 늘리는 기능
void	trimToSize()	버퍼 크기를 줄이는 기능

StringTokenizer 클래스

- ❖ 구분자를 이용해 문자열로부터 토큰을 추출하는 기능을 제공
 - - 토큰(token) : 문자열을 구성하는 단위
- ❖ `StringTokenizer st = new StringTokenizer("문자열", "구분자");`

메소드		기능
int	<code>countTokens()</code>	꺼내지 않고 남아 있는 토큰의 수
boolean	<code>hasMoreTokens()</code>	남아 있는 토큰이 있는지 여부
String	<code>nextToken()</code>	토큰을 하나씩 꺼내옴

Wrapper 클래스

- ❖ Wrapper 클래스에 대하여
- ❖ Wrapper 클래스의 기본적인 사용 방법
- ❖ Wrapper 클래스의 정적 메소드와 상수
- ❖ 자동 Boxing과 자동 Unboxing(java 1.5이후지원)

날짜와 시간에 관련된 클래스들

❖ Date 클래스

- Date 클래스는 날짜를 표현하는 클래스.
- 날짜 정보를 주고 받을 때 주로 사용.

❖ Calendar와 GregorianCalendar 클래스

- 현재 시스템의 날짜와 시간을 관리하는 클래스
- java.util 패키지
- Calendar : 추상클래스
- GregorianCalendar : Calendar 클래스의 서브 클래스

날짜와 시간에 관련된 클래스들

❖ Calendar 클래스의 주요 메서드

반환형	메서드	설명
boolean	after(Object when)	현재 Calendar 객체가 인자로 전달된 when 객체의 날짜보다 후의 시각이라면 true, 그렇지 않으면 false를 반환한다.
boolean	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)	현재 Calendar 객체의 필드 중 년도와 월, 그리고 일(DAY_OF_MONTH)을 인자로 전달된 값을 설정한다.
void	setTimeInMillis(long mills)	Calendar의 현재 시각을 인자로 전달된 long형의 값으로 설정한다.

날짜와 시간에 관련된 클래스들

❖ Calendar 클래스의 주요 상수들

상수	설명
AM_PM	HOUR가 정오보다 전인지, 후인지를 나타내주는 상수이며, 만약 전일 경우 (AM)에는 0을 반환하고 정오이후일 경우(PM)는 1을 반환한다.
DAY_OF_MONTH	한달 중의 날짜 수를 의미한다. 상수(1 ~ 31)
DAY_OF_WEEK	SUNDAY에서 STAURDAY 까지의 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 주째인지 나타내는 상수

Format 클래스

❖ 날짜 형식 클래스(SimpleDateFormat)

- 날짜와 시간을 포맷하는 기능을 가진 클래스
- `SimpleDateFormat sdf = new SimpleDateFormat("yyyy년 MM월 dd일 ");`
- `String result = 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 몇 번째 주	S	밀리세컨드(1/1000초)
W	월의 몇 번째 주		

Format 클래스

❖ 숫자 형식 클래스(DecimalFormat)

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

기호	의미	패턴 예	1234567.89 변환결과
0	10 진수(빈자리는 0으로 채움)	0 0.0 0000000000.00000	1234568 1234567.9 0001234567.89000
#	10 진수(빈자리는 채우지 않음)	# #.# #####.#####	1234568 1234567.9 1234567.89
.	소수점	#.0	1234567.9
-	음수 기호	+#.0 -#.0	+1234567.9 -1234567.9
,	단위 구분	#,###.0	1,234,567.9
E	지수 문자	0.0E0	1.2E6
;	양수와 음수의 패턴을 모두 기술할 경우, 패턴 구분자	+#,###;-#,###	+1,234,568(양수일 때) -1,234,568(음수일 때)
%	%문자	## %	123456789 %
₩u00A4	통화 기호	₩u00A4 #,###	₩ 1,234,568

Format 클래스

❖ 문자열 형식 클래스(MessageFormat)

- 문자열에 데이터가 들어갈 자리를 표시해 두고, 프로그램이 실행하면서 동적으로 데이터를 삽입해 문자열을 완성시킬 수 있다.

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

정규 표현식과 Pattern 클래스

❖ 문자열을 정규 표현식으로 검증하는 기능.

- `java.util.regex.Pattern`
- `Pattern.matches("정규식", "검증할 문자열");`

기호	설명		
[]	한 개의 문자	[abc]	a, b, c 중 하나의 문자
		[^abc]	a, b, c 이외의 하나의 문자
		[a-zA-Z]	a~z, A~Z 중 하나의 문자
\d	한 개의 숫자, [0-9]와 동일		
\s	공백		
\w	한 개의 알파벳 또는 한 개의 숫자. [a-zA-Z_0-9]		
?	없음 또는 한 개		
*	없음 또는 한 개 이상		
+	한 개 이상		
{n}	정확히 n개		
{n,}	최소한 n개		
{n, m}	n개에서부터 m개까지		
()	그룹핑		