

{ JAVA }

선린인터넷  
웹 운영 과

Calendar 클래스

clone( ) 메소드

compareTo( ) 메소드

# Calendar 클래스



## ❖ Calendar클래스

- java.util 패키지에 있는 추상 클래스
- 개발자가 프로그램 실행 동안 기억하고 싶은 날짜,시간 정보를 관리하기 위하여 Calendar 객체를 이용함

## ❖ Calendar클래스의 get( ), set( )에 사용되는 static 상수들...

필드	의미	필드	의미
YEAR	년도	DAY	한 달의 <b>날짜</b>
MONTH	달	DAY_OF_WEEK	한 주의 <b>요일</b>
HOURL	0~11시로 표현한 <b>시간</b>	MINUTE	분
HOURL_OF_DAY	24시간 기준의 <b>시간</b>	SECOND	초

## ❖ Calendar객체 생성 및 이용

### ❖ Calendar 객체 생성 (얻기?)

- new Calendar( ) 코드를 사용하지 않고, **getInstance( ) 메소드**로 생성
- 사용예시 : `Calendar 참조변수 = Calendar . getInstance( );`
- getInstance( )가 리턴한 객체는 현재 날짜와 시간 정보를 가진다.
  - 이 객체를 이용하여 현재의 년도, 달, 시간 등 정보를 알아낼 수 있다.

### ❖ Calendar객체에서 날짜, 시간 정보 얻어오기

- **get( ) 메소드**에 **상수필드**를 이용한다.
- 사용예시 : `int year = 참조변수 . get( Calendar . YEAR );`

## ❖ Calendar객체 생성 및 이용

### ❖ Calendar객체에 날짜, 시간 정보 **설정하기**

- **set( )** 메소드를 이용한다.
- 사용예시

```
Calendar 참조변수 = Calendar . getInstance( ); // 객체 얻기  
참조변수 . clear( ); // 객체의 현재 날짜, 시간 정보 삭제  
참조변수 . set( 2017 , 10, 23 ) ; // 2017년 11월 23일로 설정  
참조변수 . set( Calendar . MINUTE , 30 ) ; // 30분으로 설정
```

# Calendar클래스

## (예제1) 실습예제 : 날짜 및 시간 정보 얻어오기

```
1 import java.util.Calendar;
2 public class CalendarExam1 {
3     public static void main(String[] args) {
4         // Calendar 클래스: java.util패키지에 있는 추상클래스
5         // - 개발자가 프로그램 실행 동안 기억하고 싶은 날짜와 시간 정보를
6         //   관리하기 위하여 Calendar 객체를 이용함.
7         // - 단, 컴퓨터 현재 시간을 알아낼 수는 있지만,
8         //   컴퓨터 현재 시간을 바꾸지는 못 한다!!!
9
10        // 1 컴퓨터 현재 날짜와 시간 정보를 알아내기 : new 연산자 안 됨!!
11        Calendar now = Calendar.getInstance(); // 객체 생성
12
13        int year = now.get(Calendar.YEAR); // 현재 년도
14        // Calendar의 month는 0부터 시작함...
15        int month = now.get(Calendar.MONTH) +1 ; // 현재 달
16        // System.out.println(Calendar.JANUARY); // 0
17        int day = now.get(Calendar.DAY_OF_MONTH); // 현재 날
18
19        System.out.printf("%d-%d-%d",year,month,day);
20    }
21 }
```

## (예제2) 실습예제 : 날짜 및 시간 정보 설정하기

```
1 import java.util.Calendar;
2 public class CalendarExam2 {
3     public static void main(String[] args) {
4         // 2 개발자가 원하는 날짜와 시간정보로 설정하기
5         Calendar now = Calendar.getInstance(); // 객체 생성
6
7         now.clear(); // 현재 기억하고 있는 날짜와 시간 정보를 지우기
8         now.set(2017, 2, 2); // 2017.3.2. 단, 3월은 2로 설정!!
9         // Calendar의 month는 0부터 시작함 πππ
10        now.set(Calendar.HOUR_OF_DAY, 16); // 오후4시 설정
11        now.set(Calendar.MINUTE, 30); // 30분으로 설정
12
13        int year = now.get(Calendar.YEAR); // 년도
14        int month = now.get(Calendar.MONTH) + 1; // 달
15        int day = now.get(Calendar.DAY_OF_MONTH); // 날
16        int hour = now.get(Calendar.HOUR_OF_DAY); // 시
17        int time = now.get(Calendar.MINUTE); // 분
18
19        System.out.printf("%d-%d-%d", year, month, day);
20        System.out.printf(" %d:%d\n", hour, time);
21    }
22 }
```

Console

&lt;terminated&gt; CalendarExam2 [Ja

2017-3-2 16:30

# clone( ) 메소드

- 현재 객체를 그대로 복제하고 싶다면?





## ❖ clone( ) 메소드

- java.lang.**Object** 클래스에 속한 메소드
- 현 객체와 똑같은 객체를 만들어 리턴하는 메소드
- "Cloneable 인터페이스를 구현한 클래스"가 생성한 객체만 복제 가능.

### clone

```
protected Object clone()  
    throws CloneNotSupportedException
```

Creates and returns a copy of this object. The precise meaning of "copy" may depend on the class of the object. The general intent is that, for any object x, the expression:

### public interface Cloneable

A class implements the Cloneable interface to indicate to the `Object.clone()` method that it is legal for that method to make a field-for-field copy of instances of that class.

Invoking Object's clone method on an instance that does not implement the Cloneable interface results in the exception `CloneNotSupportedException` being thrown.


## clone() 메소드

## (예제3) 실습예제 : Calendar 클래스를 활용한 객체 복제

```

1 import java.util.Calendar;
2 // Calendar 객체를 이용한 clone() 메소드 활용 예제
3 // clone() : 현 객체와 똑같은 객체를 만들어 리턴 = 깊은(deep) 복사
4 // - "Cloneable" 인터페이스를 구현한 클래스가 생성한 객체만 가능함!
5 // - (참고) Calendar 클래스는 미리 구현된 클래스로 정의되어 있음..
6 public class CloneExam1 {
7     public static void main(String[] args) {
8         // Calendar 객체 cal1 생성
9         Calendar cal1 = Calendar.getInstance();
10        cal1.set(2017, 2, 2); // 2017-3-2
11        int y1 = cal1.get(Calendar.YEAR);
12        int m1 = cal1.get(Calendar.MONTH)+1;
13        int d1 = cal1.get(Calendar.DAY_OF_MONTH);
14        System.out.printf("cal1 : %d-%d-%d\n",y1,m1,d1);
15
16        // cal1 객체를 복제한 cal2 생성 : 상호 별도 객체로 존재함...
17        Calendar cal2 = (Calendar) cal1.clone();
18        System.out.println("**cal1의 클론객체인 cal2 생성**");
19
20        int y2 = cal1.get(Calendar.YEAR);
21        int m2 = cal1.get(Calendar.MONTH)+1;
22        int d2 = cal1.get(Calendar.DAY_OF_MONTH);
23        System.out.printf("cal1 : %d-%d-%d\n",y1,m1,d1);
24        System.out.printf("cal2 : %d-%d-%d",y2,m2,d2);
25    }
26 }

```

 Console

&lt;terminated&gt; CloneExam1 [Java Application] C:\WjavaW\

cal1 : 2017-3-2

\*\*cal1의 클론객체인 cal2 생성\*\*

cal1 : 2017-3-2

cal2 : 2017-3-2

# clone() 메소드

## (예제4) 실습예제 : Cloneable 인터페이스를 구현한 A 클래스를 활용한 객체 복제

```
1 // 사용자 정의 class A 객체를 이용한 clone() 메소드 활용 예제
2 class A implements Cloneable{
3     int a;
4     @Override // clone + ctrl키 + spacebar키 : 자동완성 이용하세요!
5     protected Object clone() throws CloneNotSupportedException {
6         return super.clone();
7     }
8 }
9 public class CloneExam2 {
10    public static void main(String[] args) {
11        A a1 = new A();
12        a1.a = 1;
13        A a2 = null; // clone된 객체를 연결하기 위한 참조 변수를 미리 선언
14
15        try {
16            a2 = (A) a1.clone(); // a1 객체를 복제 -> a2 객체 생성
17        } catch (CloneNotSupportedException e) {
18            e.printStackTrace();
19        }
20
21        System.out.println("a1 : "+a1.a);
22        System.out.println("a2 : "+a2.a); // 내용이 a1과 같죠?
23    }
24 }
```

Console

&lt;terminated&gt; Clo

a1 : 1

a2 : 1

## clone() 메소드

(예제5) 실습예제 : Student 클래스의 객체를 복제하기 - 프로그램을 완성해봅시다!

```
1 class Student {
2     private String name;
3     public Student(String name) {
4         this.name = name;
5     }
6     public String getName() {
7         return name;
8     }
9 }
10 public class CloneExam3 {
11     public static void main(String[] args) {
12         Student s1 = new Student("홍길동");
13         Student s2 = null;
14
15         s2 = s1.clone(); // s1 객체를 복제하여 s2 객체를 생성하세요!
16
17         System.out.println("학생1의 이름: " + s1.getName());
18         System.out.println("학생1의 클론인 학생2의 이름: " + s2.getName());
19     }
20 }
```

CloneExam3.java

Console

<terminated> CloneExam3 [Java Application] C:\wjava\w

학생1의 이름: 홍길동

학생1의 클론인 학생2의 이름: 홍길동

# compareTo( ) 메소드

C언어 문자열 비교 : strcmp 함수 - strcmp() 함수 기억나시죠?

문자열을 비교하는 함수 strcmp 가 있다.  
헤더는 include<string.h>

- strcmp(문자열 A, 문자열 B) 는 A와 B를 비교하는 것이다.

A>B 일때 양수

A=B 일때 '0'

A<B 일때 음수로 나온다.

이때 비교는 아스키 코드값으로 비교한다.

만약 char A[4]: "abc" , char B[4]: "def" 면 A<B 이다.

# compareTo()메소드

## (예제6) 실습예제

```
1 // compareTo() 메소드 : 2개 문자열을 비교하여 int결과값을 반환
2 // 현재 문자열 객체와 인자로 주어진 문자열을 사전 순서를 기반으로 비교
3 // int결과값: A==B이면 0, A>B이면 양수, A<B이면 음수값을 반환
4 // -"Comparable<T>인터페이스를 구현한 클래스"가 생성한 객체만 가능함!
5 public class CompareToExam1 {
6     public static void main(String[] args) {
7         String s1 = "abc";
8         String s2 = "abc";
9         String s3 = "bcd";
10        String s4 = "abc";
11
12        System.out.println(s1.compareTo(s2));
13        // 출력결과=?
14
15        System.out.println(s1.compareTo(s3));
16        // 출력결과=?
17
18        System.out.println(s1.compareTo(s4));
19        // 출력결과=?
20    }
21 }
```

예외처리

계속...

