

제 14장 Java 표준 클래스

```
1 public class Main {  
2     public static void main(String[] args) {  
3         long start = System.currentTimeMillis();  
4         // 시간이 걸리는 처리  
5  
6         long end = System.currentTimeMillis();  
7         System.out.println("처리 시간 : " + (end - start) + "ms");  
8     }  
9 }
```

```
1  import java.util.Date;
2
3  public class Main {
4      public static void main(String[] args) {
5          Date now = new Date();
6          System.out.println(now);
7          System.out.println(now.getTime());
8          Date past = new Date(131662225935L);
9          System.out.println(past);
10     }
11 }
```

java.sql.Date 와 혼동하지 말 것

```
1 import java.util.Calendar;
2 import java.util.Date;
3
4 public class Main {
5     public static void main(String[] args) {
6         // 현재의 년도 표시
7         Date now = new Date();
8         Calendar calendar = Calendar.getInstance();
9         calendar.setTime(now);
10
11         int year = calendar.get(Calendar.YEAR);
12         System.out.println("올 해는 " + year + "년 입니다");
13         // 지정한 날의 Date형의 값을 얻기
14         calendar.set(2010, 8, 22, 1, 23, 45);
15         calendar.set(Calendar.YEAR, 2011);
16         Date past = calendar.getTime();
17     }
18 }
```

<https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html>

```
1  import java.text.ParseException;
2  import java.text.SimpleDateFormat;
3  import java.util.Date;
4
5  public class Main {
6      public static void main(String[] args) throws ParseException {
7          // 오늘의 날씨
8          Date now = new Date();
9          SimpleDateFormat format = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss");
10
11         String s = format.format(now);
12         System.out.println(s);
13
14         // 지정일시의 문자열을 파싱하여 Date형으로 변환
15         Date d = format.parse("2011/09/22 01:23:45");
16     }
17 }
```

```
1 public class Main {  
2     static class Empty {    // extends Object 가 생략 됨  
3  
4     }  
5  
6     public static void main(String[] args) {  
7         Empty e = new Empty();  
8         String s = e.toString();  
9         System.out.println(s); // s.toString() 이 호출 됨  
10    }  
11 }
```

```
1 public class Main {  
2     public static void main(String[] args) {  
3         // 다형성  
4         Object o1 = new Hero();  
5         Object o2 = new Wizard();  
6         Object o3 = "안녕하세요";  
7     }  
8 }
```



```
1 public class Main {  
2     public static void main(String[] args) {  
3         // 다형성  
4         Object o1 = new Hero();  
5         Object o2 = new Wizard();  
6         Object o3 = "안녕하세요";  
7  
8         printAnything(o2);  
9     }  
10  
11     private static void printAnything(Object object) {  
12         // 어떤 타입도 좋으니, 인수로 1개를 받아서 화면에 표시  
13         System.out.println(object.toString());  
14     }  
15 }
```



```
public void println(Object x) {  
    String s = String.valueOf(x);  
    synchronized (this) {  
        print(s);  
        newLine();  
    }  
}
```

```
1 public class Main {  
2     public static void main(String[] args) {  
3         Hero hero = new Hero();  
4         hero.name = "켈타스";  
5         hero.hp = 100;  
6         // hero.toString() 대신 hero 만 넣어도 됨  
7         System.out.println(hero.toString());  
8     }  
9 }
```

```
1 public class Hero extends Character {  
2  
3     @Override  
4     public void attack(Monster monster) {  
5         System.out.println(this.name + "의 공격!");  
6         System.out.println("적에게 10포인트의 데미지를 주었다!");  
7         monster.hp -= 10;  
8     }  
9  
10    @Override  
11    public String toString() {  
12        // toString() 을 오버라이드  
13        return "이름 : " + this.name + "/HP : " + this.hp;  
14    }  
15 }
```

```
1 public class Main {  
2     public static void main(String[] args) {  
3         Hero h1 = new Hero();  
4         h1.name = "켈타스";  
5         h1.hp = 100;  
6  
7         Hero h2 = new Hero();  
8         h2.name = "켈타스";  
9         h2.hp = 100;  
10  
11         if (h1.equals(h2) == true) {  
12             System.out.println("같은 내용입니다");  
13         } else {  
14             System.out.println("다른 내용입니다");  
15         }  
16     }  
17 }
```

```
1 public class Hero extends Character {  
2     @Override  
3     public boolean equals(Object obj) {  
4         if (this == obj) {  
5             return true;  
6         }  
7         if (obj instanceof Hero) {  
8             Hero hero = (Hero) obj;  
9             if (this.name.equals(hero.name)) {  
10                return true;  
11            }  
12        }  
13        return false;  
14    }
```

| 기본 타입 (primitive type) | 래퍼 타입 (wrapper class) |
|------------------------|-----------------------|
| byte | java.lang.Byte |
| short | java.lang.Short |
| int | java.lang.Integer |
| long | java.lang.Long |
| float | java.lang.Float |
| double | java.lang.Double |
| char | java.lang.Character |
| boolean | java.lang.Boolean |

```
1 public class Main {  
2     public static void main(String[] args) {  
3         int i1 = 15;  
4         Integer i2 = Integer.valueOf(i1);  
5         int i3 = i2.intValue();  
6     }  
7 }
```



```
1 public class Main {  
2     public static void main(String[] args) {  
3         int i1 = 15;  
4         Integer i2 = i1;  
5         int i3 = i2;  
6     }  
7 }
```

날짜의 처리

- Java에서 날짜 정보는 기본적으로 `java.util.Date` 를 사용
- 필요에 의해 `long` 형, 6개의 `int`, `String` 형으로 변환하여 사용
- '년 월 일 시 분 초' 의 6개의 `int` 값으로 부터 `Date` 인스턴스를 얻기 위해 `Calendar` 클래스를 사용
- `Date` 인스턴스의 내용을 임의의 형식으로 문자열 표시하기 위해서는 `SimpleDateFormat` 클래스를 사용

Object 클래스

- Java 에서, 모든 클래스는 `Object` 클래스의 자식이다
- 모든 인스턴스는 `Object` 형 변수에 대입할 수 있다
- 모든 클래스는 `Object` 로부터 `toString()` 이나 `equals()` 를 상속 받고 있다
- 자신이 작성한 클래스를 이용해, 문자열 표현이나 같은지 아닌지 비교를 하기 위해서는, `toString()` 이나 `equals()`를 오버라이드 해야한다

래퍼 클래스 (wrapper class)

- 기본 데이터형에 대응하는 래퍼 클래스가 `java.lang` 패키지에 존재 한다
- 기본 데이터형과 래퍼 클래스의 데이터는 `valueOf()` 나 `~value()` 메소드로 명시적으로 변환 가능
- 기본타입과 래퍼 타입은 서로 `AutoBoxing` / `AutoUnboxing` 기능에 의해 암묵적으로 변환 된다.

연습문제 14-1

main() 메소드만 가지는 클래스 Main 을 정의하고, 이하의 순서로 **현재의 100일 후의 날짜** 를 “서기 **2011년 09월 24일**” 와 같은 형식으로 표시하시오.

1. 현재의 날짜를 Date 형으로 얻는다
2. 얻은 날짜정보를 Calendar 에 설정한다
3. Calendar 에서 **일(day)** 값을 얻는다.
4. 얻은 값에 100을 더한 값을 Calendar 의 **일** 에 설정한다
5. Calendar 의 날짜정보를 Date 형으로 변환한다
6. SimpleDateFormat 을 이용하여 Date 인스턴스의 내용을 표시한다

연습문제 14-2

계좌번호를 표시하는 **String** 형 필드 **accountNumber** 와 잔액을 표시하는 **int** 형 필드 **balance** 를 가지는 은행계좌 클래스를 작성하시오. 그리고, 이 클래스에 메소드 선언을 추가해, 다음의 1.과 2.의 조건을 만족하도록 수정하시오.

1. 계좌번호 4649, 잔액 1592원 **Account** 인스턴스를 변수 **a** 에 생성하고 **System.out.println(a);** 를 실행하면, 화면에 **1592원 (계좌번호=4649)** 라고 표시
2. 계좌번호가 같으면 값이 같은 것으로 판단할 것. 단, “ 4649” 처럼 계좌번호의 앞에 공백이 있으면, 공백을 무시하고 비교할 것 (“4649”와 “ 4649 ”는 같은 것으로 본다)
(힌트 : `java.lang.String` 클래스의 `trim()` 메소드를 이용)