

PART 13

13. Tarih & Zaman

Java, java.util paketi içinde mevcut olan Date sınıfını sağlar, bu sınıf güncel tarih ve zamanı kapsüllemektedir.

Date sınıfı 2 adet constructor destekler. İlk constructor, güncel tarih ve zaman ile nesneye ilk değerini atar.

```
Date ( )
```

Aşağıdaki constructor, 1 Ocak 1970 gece yarısından bu yana geçen milisaniye sayısına eşit bir bağımsız değişken kabul etmektedir.

```
Date(long millisec)
```

Bir kere mevcut Date nesnesine sahipseniz, tarihlerle oynamak için aşağıda desteklenen metotlardan herhangi birini çağırabilirsiniz.

Methods with Description
boolean after(Date date) Date nesnesi çağrılırken belirtilen tarihten sonra bir tarih içeriyorsa, true değeri döndürür, aksi halde false değeri döndür.
boolean before(Date date) Date nesnesi çağrılırken belirtilen tarihten önce bir tarih içeriyorsa, true değeri döndürür, aksi halde false değeri döndür.
int compareTo(Date date) Çağrılan nesnenin değeri ile bu tarihi karşılaştırır. Değerler eşit ise , 0 döndürür. Çağrılan nesne tarihten daha önce ise negatif değer döndürür. Çağrılan nesne tarihten daha sonra ise pozitif değer döndürür.
boolean equals(Object date) Date nesnesi çağrılırken belirtilen tarihe eşit bir tarih ve zaman içeriyorsa, true değeri döndürür, aksi halde false değeri döndür.
String toString() Çağrılan Date nesnesini string türüne çevirir ve değeri döndürür.

13.1 Güncel Tarih ve Zamanı Alma

Javada, güncel tarih ve zamanı alma çok kolaydır. Aşağıdaki gibi güncel tarih ve zamanı yazdırmak için basit bir Date nesnesini toString() metodu ile kullanabilirsiniz.

```
import java.util.Date;

public class DateDemo {
    public static void main(String args[]) {
        // Instantiate a Date object
        Date date = new Date();

        // display time and date using toString()
        System.out.println(date.toString());
    }
}
```

Bu aşağıdaki sonucu üretecektir:

```
Mon May 04 09:51:52 CDT 2012
```

13.2 Tarih Karşılaştırma

2 farklı tarihi karşılaştırmak için iki adet yöntem vardır.

1. before(), after() ve equals() metotlarını kullanabilirsiniz.
2. compareTo() metodunu kullanabilirsiniz.

Örnek:

```
import java.util.*;
import java.text.*;

public class DateDemo {
    public static void main(String args[]) {

        Date d1 = new Date();
        SimpleDateFormat ft1 = new SimpleDateFormat ("yyyy.MM.dd hh:mm:ss");
        ft1.format(d1);

        Date d2 = new Date( );
        SimpleDateFormat ft2 = new SimpleDateFormat ("2012.11.dd hh:mm:ss");
        ft2.format(d2);

        System.out.println("D1: " + ft1.format(d1) );
        System.out.println("D2: " + ft2.format(d2) );

        System.out.println(d1.after(d2));
        System.out.println(d1.before(d2));
        System.out.println(d1.compareTo(d2));
        System.out.println(d1.equals(d2));
    }
}
```

Bu aşağıdaki sonucu üretecektir:

```
D1: 2013.03.10 10:38:12
D2: 2012.11.10 10:38:12
false
true
-1
false
```

13.3 Tarih Biçimlendirme (SimpleDateFormat)

SimpleDateFormat, yerele duyarlı bir şekilde tarihleri biçimlendirmek ve öğelerine ayırmak için kullanılan somut bir sınıftır. SimpleDateFormat, tarih-zaman biçimlendirme için herhangi kullanıcı tanımlı bir kalıp seçerek başlatmanıza izin verir.

```
import java.util.*;
import java.text.*;

public class DateDemo {
    public static void main(String args[]) {

        Date dNow = new Date( );
        SimpleDateFormat ft = new SimpleDateFormat ("yyyy.MM.dd hh:mm:ss");

        System.out.println("Current Date: " + ft.format(dNow));
    }
}
```

Bu aşağıdaki sonucu üretecektir:

```
Current Date: 2013.03.10 10:13:46
```

SimpleDateFormat biçimlendirme kodları

Zaman biçimini belirtmek için ,time pattern string kullanın. Bu kalıpta, bütün ASCII karakterleri, kalıp karakterleri olarak ayrılmıştır. Aşağıdaki tabloda gösterilmektedir:

Karakter	Tanımlama	Örnek
G	Çağ göstergesi	AD
y	Dört basamaklı yıllar	2001
M	Yılın ayı	July or 07
d	Ayın günü	10

h	Saat A.M./P.M. türünde (1~12)	12
H	Saat (0~23)	22
m	Dakika	30
s	Saniye	55
S	Milisaniye	234
E	Haftanın günü	Tuesday
D	Yılın günü	360
F	Ay içinde haftanın günü	2 (second Wed. in July)
w	Yılın haftası	40
W	Ayın haftası	1
a	A.M./P.M. işaretçisi	PM
k	Günün saati (1~24)	24
K	Saat A.M./P.M. türünde (0~11)	10
z	Saat dilimi	Eastern Standard Time
'	Metin için kaçış	Delimiter
"	Tek tırnak	`

13.4 Tarihleri Stringlere Ayırıştırma

SimpleDateFormat sınıfı bazı ek metotlara sahiptir, özellikle parse(), bu metot SimpleDateFormat nesnesi içinde depolanmış biçime göre string'i ayırıştırır. Örnek:

```
import java.util.*;
import java.text.*;

public class DateDemo {

    public static void main(String args[]) {
        SimpleDateFormat ft = new SimpleDateFormat ("yyyy-MM-dd");

        String input = args.length == 0 ? "1818-11-11" : args[0];

        System.out.print(input + " Parses as ");

        Date t;

        try {
            t = ft.parse(input);
            System.out.println(t);
        } catch (ParseException e) {
            System.out.println("Unparseable using " + ft);
        }
    }
}
```

A sample run of the above program would produce following result:

```
$ java -jar DateDemo.jar
1818-11-11 Parses as Wed Nov 11 00:00:00 GMT 1818

$ java -jar DateDemo.jar 2007-12-01
2007-12-01 Parses as Sat Dec 01 00:00:00 GMT 2007
```

13.5 Kısa Süreli Uyku

Bir milisaniyeden bilgisayarınızın yaşam süresine kadar herhangi bir süre sleep yapabilirsiniz. Bu örnekte, program 3 saniye sleep yapmaktadır.

```
import java.util.*;

public class SleepDemo {
    public static void main(String args[]) {
        try {
            Date start = new Date();
            System.out.println(start.toString() + "\n");

            Thread.sleep(3000);

            Date stop = new Date();
            System.out.println(stop.toString() + "\n");

        } catch (Exception e) {
            System.out.println("Got an exception!");
        }
    }
}
```

Bu aşağıdaki sonucu üretecektir:

```
Sun May 03 18:04:41 GMT 2012
Sun May 03 18:04:44 GMT 2012
```

13.6 Geçen Zamanı Ölçme

Bazen zaman içinde bir noktayı milisaniye cinsinde ölçmeniz gerekebilir. O halde, yukarıdaki örneği bir kere daha yazalım:

```
import java.util.*;

public class DiffDemo {

    public static void main(String args[]) {
        try {

            long start = System.currentTimeMillis( );
            Date startdate = new Date();
```

```
        System.out.println(startdate.toString() + "\n");

        Thread.sleep(3000);

        long end = System.currentTimeMillis();
        Date stopdate = new Date();
        System.out.println(stopdate.toString() + "\n");

        long diff = end - start;

        System.out.println("Difference is : " + diff);

    } catch (Exception e) {
        System.out.println("Got an exception!");
    }
}
```

Bu aşağıdaki sonucu üretecektir:

```
Sun May 03 18:16:51 GMT 2012
Sun May 03 18:16:54 GMT 2012
Difference is : 3010
```