

Java

Date and Time

Date

- Java provides the Date class available in `java.util` package, this class encapsulates the current date and time.
- The Date class supports two constructors.
 1. The first constructor initializes the object with the current date and time.

`Date ()`

2. Second constructor accepts one argument that equals the number of milliseconds that have elapsed since midnight, January 1, 1970

`Date(long millisec)`

Example: DateDemo.java

Date Methods

Method	Description
<code>boolean after(Date date)</code>	Returns true if the invoking Date object contains a date that is later than the one specified by <i>date</i> . Otherwise, it returns false .
<code>boolean before(Date date)</code>	Returns true if the invoking Date object contains a date that is earlier than the one specified by <i>date</i> . Otherwise, it returns false .
<code>Object clone()</code>	Duplicates the invoking Date object.
<code>int compareTo(Date date)</code>	Compares the value of the invoking object with that of <i>date</i> . Returns 0 if the values are equal. Returns a negative value if the invoking object is earlier than <i>date</i> . Returns a positive value if the invoking object is later than <i>date</i> .
<code>boolean equals(Object date)</code>	Returns true if the invoking Date object contains the same time and date as the one specified by <i>date</i> . Otherwise, it returns false .
<code>long getTime()</code>	Returns the number of milliseconds that have elapsed since January 1, 1970.
<code>int hashCode()</code>	Returns a hash code for the invoking object.
<code>void setTime(long time)</code>	Sets the time and date as specified by <i>time</i> , which represents an elapsed time in milliseconds from midnight, January 1, 1970.
<code>String toString()</code>	Converts the invoking Date object into a string and returns the result.

Date Comparison

- There are following three ways to compare two dates:
 - You can use **getTime()** to obtain the number of milliseconds that have elapsed since midnight, January 1, 1970, for both objects and then compare these two values.
 - You can use the methods **before(), after(), and equals().**
 - **Example:** **new Date(99, 2, 12).before(new Date (99, 2, 18))** returns true. Because the 12th of the month comes before the 18th
 - You can use the **compareTo()** method, which is defined by the Comparable interface and implemented by Date.
- ↓
- **Example:** *CompareTwoDatesTest.java*

Date Formatting Using SimpleDateFormat

- **SimpleDateFormat** is a concrete class for formatting and parsing dates in a locale-sensitive manner.
- **SimpleDateFormat** allows you to start by choosing any user-defined patterns for date-time formatting.

Example: DateFormatDemo.java

Date Formatting Using Printf

- Format string can indicate the index of the argument to be formatted.
 - The index must immediately follow the % and it must be terminated by a \$.
- Alternatively, you can use the < flag. It indicates that the same argument as in the preceding format specification should be used again.

Format Specifier	Conversion Applied
%a %A	Floating-point hexadecimal
%b %B	Boolean
%c %C	Character
%d	Decimal integer
%h %H	Hash code of the argument
%e %E	Scientific notation
%f	Decimal floating-point
%g %G	Uses %e or %f, based on the value being formatted and the precision
%o	Octal integer
%n	Inserts a newline character
%s %S	String
%t %T	Time and date
%x %X	Integer hexadecimal
%%	Inserts a % sign <i>Example: DatePrintfDemo.java</i>

Parsing Strings Into Dates:

- The SimpleDateFormat class has some additional methods, notably parse(), which tries to parse a string.
- According to the format stored in the given SimpleDateFormat object.
- *Example: DateStringDemo.java*

2

Sleeping For A While

- You can sleep for any period of time from one millisecond up to the lifetime of your computer.
- *Thread.sleep()* method can be used to pause the execution of current thread for specified time in milliseconds.
- Example: SleepDemo.java

Measuring Elapsed Time

```
1 import java.util.*;
2 public class TimeElapse {
3     public static void main(String args[]) {
4         try {
5             long start = System.currentTimeMillis( );
6             System.out.println(new Date( ) + "\n");
7             Thread.sleep(3000);
8             System.out.println(new Date( ) + "\n");
9             long end = System.currentTimeMillis( ); //current time |
10            long diff = end - start;
11            System.out.println("Difference is : " + diff + " milliseconds");
12        } catch (Exception e) {
13            System.out.println("Got an exception!");
14        }
15    }
16 }
```

Problems Javadoc Declaration Console × Terminal

<terminated> TimeElapse [Java Application] C:\Program Files\Java\jdk-17.0.1\bin\javaw.exe (14-Nov-2021, 1:16:53 pm – 1:16:57 pm)

Sun Nov 14 13:16:54 IST 2021

Sun Nov 14 13:16:57 IST 2021

Difference is : 3060 milliseconds