

נספח לבינה בתוכנה 1
סמסטר א', מועד א', תש"ע
7/02/2010

java.util

Interface Set<E>

Type Parameters:

E - the type of elements maintained by this set

All Superinterfaces:

[Collection<E>](#), [Iterable<E>](#)

All Known Subinterfaces:

[NavigableSet<E>](#), [SortedSet<E>](#)

All Known Implementing Classes:

[AbstractSet](#), [ConcurrentSkipListSet](#), [CopyOnWriteArraySet](#), [EnumSet](#),
[HashSet](#), [JobStateReasons](#), [LinkedHashSet](#), [TreeSet](#)

Method Summary

boolean	add (E e) Adds the specified element to this set if it is not already present (optional operation).
boolean	addAll (Collection <? extends E > c) Adds all of the elements in the specified collection to this set if they're not already present (optional operation).
void	clear () Removes all of the elements from this set (optional operation).
boolean	contains (Object o) Returns true if this set contains the specified element.
boolean	containsAll (Collection <?> c) Returns true if this set contains all of the elements of the specified collection.
boolean	equals (Object o) Compares the specified object with this set for equality.
int	hashCode () Returns the hash code value for this set.
boolean	isEmpty () Returns true if this set contains no elements.
Iterator < E >	iterator () Returns an iterator over the elements in this set.
boolean	remove (Object o) Removes the specified element from this set if it is present (optional operation).
boolean	removeAll (Collection <?> c) Removes from this set all of its elements that are contained in the specified collection (optional operation).

boolean	retainAll (Collection <?> c) Retains only the elements in this set that are contained in the specified collection (optional operation).
int	size () Returns the number of elements in this set (its cardinality).
Object []	toArray () Returns an array containing all of the elements in this set.
<T> T[]	toArray (T[] a) Returns an array containing all of the elements in this set; the runtime type of the returned array is that of the specified array.

java.io

Class File

[java.lang.Object](#)

[java.io.File](#)

All Implemented Interfaces:

[Serializable](#), [Comparable](#)<[File](#)>

```
public class File
extends Object
implements Serializable, Comparable<File>
```

Constructor Summary

[File](#)([String](#) pathname)

Creates a new File instance by converting the given pathname string into an abstract pathname.

[File](#)([String](#) parent, [String](#) child)

Creates a new File instance from a parent pathname string and a child pathname string.

Method Summary

boolean	canExecute () Tests whether the application can execute the file denoted by this abstract pathname.
boolean	canRead () Tests whether the application can read the file denoted by this abstract pathname.
boolean	canWrite () Tests whether the application can modify the file denoted by this abstract pathname.
int	compareTo (File pathname) Compares two abstract pathnames lexicographically.
boolean	createNewFile () Atomically creates a new, empty file named by this abstract pathname if and only if a file with this name does not yet exist.
boolean	delete () Deletes the file or directory denoted by this abstract pathname.
boolean	exists () Tests whether the file or directory denoted by this abstract pathname exists.
File	getAbsoluteFile () Returns the absolute form of this abstract pathname.
String	getName () Returns the name of the file or directory denoted by this abstract pathname.
String	getPath () Converts this abstract pathname into a pathname string.

boolean	<u>isDirectory()</u> Tests whether the file denoted by this abstract pathname is a directory.
boolean	<u>isFile()</u> Tests whether the file denoted by this abstract pathname is a normal file.
long	<u>lastModified()</u> Returns the time that the file denoted by this abstract pathname was last modified.
long	<u>length()</u> Returns the length of the file denoted by this abstract pathname.
<u>String[]</u>	<u>list()</u> Returns an array of strings naming the files and directories in the directory denoted by this abstract pathname.
<u>String[]</u>	<u>list(</u> <u>FilenameFilter</u> <u> filter)</u> Returns an array of strings naming the files and directories in the directory denoted by this abstract pathname that satisfy the specified filter.
<u>File[]</u>	<u>listFiles()</u> Returns an array of abstract pathnames denoting the files in the directory denoted by this abstract pathname.
<u>File[]</u>	<u>listFiles(</u> <u>FileFilter</u> <u> filter)</u> Returns an array of abstract pathnames denoting the files and directories in the directory denoted by this abstract pathname that satisfy the specified filter.
<u>File[]</u>	<u>listFiles(</u> <u>FilenameFilter</u> <u> filter)</u> Returns an array of abstract pathnames denoting the files and directories in the directory denoted by this abstract pathname that satisfy the specified filter.
boolean	<u>renameTo(</u> <u>File</u> <u> dest)</u> Renames the file denoted by this abstract pathname.
boolean	<u>setExecutable(</u> boolean executable) A convenience method to set the owner's execute permission for this abstract pathname.
boolean	<u>setLastModified(</u> long time) Sets the last-modified time of the file or directory named by this abstract pathname.
boolean	<u>setReadable(</u> boolean readable) A convenience method to set the owner's read permission for this abstract pathname.
boolean	<u>setReadOnly()</u> Marks the file or directory named by this abstract pathname so that only read operations are allowed.
Boolean	<u>setWritable(</u> boolean writable) A convenience method to set the owner's write permission for this abstract pathname.
<u>String</u>	<u>toString()</u> Returns the pathname string of this abstract pathname.