

Class File

Constructor Summary

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

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

Class Scanner

Constructor Summary

[Scanner](#)([File](#) source)

Constructs a new `Scanner` that produces values scanned from the specified file.

[Scanner](#)([InputStream](#) source)

Constructs a new `Scanner` that produces values scanned from the specified input stream.

Method Summary

void	close ()
boolean	hasNext () Returns true if this scanner has another token in its input.
boolean	hasNextInt () Returns true if the next token in this scanner's input can be interpreted as an int value using the <code>nextInt()</code> method.
boolean	hasNextLine () Returns true if there is another line in the input of this scanner.
IOException	ioException () Returns the IOException last thrown by this Scanner 's underlying <code>Readable</code> .
String	next () Finds and returns the next complete token from this scanner.
int	nextInt () Reads the next token and
String	nextLine () Advances this scanner past the current line and returns the input that was skipped.
Scanner	useDelimiter (String pattern) Sets this scanner's delimiting pattern to a pattern constructed from the specified String .

Class FileReader

Constructor Summary

[FileReader](#)([File](#) file)

Creates a new `FileReader`, given the `File` to read from.

[FileReader](#)([String](#) fileName)

Creates a new `FileReader`, given the path of the file to read from.

Class BufferedReader

Constructor Summary	
	BufferedReader (Reader in) Create a buffering character-input stream that uses a default-sized input buffer.
Method Summary	
void	close () Close the stream.
int	read () Read a single character.
int	read (char[] cbuf, int off, int len) Read characters into a portion of an array.
String	readLine () Read a line of text.

Interface List<E>

Method Summary	
boolean	add (E e) Appends the specified element to the end of this list (optional operation).
void	add (int index, E element) Inserts the specified element at the specified position in this list (optional operation).
boolean	addAll (Collection <? extends E > c) Appends all of the elements in the specified collection to the end of this list, in the order that they are returned by the specified collection's iterator (optional operation).
boolean	contains (Object o) Returns true if this list contains the specified element.
E	get (int index) Returns the element at the specified position in this list.
E	remove (int index) Removes the element at the specified position in this list (optional operation).
boolean	removeAll (Collection <?> c) Removes from this list all of its elements contained in the specified collection (optional operation).
boolean	retainAll (Collection <?> c) Retains only the elements in this list that are contained in the specified collection.
int	size () Returns the number of elements in this list.

Interface Set<E>

Method Summary	
boolean	add (E e)
boolean	addAll (Collection <? extends E > c)
boolean	contains (Object o)
boolean	remove (Object o)
boolean	removeAll (Collection <?> c)
boolean	retainAll (Collection <?> c)
int	size ()

Interface Map<K,V>

Method Summary	
boolean	containsKey (Object key)
boolean	containsValue (Object value)
Set < Map.Entry < K , V >>	entrySet ()
V	get (Object key)
boolean	isEmpty ()
Set < K >	keySet ()
V	put (K key, V value)
V	remove (Object key)
int	size ()
Collection < V >	values () Returns a Collection view of the values contained in this map.
Nested Class Summary	
static interface	Map.Entry < K , V > A map entry (key-value pair).

Class Collections

Method Summary	
static boolean	addAll (Collection <? super I > c, I ... elements)
static void	copy (List <? super I > dest, List <? extends I > src) Copies all of the elements from one list into another. The destination list must be at least as long as the source list. If it is longer, the remaining elements in the destination list are unaffected.
static ArrayList < T >	list (Enumeration < T > e) Returns an array list containing the elements returned by the specified enumeration in the order they are returned by the enumeration
static void	reverse (List <?> list)
static void	sort (List < T > list) Sorts the specified list into ascending order, according to the <i>natural ordering</i> of its elements.
static void	sort (List < T > list, Comparator <? super I > c) Sorts the specified list according to the order induced by the specified comparator.
static T	max (Collection <? extends I > coll) Returns the maximum element of the given collection.
static T	min (Collection <? extends I > coll) Returns the minimum element of the given collection.

Class Exception

Method Summary

String	getMessage() Returns the detail message string of this exception.
void	printStackTrace() Prints this exception and its stack trace to the standard error stream.
void	printStackTrace(PrintStream s) Prints this exception and its stack trace to the specified print stream.
void	printStackTrace(PrintWriter s) Prints this exception and its stack trace to the specified print writer.
String	toString() Returns a short description of this exception.

org.eclipse.swt.events

Class SelectionEvent

Method Summary

Object	getSource() The object on which the Event occurred, e.g., the button that was clicked, the selected list item, etc.
------------------------	--

Class String

Method Summary

char	charAt(int index) Returns the char value at the specified index.
boolean	equalsIgnoreCase(String anotherString) Compares this String to another String , ignoring case considerations.
int	indexOf(String str) Returns the index within this string of the first occurrence of the specified substring.
int	lastIndexOf(String str, int fromIndex) Returns the index within this string of the last occurrence of the specified substring, searching backward starting at the specified index.
int	length() Returns the length of this string.
String	replaceAll(String regex, String replacement) Replaces each substring of this string that matches the given <u>regular expression</u> with the given replacement.
String[]	split(String regex) Splits this string around matches of the given <u>regular expression</u> .
String	substring(int beginIndex, int endIndex) Returns a new string that is a substring of this string.
String	toLowerCase() Converts all of the characters in this String to lower case using the rules of the default locale.
String	toUpperCase() Converts all of the characters in this String to upper case using the rules of the default locale.
String	trim() Returns a copy of the string, with leading and trailing whitespace omitted.