Table Of Content

Package codeshine

Class Summary

Activator

Controla el ciclo de vida del plugin

---

codeshine

Class Activator

```
java.lang.Object
    |
    +--org.eclipse.core.runtime.Plugin
        |
        +--org.eclipse.ui.plugin.AbstractUIPlugin
            |
            +--codeshine.Activator
```

All Implemented Interfaces:

org.osgi.framework.BundleActivator

< Fields > < Constructors > < Methods >

---

public class **Activator**
extends org.eclipse.ui.plugin.AbstractUIPlugin

Controla el ciclo de vida del plugin

Fields

PLUGIN_ID

```
public static final java.lang.String PLUGIN_ID
```

El plug-in ID

---

tts

```
public static TtsClass tts
```

Objeto de la clase TtsClass

Constructors

Activator

```
public  Activator()
```

Constructor

Methods

getDefault

```
public static Activator getDefault()
```

Devuelve el objeto compartido plugin

Returns:

El objeto compartido

---

getImageDescriptor

```
public static org.eclipse.jface.resource.ImageDescriptor
getImageDescriptor(java.lang.String path)
```

Crea y devuelve un nuevo ImageDescriptor para un archivo de imagen ubicado dentro del plugin especificado.

Parameters:

path - El path del plugin

Returns:

Un ImageDescriptor, o null si no se ha encontrado ninguna imagen.

---

start

```
public void start(org.osgi.framework.BundleContext context)
```

Se inicia la comunicacion para que el plugin pueda realizar sus actividades.

Parameters:

context - es un objeto que se usa para poder interactuar con el Framework

Overrides:

start in class org.eclipse.ui.plugin.AbstractUIPlugin

---

stop

```
public void stop(org.osgi.framework.BundleContext context)
```

Detiene la ejecucion del bundle y a ejecucion del plugin debe detenerse.

Parameters:

context - es un objeto que se usa para poder interactuar con el Framework

Overrides:

stop in class org.eclipse.ui.plugin.AbstractUIPlugin

Package codeshine.preferences

Interface Summary

IPreferenceConstants

Interfaz que define las constantes para el plugin

Class Summary

CodeAppearancePreferencePage

CodePreferencePage

Esta clase representa la pagina de preferencias que contribuye al dialogo de Preferencias.

CodeProfilesPreferencePage

Guarda todos los campos que has introducido en las preferencias para poder guardarlo como perfil predeterminado

CodeSpeechPreferencePage

Esta clase sirve para definir la ventana de las preferencias sobre el speech.

PreferenceInitializer

Clase que se usa para incializar los valores por defecto de las preferencias.

ProfileConfigProvider

Configura el perfil

---

codeshine.preferences

Class CodeAppearancePreferencePage

```
java.lang.Object
    |
    +--org.eclipse.jface.dialogs.DialogPage
        |
        +--org.eclipse.jface.preference.PreferencePage
            |
            +--org.eclipse.jface.preference.FieldEditorPreferencePage
                |
                +--codeshine.preferences.CodeAppearancePreferencePage
```

All Implemented Interfaces:

org.eclipse.jface.dialogs.IDialogPage, org.eclipse.jface.dialogs.IMessageProvider, org.eclipse.jface.preference.IPreferencePage, org.eclipse.jface.util.IPropertyChangeListener, org.eclipse.ui.IWorkbenchPreferencePage

---

< Constructors > < Methods >

---

public class **CodeAppearancePreferencePage**
extends org.eclipse.jface.preference.FieldEditorPreferencePage
implements org.eclipse.ui.IWorkbenchPreferencePage

Constructors

CodeAppearancePreferencePage

```
public  CodeAppearancePreferencePage()
```

Constructor por defecto se instancia el grid

CodeAppearancePreferencePage

```
public  CodeAppearancePreferencePage(java.lang.String title,
org.eclipse.jface.resource.ImageDescriptor image,
                                     int style)
```

Constructor, creara un objeto de tipo Field Editor Preference Page

Parameters:

title - el titulo
image - la imagen
style - tipo de estilo

Methods

init

```
public void init(org.eclipse.ui.IWorkbench workbench)
```

Inicializa a pagina de preferencias para el espacio de trabajo (workbench)

Parameters:

workbench - - el espacio de trabajo

---

performCancel

```
public boolean performCancel()
```

Notifica que se ha pulsado el boton cancelar de la pÃ¡gina

Returns:

false - Para poder cancelaro

Overrides:

performCancel in class org.eclipse.jface.preference.PreferencePage

---

performOk

```
public boolean performOk()
```

Notifica que se ha pulsado el boton OK

Returns:

True o False True para que se pulse el boton o False si se debe cancelar la acciÃ³n

Overrides:

performOk in class org.eclipse.jface.preference.FieldEditorPreferencePage

---

codeshine.preferences

Class CodePreferencePage

```
java.lang.Object
    |
    +--org.eclipse.jface.dialogs.DialogPage
        |
        +--org.eclipse.jface.preference.PreferencePage
            |
            +--org.eclipse.jface.preference.FieldEditorPreferencePage
                |
                +--codeshine.preferences.CodePreferencePage
```

All Implemented Interfaces:

org.eclipse.jface.dialogs.IDialogPage, org.eclipse.jface.dialogs.IMessageProvider, org.eclipse.jface.preference.IPreferencePage, org.eclipse.jface.util.IPropertyChangeListener, org.eclipse.ui.IWorkbenchPreferencePage

---

< Constructors > < Methods >

---

public class **CodePreferencePage**
extends org.eclipse.jface.preference.FieldEditorPreferencePage
implements org.eclipse.ui.IWorkbenchPreferencePage

Esta clase representa la pagina de preferencias que contribuye al dialogo de Preferencias. Se usa solo para modificar las preferencias

Constructors

CodePreferencePage

```
public  CodePreferencePage()
```

Constructor por defecto

Methods

createFieldEditors

`public void createFieldEditors()`

Se crean los field editors. Cada Field Editor sabe como guardarse y borrarse a el mismo.

Overrides:

createFieldEditors in class org.eclipse.jface.preference.FieldEditorPreferencePage

---

init

`public void init(org.eclipse.ui.IWorkbench workbench)`

Inicializa la pagina de preferencias para el espacio de trabajo (workbench)

Parameters:

workbench - el espacio de trabajo

---

performDefaults

`public void performDefaults()`

Carga los valores por defecto.

Overrides:

performDefaults in class org.eclipse.jface.preference.FieldEditorPreferencePage

---

codeshine.preferences

Class CodeProfilesPreferencePage

```
java.lang.Object
    |
    +--org.eclipse.jface.dialogs.DialogPage
        |
        +--org.eclipse.jface.preference.PreferencePage
            |
            +--org.eclipse.jface.preference.FieldEditorPreferencePage
                |
                +--codeshine.preferences.CodeProfilesPreferencePage
```

All Implemented Interfaces:

org.eclipse.jface.dialogs.IDialogPage, org.eclipse.jface.dialogs.IMessageProvider, org.eclipse.jface.preference.IPreferencePage, org.eclipse.jface.util.IPropertyChangeListener, org.eclipse.ui.IWorkbenchPreferencePage

---

< Constructors > < Methods >

---

public class **CodeProfilesPreferencePage**
extends org.eclipse.jface.preference.FieldEditorPreferencePage

implements org.eclipse.ui.IWorkbenchPreferencePage

Guarda todos los campos que has introducido en las preferencias para poder guardarlo como perfil predeterminado

Constructors

CodeProfilesPreferencePage

`public` **`CodeProfilesPreferencePage`**`()`

Constructor por defecto se instancia el grid

Methods

init

`public void` **`init`**`(org.eclipse.ui.IWorkbench workbench)`

Inicializa la pagina de preferencias para el espacio de trabajo (workbench)

Parameters:

workbench - El espacio de trabajo

---

performApply

`public void` **`performApply`**`()`

Este metodo se usa para guardar el perfil

Overrides:

performApply in class org.eclipse.jface.preference.PreferencePage

---

performOk

`public boolean` **`performOk`**`()`

Notifica que se ha pulsado el boton OK

Returns:

True o False True para que se pulse el boton o False si se debe cancelar la accion

Overrides:

performOk in class org.eclipse.jface.preference.FieldEditorPreferencePage

codeshine.preferences

Class CodeSpeechPreferencePage

```
java.lang.Object
    |
    +--org.eclipse.jface.dialogs.DialogPage
        |
        +--org.eclipse.jface.preference.PreferencePage
            |
            +--org.eclipse.jface.preference.FieldEditorPreferencePage
                |
                +--codeshine.preferences.CodeSpeechPreferencePage
```

All Implemented Interfaces:

org.eclipse.jface.dialogs.IDialogPage, org.eclipse.jface.dialogs.IMessageProvider, org.eclipse.jface.preference.IPreferencePage, org.eclipse.jface.util.IPropertyChangeListener, org.eclipse.ui.IWorkbenchPreferencePage

---

< Constructors > < Methods >

---

public class **CodeSpeechPreferencePage**
extends org.eclipse.jface.preference.FieldEditorPreferencePage
implements org.eclipse.ui.IWorkbenchPreferencePage

Esta clase sirve para definir la ventana de las preferencias sobre el speech.

Constructors

CodeSpeechPreferencePage

`public` **`CodeSpeechPreferencePage`**`()`

Constructor por defecto donde se instancia el grid

Methods

init

`public void` **`init`**`(org.eclipse.ui.IWorkbench workbench)`

Inicializa la pagina de preferencias para el espacio de trabajo (workbench)

Parameters:

workbench - el espacio de trabajo

---

performCancel

`public boolean performCancel()`

Notifica que se ha pulsado el boton cancelar de la pÃ¡gina

Returns:

false Para poder cancelaro

Overrides:

performCancel in class org.eclipse.jface.preference.PreferencePage

---

performOK

`public boolean performOK()`

Notifica que se ha pulsado el boton OK

Returns:

True o False True para que se pulse el boton o False si se debe cancelar la accion

---

codeshine.preferences

Interface IPreferenceConstants

< Fields >

---

public interface **IPreferenceConstants**

Interfaz que define las constantes para el plugin

Fields

BACK_COLOR

`public static final java.lang.String BACK_COLOR`

CUSTOM_PROFILE

`public static final java.lang.String CUSTOM_PROFILE`

FONT_COLOR

`public static final java.lang.String FONT_COLOR`

FONT_TYPE

public static final java.lang.String **FONT_TYPE**

HIGHTLIGHT

public static final java.lang.String **HIGHTLIGHT**

MAIN_PROFILE

public static final java.lang.String **MAIN_PROFILE**

PITCH

public static final java.lang.String **PITCH**

PROFILE

public static final java.lang.String **PROFILE**

PROFILECONTENT

public static final java.lang.String **PROFILECONTENT**

PROFILEPATH

public static final java.lang.String **PROFILEPATH**

P_BOOLEAN

public static final java.lang.String **P_BOOLEAN**

P_CHOICE

public static final java.lang.String **P_CHOICE**

P_COLOR

public static final java.lang.String **P_COLOR**

P_PATH

public static final java.lang.String **P_PATH**

P_STRING

public static final java.lang.String **P_STRING**

SOUND_EVENTS

public static final java.lang.String **SOUND_EVENTS**

SPECIAL

public static final java.lang.String **SPECIAL**

TEST_AREA

public static final java.lang.String **TEST_AREA**

TRAINING

public static final java.lang.String **TRAINING**

VOICE

public static final java.lang.String **VOICE**

---

codeshine.preferences

Class PreferenceInitializer

```
java.lang.Object
    |
    +--org.eclipse.core.runtime.preferences.AbstractPreferenceInitializer
        |
        +--codeshine.preferences.PreferenceInitializer
```

< Constructors > < Methods >

---

public class **PreferenceInitializer**
extends org.eclipse.core.runtime.preferences.AbstractPreferenceInitializer

Clase que se usa para incializar los valores por defecto de las preferencias.

Constructors

PreferenceInitializer

```
public  PreferenceInitializer()
```

Methods

initializeDefaultPreferences

```
public void initializeDefaultPreferences()
```

Se inicializan los valores por defecto

Overrides:

initializeDefaultPreferences in class
org.eclipse.core.runtime.preferences.AbstractPreferenceInitializer

codeshine.preferences

Class ProfileConfigProvider

```
java.lang.Object
    |
    +--org.eclipse.jface.viewers.ArrayContentProvider
        |
        +--codeshine.preferences.ProfileConfigProvider
```

All Implemented Interfaces:

ITableContentProvider, org.eclipse.jface.viewers.IStructuredContentProvider

---

< Constructors > < Methods >

---

public class **ProfileConfigProvider**
extends org.eclipse.jface.viewers.ArrayContentProvider
implements ITableContentProvider

Configura el perfil

Constructors

ProfileConfigProvider

```
public  ProfileConfigProvider()
```

Constructor

Methods

getColumnValue

```
public java.lang.Object getColumnValue(java.lang.Object element,
                                       int columnIndex)
```

Devuelve el valor de la columna para el indice especificado

Parameters:

element - el objeto elemento para el cual consulta el valor de la columna
columnIndex - el Ã-ndice del valor de la columna que queremos consultar.

Returns:

el valor del elemento columnIndex

---

getElements

```
public java.lang.Object[] getElements(java.lang.Object collection)
```

Devuelve en formato TokenList los objetos que se le pasan.

Parameters:

collection - Coleccion de objectos

Returns:

una Token List con los ebjetos que se reciben

Overrides:

getElements in class org.eclipse.jface.viewers.ArrayContentProvider

---

inputChanged

```
public void inputChanged(org.eclipse.jface.viewers.Viewer viewer,
                         java.lang.Object oldInput,
                         java.lang.Object newInput)
```

Notifica que el contenido que proviene del viewer ha cambiado a un elemento diferente

Parameters:

viewer - El visor
oldInput - Elemento antiguo
newInput - Elemento nuevo

Package codeshine.speech

Interface Summary

AudioRecorder.Recorder

Interfaz para la grabaciÃ³n

Class Summary

AudioCommon

Metodos comunes para ejemplos de muestras de audio.

AudioRecorder

Clase que permite grabar audio y convertirlo en un archivo Este programa abre dos lineas: una para grabar y otra para reproducir.

AudioRecorder.AbstractRecorder

AudioRecorder.DirectRecorder

Clase que sirve para poder hacer la grabacion en directo

TtsClass

La principal funcionalidad de esta clase es convertir el texto a voz TextToSpeech

---

codeshine.speech

Class AudioCommon

```
java.lang.Object
    |
    +--codeshine.speech.AudioCommon
```

< Constructors > < Methods >

---

public class **AudioCommon**
extends java.lang.Object

Metodos comunes para ejemplos de muestras de audio.

Constructors

AudioCommon

```
public  AudioCommon()
```

Methods

findTargetType

```
public static javax.sound.sampled.AudioFileFormat.Type
findTargetType(java.lang.String strExtension)
```

Metodo que intenta obtener un tipo de archivo de audio para la extension que le pasamos. Examina todos los tipos de archivo disponibles y si la extension que le pasamos coincide la devuelve.

Parameters:

strExtension - La extensiÃ³n

Returns:

un objeto que coinicide con la extensiÃ³n que le pasamos null si no coincide nada

---

getMixerInfo

```
public static javax.sound.sampled.Mixer.Info getMixerInfo(java.lang.String
strMixerName)
```

Este mÃ©todo trata de devolver un Mixer. Se busca en un array un mixer que coincida con el strMixerName que se le pasa

Parameters:

strMixerName - - nombre del mixer a buscar

Returns:

el objeto Mixer si lo encuentra, null si no.

---

getTargetDataLine

```
public static javax.sound.sampled.TargetDataLine
getTargetDataLine(java.lang.String strMixerName,
javax.sound.sampled.AudioFormat audioFormat,
                                                                    int
nBufferSize)
```

MÃ©todo que devuelve un TargetDataLine. CÃ³mo esto es complicado, se tiene que contruir un objeto Info que especifique las propiedades que buscamos en la linea. Primero se especifica que tipo de lÃ-nea queremos: SourceDataline (para reproducir), TargetDataLine (para grabar), Clip (repite la reproducciÃ³n). En este caso queremos hacer una captura normal, por lo que pedimos un TargetDataLine. DespuÃ©s, tenemos que pasar un objeto AudioFormat, para que la Linea sepa el formato de dato que le pasaremos. Java Sound establecerÃ¡ un tamaÃ±o del buffer por defecto.

Parameters:

strMixerName - Define el nombre del Mixer
audioFormat - Objeto de tipo AudioFormat
nBufferSize - TamaÃ±o del buffer

Returns:

null o el objeto TargetDataLine

---

isPcm

```
public static boolean isPcm(javax.sound.sampled.AudioFormat.Encoding encoding)
```

Verifica si la codificacion es PCM

Parameters:

encoding - La codificacion

Returns:

True o False

---

listMixersAndExit

```
public static void listMixersAndExit()
```

Crea una lista con los mixers que hay, si no hay ninguno, sale de la ejecucion.

---

listMixersAndExit

```
public static void listMixersAndExit(boolean bPlayback)
```

Lista los mixers. Solo los mixers que soportan TargetDataLines or SourceDataLines se aÃ±aden a la lista dependiendo del valor de bPlayback. Si no encuentra nada sale de la ejecuciÃ³n.

Parameters:

bPlayback - True o False

---

listSupportedTargetTypes

```
public static void listSupportedTargetTypes()
```

Lista los destinos compatibles

---

setDebug

```
public static void setDebug(boolean bDebug)
```

Modifica el nombre de debug

Parameters:

bDebug - boolean

---

codeshine.speech

Class AudioRecorder

```
java.lang.Object
    |
    +--codeshine.speech.AudioRecorder
```

< Constructors > < Methods >

---

public class **AudioRecorder**
extends java.lang.Object

Clase que permite grabar audio y convertirlo en un archivo Este programa abre dos lineas: una para grabar y otra para reproducir. En un bucle infinito lee de las lÃ-neas de grabar y las escribe en la de reproducir.

Constructors

AudioRecorder

```
public  AudioRecorder()
```

Methods

performRecognition

```
public java.lang.String performRecognition()
                        throws java.lang.Exception
```

Empieza el reconociemiento de voz, el audio que graba pasa por diversas conversiones hasta que se convierte en un String

Returns:

Una cadena String donde se refleja lo que le hemos dicho al reconocedor

Throws:

java.lang.Exception - Captura cualquier excepcion

---

restoreRecording

```
public void restoreRecording()
```

Antes de empezar una nueva grabacion se resetea el estado

---

startRecording

```
public void startRecording()
```

Empieza la grabacion

---

stopRecording

```
public void stopRecording()
```

Detiene la grabaciÃ³n

---

terminateRecogniser

```
public void terminateRecogniser()
```

Cuando hemos terminado de hablar, pulsamos el boton otra vez y se ejecuta este metodo, el de terminar de ejectarse el reconocedor.

codeshine.speech

Class AudioRecorder.AbstractRecorder

```
java.lang.Object
    |
    +--java.lang.Thread
        |
        +--codeshine.speech.AudioRecorder.AbstractRecorder
```

All Implemented Interfaces:

AudioRecorder.Recorder, java.lang.Runnable

Direct Known Subclasses:

AudioRecorder.DirectRecorder

< Constructors > < Methods >

---

public class **AudioRecorder.AbstractRecorder**
extends java.lang.Thread
implements AudioRecorder.Recorder

Constructors

AbstractRecorder

```
public  AbstractRecorder(javax.sound.sampled.TargetDataLine line,
                         javax.sound.sampled.AudioFileFormat.Type targetType,
                         java.io.File file)
```

Methods

start

```
public void start()
```

Empieza la grabaciÃ³n. Para realizar esto (i) la lÃ-nea empieza (ii) se ejecuta el thread.

Overrides:

start in class java.lang.Thread

---

stopRecording

```
public void stopRecording()
```

Detiene la grabacion. La linea debe pararse despues de que se quede vacia (con el mÃ©todo drain())

codeshine.speech

Class AudioRecorder.DirectRecorder

```
java.lang.Object
    |
    +--java.lang.Thread
        |
        +--AudioRecorder.AbstractRecorder
            |
            +--codeshine.speech.AudioRecorder.DirectRecorder
```

All Implemented Interfaces:
AudioRecorder.Recorder, java.lang.Runnable

< Constructors > < Methods >

---

public class **AudioRecorder.DirectRecorder**
extends AudioRecorder.AbstractRecorder

Clase que sirve para poder hacer la grabacion en directo

Constructors

DirectRecorder

```
public  DirectRecorder(javax.sound.sampled.TargetDataLine line,
                       javax.sound.sampled.AudioFileFormat.Type targetType,
                       java.io.File file)
```

Constructor

Parameters:

line - La lÃ-nea
targetType - Objeto que tiene el tipo de archivo
file - El archivo final

Methods

run

```
public void run()
```

Escribe una secuencia de bytes que representan un archivo de audio del tipo de archivo especificado (mTargetType) en el archivo externo proporcionado (mfile)

Overrides:

run in class java.lang.Thread

---

codeshine.speech

Interface AudioRecorder.Recorder

< Methods >

public static interface **AudioRecorder.Recorder**

Interfaz para la grabaciÃ³n

Methods

start

`public void start()`

Empezar la grabaciÃ³n

stopRecording

`public void stopRecording()`

Detener la grabaciÃ³n

codeshine.speech

Class TtsClass

```
java.lang.Object
    |
    +--java.lang.Thread
        |
        +--codeshine.speech.TtsClass
```

All Implemented Interfaces:
java.lang.Runnable

< Constructors > < Methods >

public class **TtsClass**
extends java.lang.Thread

La principal funcionalidad de esta clase es convertir el texto a voz TextToSpeech

Constructors

TtsClass

`public` **`TtsClass`**`()`

Constructor

Methods

deallocate

`public void` **`deallocate`**`()`

Cancela la ejecucion del procesamiento de la voz

---

run

`public void` **`run`**`()`

Se ejecutan los hilos.

Overrides:

run in class java.lang.Thread

---

setProfile

`public void` **`setProfile`**`(`TokenList` `profile)`

Se modifica el parametro de tipo TokenList profile

Parameters:

profile - - El perfil

---

setText

`public void` **`setText`**`(java.lang.String input)`

Informa con un logger del string que se le pasa.

Parameters:

input - Una cadena string

---

speak

```
public void speak()
```

Ejecuta un hilo para poder hablar.

---

speak

```
public void speak(boolean preproccessor)
```

Determina si debe empezar la ejecuciÃ³n de habla.

Parameters:

preproccessor - si es F ejecutarÃ¡ un hilo. Si es T llamarÃ¡ al mÃ©todo speak()

Package codeshine.utils

Interface Summary

ITableContentProvider

Interfaz que se sirve para dar un soporte generico a JFace.

Class Summary

StringUtils

Contiene metodos de utilidad relacionados con el uso de Strings

TableFieldEditor

Una implementacion de FieldEditor que soporta la seleccion de datos tabulares.

TableViewerSorter

Extension generica de ViewerSorte para Viewer las instancias usan implementaciones de ITableContentProvider .

TableViewerSorterHandler

Permite la facilidad para las instancias de Table usando un arbitrario TableViewerSorter para ordenar.

Token

Se establecen los metodos necesarios para poder usar los objetos de tipo Token

TokenList

Clase que permite el funcionamiento de objetos de tipo Token para manejar eventos de tipo DefaultHandler en listas

TokensLabelProvider

Trie

Clase que define los objetos Trie y los metodos que pueden usar.

XMLHandler

XMLWriter

---

codeshine.utils

Interface ITableContentProvider

All Implemented Interfaces:

org.eclipse.jface.viewers.IStructuredContentProvider

< Methods >

---

public interface **ITableContentProvider**
implements org.eclipse.jface.viewers.IStructuredContentProvider

Interfaz que se sirve para dar un soporte generico a JFace.

Author:

Sebastian Machhausen

Methods

getColumnValue

```
public java.lang.Object getColumnValue(java.lang.Object element,
                                       int columnIndex)
```

Devuelve el valor de la columna para un objeto especificado en el indice de la columna que se le pasa.

Parameters:

element - El elemento del modelo para el cual consultar el valor de la columna
columnIndex - el indice de la columna

Returns:

el valor del elemento en ese columnIndex

---

codeshine.utils

Class StringUtils

```
java.lang.Object
    |
    +--codeshine.utils.StringUtils
```

< Fields > < Constructors > < Methods >

---

public class **StringUtils**
extends java.lang.Object

Contiene metodos de utilidad relacionados con el uso de Strings

Author:

Sebastian Machhausen

Fields

DEFAULT_DELIMITER

```
public static final char DEFAULT_DELIMITER
```

Delimitador por defecto para char

EMPTY_STRING

```
public static final java.lang.String EMPTY_STRING
```

Constante para Strings vacios

LINE_SEPARATOR

```
public static final java.lang.String LINE_SEPARATOR
```

Contante para las lineas separadoras

Constructors

StringUtils

```
public  StringUtils()
```

Methods

abbreviate

```
public static java.lang.String abbreviate(java.lang.String str,
                                          int maximumChars)
```

Se recorta el String que se le pasa si sobrepasa el valor de maximumChars. Si la longitud del String es mayor a ese valor se recorta. El resultado del String sera un substring de str que empezarÃ¡ en el indice 0 y acabarÃ¡ en el indice maximunChars concatenado con "..." para mostrar que la abrebiaciÃ³n se ha hecho. Si la longitud es menor a la que se le pasa la cadena se mantiene intacta.

Parameters:

str - El String a recortar..
maximumChars - El maximo de caracteres.

Returns:

El String recortado.

arrayToCSV

```
public static java.lang.String arrayToCSV(int[] input,
                                           char delimiter)
```

Convierte una cadema de int en un solo String en el cual los elementos estan seprados por un delimitador

Parameters:

input - El array de int
delimiter - El delimitador

Returns:

un solo String cuyos elementos estan separados por el delimitador

---

arrayToCSV

```
public static java.lang.String arrayToCSV(java.lang.String[] input,
                                           char delimiter)
```

Convierte una cadema de string en un solo String en el cual los elementos estan seprados por un delimitador

Parameters:

input - El array de Strings
delimiter - Es el delimitador

Returns:

un solo String cuyos elementos estan separados por el delimitador

---

arrayToCSV

```
public static java.lang.String arrayToCSV(org.eclipse.swt.graphics.Point[]
input,
                                           char delimiter)
```

Convierte el input en un solo String en el cual los elementos estan separados por un delimitador

Parameters:

input - Un array de POINT con los elementos
delimiter - El delimitador

Returns:

un solo String cuyos elementos estÃ¡n separados por el delimitador

---

convertEntities

```
public static java.lang.String convertEntities(java.lang.String text)
```

Convierte HTML a entidades de conversion de texto como " back to " y < back to < El texto ordinario no se cambia.

Parameters:

text - Es el String que representa el texto que va a ser procesado. No puede ser null

Returns:

El texto convertido; HTML 4.0. Si se le pasa un texto null, devuelve null.

---

convertHTML

```
public static java.lang.String convertHTML(java.lang.String text)
```

Convierte el HTML que se le pasa a a texto puro. Todas las entidades son convertidas y las etiquetas quitadas.

Parameters:

text - El string a convertir. Si es null se convertira a un String vacio.

Returns:

el texto convertido

---

csvToArray

```
public static java.lang.String[] csvToArray(java.lang.String input,
                                            char delimiter)
```

Usa el delimitador para convertir el input en un array de un solo string

Parameters:

input - El string cuyos elementos estaran separados por el delimitador
delimiter - El delimitador

Returns:

un array de string con el input separador por el delimitador.

---

csvToIntArray

```
public static int[] csvToIntArray(java.lang.String input,
                                  char delimiter)
```

Usa el delimitador para convertir el input en un array de int.

Parameters:

input - El string cuyos elementos estarÃ¡n separados por el delimitador
delimiter - El delimitador

Returns:

UN array de int con los valores del input

---

csvToPointArray

```
public static org.eclipse.swt.graphics.Point[]
csvToPointArray(java.lang.String input,
                                                  char delimiter)
```

Usa el delimitador para generar un array de Point con los elementos del input separados por el delimitador

Parameters:

input - El string a convertir
delimiter - El delimitador

Returns:

Un array de 'Point' con los elementos del input

---

entityToChar

```
public static char entityToChar(java.lang.String entity)
```

Convierte el string que recibe a un char especifico

Parameters:

entity. - La entidad String a convertir. Trabaja mas deprisa si es en minuscula.

Returns:

El caracter equivalete o 0 si no se conoce a entidad.

---

isEmpty

`public static boolean **isEmpty**(java.lang.String str)`

Comprueba si el String que se le pasa esta vacio.

Parameters:

str - El string que comprobar

Returns:

True si esta vacio y False en el caso contrario

---

nullValueToEmptyString

`public static java.lang.String **nullValueToEmptyString**(java.lang.String str)`

Convierte string que es null a un objeto String vacio. Si el valor no es null solo se recorta usando {@link java.lang.String#trim}

Parameters:

str - EL String a convertir

Returns:

Un String vacio si el valor que se le ha pasado es null. De lo contrario se recortara usando {@link java.lang.String#trim}

---

stripHTML

`public static java.lang.String **stripHTML**(java.lang.String text)`

Quita todas las etiquetas HTML del texto

Parameters:

text - El String al que hay que quitarle las etiquetas HTML

Returns:

el texto con las etiquetas quitadas o null si el texto que se le ha pasado es null

codeshine.utils

Class TableFieldEditor

```
java.lang.Object
    |
    +--org.eclipse.jface.preference.FieldEditor
        |
        +--codeshine.utils.TableFieldEditor
```

All Implemented Interfaces:

javax.accessibility.Accessible

< Constructors > < Methods >

---

public class **TableFieldEditor**
extends org.eclipse.jface.preference.FieldEditor
implements javax.accessibility.Accessible

Una implementacion de FieldEditor que soporta la seleccion de datos tabulares.

Author:

Sebastian Machhausen

Constructors

TableFieldEditor

```
public  TableFieldEditor(java.lang.String name,
                         java.lang.String labelText,
                         org.eclipse.swt.widgets.Composite parent,
                         org.eclipse.jface.viewers.IStructuredContentProvider
contentProvider,
                         org.eclipse.jface.viewers.ITableLabelProvider
labelProvider,
                         java.lang.String[] columnHeaders,
                         java.lang.Object input)
```

Crea una nueva instancia de TableFieldEditor

Parameters:

name - El nombre con el que trabaja el FielEditor
labelText - El text label del FieldEditor
parent - El padre que tiene el control del FieldEditor
contentProvider - El IStructuredContentProvider usado para consultar los valores de la tabla.
labelProvider - El ITableLabelProvider usado para convertir objetos de dominio a ui representaciones textuales especificas.
columnHeaders - Un array de objetos String que representa la cabecera de las columnas.
input - El objeto de entrada o modelos que contiene los datos para este FieldEditor

Methods

getAccessibleContext

```
public javax.accessibility.AccessibleContext getAccessibleContext()
```

Returns:

Siempre Null

---

getColumnWidth

```
public int getColumnWidth(int columnIndex)
```

Devuelve la anchura en pixeles de la columna que se especifica en columnIndex Si no encuentra una TableColum en la columnIndex el metodo devuelve 0

Parameters:

columnIndex - El indice de la columna de la que se quiere saber la anchura

Returns:

La anchura de la columna

---

getInput

```
public java.lang.Object getInput()
```

Devuelve el input

Returns:

el input

---

getNumberOfControls

```
public int getNumberOfControls()
```

Devuelve el numero de controles que tiene el TableFieldEditor Devuelve 1 si Table es el unico que tiene el control.

Overrides:

getNumberOfControls in class org.eclipse.jface.preference.FieldEditor

---

getSelection

`public java.lang.String` **`getSelection`**`()`

Devuelve el valor seleccionado del TableFieldEditor El valor retornado en este metodo depende de la columna seleccionada devuelta por {@link #getSelectionColumn()}. Si la columna seleccionada devuelve -1 la fila completa representada por los objetos de dominio es devuelta llamando a {@link #toString()} De lo contrario el valor de la columna respectiva se consulta y se devuelve usando el ITableLabelProvider vinculado al TableFieldEditor.

Returns:

El valor actualmente seleccionado o un String vacio si no hay seleccion

---

getSelectionColumn

`public int` **`getSelectionColumn`**`()`

Devuelve la seleccion de la columna que se representa en el indice cuyo valor se almacena/recupera en este TableFieldEditor. El valor -1 significa que la columna del objeto del dominio esta almacenada /recuperada. Esto se logra llamando a {@link #toString()} en el respectivo objeto de dominio.

Returns:

la columna cuyo valor es almacenado / recuperado en este TableField

---

getSortingColumn

`public int` **`getSortingColumn`**`()`

Devuelve el valor de la columna que se va a ordenar. Si la ordenaciÃ³n estÃ¡ deshabilitada se devueve -1

Returns:

El indice de la columna que se ha ordenado; O -1 si estÃ¡ desactivada la clasificacion

---

isSortAscending

`public boolean` **`isSortAscending`**`()`

Devuelve True si se ha ordenado de forma ascendente Falso si se ha ordenado de forma descendente o la clasificacion estÃ¡ deshabilitada

Returns:

True si se ha ordenado de forma ascendente False si se ha ordenado de forma descendente o la clasificacion esta deshabilitada

---

isSortingEnabled

```
public boolean isSortingEnabled()
```

Devuelve True si la clasificacion esta habilitada y False en caso contrario

Returns:

True si la clasificacion esta habilitada y False en caso contrario

---

setColumnWidth

```
public void setColumnWidth(int columnIndex,
                           int width)
```

Establece el ancho de la columna especificada en el columnIndex. Si no existe una TableColumn en la columnIndez el metodo no hace nada

Parameters:

columnIndex - el indice de la columna
width - la anchura en pixeles

---

setInput

```
public void setInput(java.lang.Object newInput)
```

Inicializa el input

Parameters:

newInput - el nuevo objeto input

---

setSelectionColumn

```
public void setSelectionColumn(int columnIndex)
```

Establece la columna que se selecciona en el columnIndex especificado El indice representa la columna cuyo valor se almacena / recupera en este Â Â Â Â Â * TableFieldEditor.

Parameters:

columnIndex - La columna cuyo valor se almacena/recupera en el TableFieldEditor

---

setSortingEnabled

```
public void setSortingEnabled(boolean enabled)
```

Habilita o deshabilita la ordenacion de tabla dependiendo del status El IStructuredContentProvider utilizado en este TableFieldEditor tiene que implementar la interfaz para habilitar la clasificacion

Parameters:

enabled - True para habilitar la clasificacion; Falso para deshabilitarlo

---

sort

```
public void sort(int columnIndex,
                 boolean ascending)
```

Ordena la Table en el valor especificado en columIndex en un orden especifico. Si la clasificacion esta deshabilitada este metodo no hace nada

Parameters:

columnIndex - El indice de la columna a ordenar
ascending - True para ordenarlo en ascendente o false para que sea en descendente.

---

codeshine.utils

Class TableViewerSorter

```
java.lang.Object
    |
    +--org.eclipse.jface.viewers.ViewerComparator
        |
        +--org.eclipse.jface.viewers.ViewerSorter
            |
            +--codeshine.utils.TableViewerSorter
```

< Constructors > < Methods >

---

public class **TableViewerSorter**
extends org.eclipse.jface.viewers.ViewerSorter

Extension generica de ViewerSorte para Viewer las instancias usan implementaciones de ITableContentProvider .

El extends ViewerSorter ya no estÃ¡ en uso

Author:

Sebastian Machhausen

Constructors

TableViewerSorter

```
public  TableViewerSorter(org.eclipse.jface.viewers.Viewer viewer,
                          ITableContentProvider contentProvider)
```

Crea una nueva instancia de TableViewerSorter enlazada a un especifico Viewer

Parameters:

viewer - El Viewer que se enlaza al TableViewerSorter
contentProvider - Contenido proporcionado

Methods

compare

```
public int compare(org.eclipse.jface.viewers.Viewer viewer,
                   java.lang.Object e1,
                   java.lang.Object e2)
```

Devuelve un numero negativo, cero o positivo dependiendo de sobre si el Primer elemento es menor que, igual o mayor que el segundo elemento.

Parameters:

viewer - El Viewer
e1 - El primer elemento
e2 - El segundo elemento

Returns:

Un numero negativo si el primer elemento es menor que el segundo; 0 si el primer y el segundo elemento son iguales; y un numero positivo si el segundo elemento es mayor al primero.

Overrides:

compare in class org.eclipse.jface.viewers.ViewerComparator

---

getSortingColumn

```
public int getSortingColumn()
```

Obtiene el indice de la columna para el cual se ha hecho una clasificacion

Returns:

El indice de la columna para el que se ha hecho la clasificacion

---

isAscending

`public boolean isAscending()`

Devuelve el tipo de ordenacion, True indica ascendente, False, descendente

Returns:

True para ascendente, False para descendente

---

setAscending

`public void setAscending(boolean ascending)`

Establece el tipo de ordenacion que se va a usar True indica ascendente y Falso descendente

Parameters:

ascending - True para ascendente, False para descendente

---

setSortingColumn

`public void setSortingColumn(int columnIndex)`

Establece el Ã-ndice de columna por el cual se va a realizar la clasificacion.

Parameters:

columnIndex - El indice de la columna por el cual se va a hacer la clasificacion

---

sort

`public void sort()`

Ordena los datos del modelo subyacente y actualiza los datos asociados del TableViewer para reflejar la nueva clasificaciÃ³n.

---

codeshine.utils

Class TableViewerSorterHandler

```
java.lang.Object
    |
    +--org.eclipse.swt.events.SelectionAdapter
        |
        +--codeshine.utils.TableViewerSorterHandler
```

All Implemented Interfaces:

org.eclipse.swt.events.SelectionListener

< Constructors > < Methods >

public class **TableViewerSorterHandler**
extends org.eclipse.swt.events.SelectionAdapter

Permite la facilidad para las instancias de Table usando un arbitrario TableViewerSorter para ordenar.

Author:

Sebastian Machhausen

Constructors

TableViewerSorterHandler

```
public  TableViewerSorterHandler(org.eclipse.swt.widgets.Table table,
                                 TableViewerSorter sorter)
```

Crea una nueva instancia de TableViewerSorterHandler y lo enlaza en la Table especificada usando el TableViewerSorter que se le pasa para ordenar el modelo de elementos

Parameters:

table - La Table que lo enlaza al TableViewerSorterHandler
sorter - clasifica el TableViewerSorter para usarlo para ordenar el modelo de elementos

Methods

dispose

```
public void dispose()
```

Coloca el TableViewerSorterHandler

---

sort

```
public void sort(int columnIndex)
```

Ordena el modelo subyacente por la columna que se le pasa. La manera de ordenarlo esta al reves

Parameters:

columnIndex - int the index of the column to sort

---

sort

```
public void sort(int columnIndex,
                 boolean ascending)
```

Ordena el modelo subyacente por la especifica columnIndex

Parameters:

columnIndex - El numero de la columna a ordenar.
ascending - True para ascendente, False para descendente.

---

widgetSelected

```
public void widgetSelected(org.eclipse.swt.events.SelectionEvent event)
```

Maneja el SelectionEvent que se dispara cuando la columna de clasificacion y/o ordena al Table cambiar. La clasificacion del modelo subyacente se usa para selecionar la columna a ordenar El orden estÃ¡ reservado, por ejemplo, de ascendente a descendente o al contrario.

Parameters:

event - El evento SelectionEvent que se dispara

Overrides:

widgetSelected in class org.eclipse.swt.events.SelectionAdapter

---

codeshine.utils

Class Token

```
java.lang.Object
    |
    +--codeshine.utils.Token
```

< Constructors > < Methods >

---

public class **Token**
extends java.lang.Object

Se establecen los metodos necesarios para poder usar los objetos de tipo Token

Constructors

Token

```
public  Token()
```

Constructor vacio

Token

```
public  Token(java.lang.String value,
              java.lang.String replacement,
              java.lang.String info)
```

Constructor de objeto token

Parameters:

value - String
replacement - String
info - String

Methods

equals

```
public boolean equals(Token token)
```

Devuelve True o False si los parametros value de dos objetos token son iguales

Parameters:

token - el objeto token

Returns:

True si son iguales, False si no los son

---

getInfo

```
public java.lang.String getInfo()
```

Devuelve el valor que hay en info

Returns:

Un string

---

getReplacement

```
public java.lang.String getReplacement()
```

Devuelve el valor que hay en Replacement

Returns:

Un string

---

getValue

`public java.lang.String` **`getValue`**`()`

Devuelve el valor que hay en Value

Returns:

Un string

---

setInfo

`public void` **`setInfo`**`(java.lang.String info)`

Establece un nuevo valor para info

Parameters:

info - un string del objeto Token

---

setReplacement

`public void` **`setReplacement`**`(java.lang.String replacement)`

Establece un nuevo valor para replacement

Parameters:

replacement - un string del objeto Token

---

setValue

`public void` **`setValue`**`(java.lang.String value)`

Establece un nuevo valor para value

Parameters:

value - un string del objeto Token

---

toString

`public java.lang.String` **`toString`**`()`

Construye una cadena de String con los datos del objeto

Overrides:

toString in class java.lang.Object

---

codeshine.utils

Class TokenList

```
java.lang.Object
    |
    +--org.xml.sax.helpers.DefaultHandler
        |
        +--codeshine.utils.TokenList
```

All Implemented Interfaces:

org.xml.sax.ContentHandler, org.xml.sax.DTDHandler, org.xml.sax.EntityResolver, org.xml.sax.ErrorHandler

---

< Constructors > < Methods >

---

public class **TokenList**
extends org.xml.sax.helpers.DefaultHandler

Clase que permite el funcionamiento de objetos de tipo Token para manejar eventos de tipo DefaultHandler en listas

Constructors

TokenList

```
public  TokenList()
```

Contructor de listas de tipo Token

TokenList

```
public  TokenList(java.io.File fInput)
        throws java.io.InvalidObjectException
```

Analiza el contenido del archivo especificado fInput como XML utilizando el archivo org.xml.sax.helpers.DefaultHandler.

Parameters:

fInput - Un objeto de tipo File

Throws:

java.io.InvalidObjectException - si se genera un objeto invalido

---

TokenList

```
public  TokenList(java.lang.String configFile)
        throws java.io.InvalidObjectException
```

Crea un objeto de tipo File dado un pathfile

Parameters:

configFile - el pathfile

Throws:

java.io.InvalidObjectException - si se genera un objeto invalido

Methods

addToken

```
public void addToken(Token t)
```

AÃ±ade un token a la lista

Parameters:

t - el token que se aÃ±ade a la lista

---

equals

```
public boolean equals(java.lang.Object obj)
```

Establece si los objetos Key del Hashtable son iguales

Returns:

True si los objetos Key del Hashtable son iguales o False en sentido contrario

Overrides:

equals in class java.lang.Object

---

getCollection

```
public java.util.Hashtable getCollection()
```

Devuelve una coleccion de objetos de tipo HashTable

Returns:

La coleccion de objetos Hashatble

---

getElements

```
public java.util.Enumeration getElements()
```

Indica el numero de elementos en un Hashtable

Returns:

Devuelve un int con el numero de elementos en el hastable

---

length

```
public int length()
```

Devuelve la longitud de la TokenList

Returns:

Un int que indica la longitud de la TokenList

---

main

```
public static void main(java.lang.String[] argv)
```

Crea una lista tokenlist

Parameters:

argv - Los argumentos que recibe el mÃ©todo

---

removeToken

```
public void removeToken(java.lang.String key)
```

Elimina el token que se especifica en el String key

Parameters:

key - Especifica el token a eliminar

---

toArray

```
public java.lang.Object[] toArray()
```

Devuelve un array de objetos

Returns:

Un array de objetos en el mismo orden en el que estaba

---

toString

`public java.lang.String **toString**()`

Devuelve un string literal del objeto

Overrides:

toString in class java.lang.Object

---

toTrie

`public Trie **toTrie**()`

AÃ±ade elementos a la lista

Returns:

El objeto Trie

---

toXML

`public java.lang.String **toXML**(java.lang.String filepath)`

Convierte el String que se le pasa a formato XML

Parameters:

filepath - el string que queremos convertir

Returns:

Un String en formato XML

---

codeshine.utils

Class TokensLabelProvider

```
java.lang.Object
    |
    +--org.eclipse.core.commands.common.EventManager
        |
        +--org.eclipse.jface.viewers.BaseLabelProvider
            |
            +--org.eclipse.jface.viewers.LabelProvider
                |
                +--codeshine.utils.TokensLabelProvider
```

All Implemented Interfaces:

org.eclipse.jface.viewers.IBaseLabelProvider, org.eclipse.jface.viewers.ILabelProvider, org.eclipse.jface.viewers.ITableLabelProvider

---

< Constructors > < Methods >

public class **TokensLabelProvider**
extends org.eclipse.jface.viewers.LabelProvider
implements org.eclipse.jface.viewers.ITableLabelProvider

Constructors

TokensLabelProvider

```
public  TokensLabelProvider()
```

Methods

getColumnImage

```
public org.eclipse.swt.graphics.Image getColumnImage(java.lang.Object element,
                                                     int columnIndex)
```

Devuelve el label de la imagen del elemento en la columnIndex

Parameters:

element - El objeto que representa la fila entera o null que indica que no hay ningun objeto
columnIndex - El indice de la columna

Returns:

el objeto Image o null si no hay nada

---

getColumnText

```
public java.lang.String getColumnText(java.lang.Object element,
                                      int columnIndex)
```

Devuelve el texto del label para la columna dado el elemento que se le pasa

Parameters:

element - El objeto que representa la fila entera o null que indica que no hay ningun objeto
columnIndex - el indice de la columna

Returns:

String o null Si no hay texto en la columnIndex

codeshine.utils

Class Trie

```
java.lang.Object
    |
    +--codeshine.utils.Trie
```

< Constructors > < Methods >

---

public class **Trie**
extends java.lang.Object

Clase que define los objetos Trie y los metodos que pueden usar.

Constructors

Trie

```
public  Trie()
```

Constructor

Methods

addChild

```
public void addChild(int index,
                     Trie child)
```

AÃ±ade un elemento en el indice que se le pasa. En el caso de que ahi hubiera un elemento lo mueve a la derecha sumando 1 a su indice.

Parameters:

index - El indice donde se quiere aÃ±adir el objeto
child - El objeto a aÃ±adir

---

addToken

```
public boolean addToken(char[] chars,
                        int pos,
                        java.lang.Object token)
```

Metodo recursivo que aÃ±ade el objeto token en el parametro data. El metodo solo aÃ±ade el objeto cuando llega al final de la lista.

Parameters:

chars - Array de objetos char
pos - La posicion
token - El objeto que se aÃ±ade

Returns:

Devuelve True si aÃ±ade al objeto y False en caso contrario

---

addToken

```
public boolean addToken(java.lang.Object token)
```

AÃ±ade el valor del objeto convertido de String a Char en un array de chars

Parameters:

token - El objeto a aÃ±adir

Returns:

True si lo aÃ±ade y False en caso contrario

---

appendChild

```
public void appendChild(Trie child)
```

AÃ±ade 'hijos' (elementos de tipo Trie) a la lista

Parameters:

child - El objeto que se quiere aÃ±adir

---

getChild

```
public Trie getChild(int pos)
```

Devuelve el objeto Trie en la pos que se indica

Parameters:

pos - La posicion

Returns:

El objeto

getChildrens

`public java.util.List` **`getChildrens`**`()`

Devuelve una lista de objetos Trie

Returns:

Una lista de objetos Trie

---

getNumChildren

`public int` **`getNumChildren`**`()`

Informa del numero de elementos que hay en la ArrayList

Returns:

El numero de elementos del ArrayList

---

getRoot

`public java.lang.Object` **`getRoot`**`()`

Devuelve el objeto root, el principal de la lista

Returns:

El objeto root

---

isLeaf

`public boolean` **`isLeaf`**`()`

Indica si la lista estÃ¡ vacÃ-a

Returns:

True si estÃ¡ vacÃ-a o False en el caso contrario

---

isRoot

`public boolean` **`isRoot`**`()`

Metodo que devuelve True o False para indicar si el objeto es Root

Returns:

True si el objeto es Null, False en caso contrario

---

isTerminal

`public boolean isTerminal()`

Indica si un objeto es el ultimo de la lista

Returns:

True si lo es, False si no.

---

main

`public static void main(java.lang.String[] args)`

Metodo que al ejecutarse crea objetos de tipo Trie y los aÃ±ade a una lista

Parameters:

args - Los argumentos que recibe el metodo

---

removeAll

`public void removeAll()`

Elimina todos los objetos de la lista

---

removeChild

`public void removeChild(int index)`

Elimina 'hijos' a la lista

Parameters:

index - el indice en la lista que se quiere eliminar

---

search

`public int search(java.lang.Object item)`

Busca el objeto item en la lista

Parameters:

item - el objeto a buscar

Returns:

La posicion en la que esta o -1 si no lo encuentra

---

setRoot

```
public void setRoot(java.lang.Object root)
```

Establece un valor para root

Parameters:

root - El objeto a modificar

---

toString

```
public java.lang.String toString(java.lang.String indent)
```

Concatena los valores de la lista en indent

Parameters:

indent - String al que se iran concatenando los valores de la lista

Returns:

Un String que contendra el string inicial indent + los valores d ela lista

---

trackRecursive

```
public Token trackRecursive(char[] input,
                            int current)
```

Si la lista tiene un solo elemento lo devuelve. Si llega al final, tambiÃ©n. En caso de que haya mas elementos en la lista los va analizando hasta que se den los casos anteriores.

Parameters:

input - El array de objetos tipo Char
current - un numero

Returns:

El objeto token, si el metodo se ha ejecutado bien tendra valor, si no, devolvera null

---

codeshine.utils

Class XMLHandler

```
java.lang.Object
    |
    +--org.xml.sax.helpers.DefaultHandler
        |
        +--codeshine.utils.XMLHandler
```

All Implemented Interfaces:

org.xml.sax.ContentHandler, org.xml.sax.DTDHandler, org.xml.sax.EntityResolver, org.xml.sax.ErrorHandler

---

< Fields > < Constructors > < Methods >

public class **XMLHandler**
extends org.xml.sax.helpers.DefaultHandler

Fields

out

```
public static java.io.Writer out
```

Constructors

XMLHandler

```
public  XMLHandler(TokenList tl)
```

Constructor

Parameters:

tl - la TokenList

Methods

startElement

```
public void startElement(java.lang.String namespaceURI,
                         java.lang.String lName,
                         java.lang.String qName,
                         org.xml.sax.Attributes attrs)
             throws org.xml.sax.SAXException
```

Recibe una notificacion de que un elemento se estÃ¡ ejecutando. Por defecto no hace nada.

Parameters:

namespaceURI - El Namespace URI, o un String vacio si no hay un namespaceURI
lName - Nombre local (sin prefijo) o un String vacio si no se ha formado
qName - Nombre 'qualified' (con prefijo) o un Strin vacio si no se ha formado
attrs - Los atributos unidos al elemento. Si no hay atributos sera un objeto vacio

Overrides:

startElement in class org.xml.sax.helpers.DefaultHandler

Throws:

org.xml.sax.SAXException - cualquier excepcion de tipo SAX

codeshine.utils

Class XMLWriter

```
java.lang.Object
    |
    +--codeshine.utils.XMLWriter
```

< Constructors > < Methods >

---

public class **XMLWriter**
extends java.lang.Object

Constructors

XMLWriter

```
public  XMLWriter(java.util.Enumeration content,
                  java.lang.String filePath)
        throws java.io.IOException
```

Constructor

Parameters:

content - Objetos enumeration
filePath - El nombre del filepath

Throws:

java.io.IOException - excepcion de E/S

Methods

writeFile

```
public void writeFile()
```

Metodo que escribe los documentos XML

Package codeshine.views

Class Summary

CodeControl

Clase que implementa los metodos necesarios para el funcionamiento de la vista

CodeView

Clase que implementa los metodos necesarios para la interfaz

EscribirEnFichero1

Convierte ficheros de texto en voz

SampleView

Esta clase sirve para conetar el plugin a la vista workbench.

---

codeshine.views

Class CodeControl

```
java.lang.Object
    |
    +--codeshine.views.CodeControl
```

All Implemented Interfaces:

org.eclipse.swt.events.MouseListener

< Constructors > < Methods >

---

public class **CodeControl**
extends java.lang.Object
implements org.eclipse.swt.events.MouseListener

Clase que implementa los metodos necesarios para el funcionamiento de la vista

Constructors

CodeControl

```
public  CodeControl(org.eclipse.swt.custom.StyledText st)
```

Constructor

Parameters:

st - Es un atributo de tipo StyledText que permite definir la fuente, color del fondo entre otros.

Methods

getDisplay

```
public org.eclipse.swt.widgets.Display getDisplay()
```

Devuelve el display (pantalla) a la que esta asociado

Returns:

El display

---

launchFind

```
public void launchFind()
```

Metodo que ejecuta la opcion "Buscar"

---

launchPreferences

```
public void launchPreferences()
```

Metodo que ejecuta las preferencias del menu action

---

mouseDoubleClick

```
public void mouseDoubleClick(org.eclipse.swt.events.MouseEvent e)
```

Metodo que informa de un evento de doble click en el raton

---

mouseDown

```
public void mouseDown(org.eclipse.swt.events.MouseEvent e)
```

Metodo que informa de un evento de que el raton se ha movido hacia abajo

---

mouseUp

```
public void mouseUp(org.eclipse.swt.events.MouseEvent e)
```

Metodo que informa de un evento de que el raton se ha movido hacia arriba

---

redraw

```
public void redraw()
```

---

setBackground

```
public void setBackground(org.eclipse.swt.graphics.Color backgroundColor)
```

Establece un color de fondo

Parameters:

backgroundColor - El nuevo color de fondo

---

setFont

```
public void setFont(org.eclipse.swt.graphics.Font newFont)
```

Establece una nueva fuente

Parameters:

newFont -

---

setForeground

```
public void setForeground(org.eclipse.swt.graphics.Color foregroundColor)
```

Establece un color para el texto

Parameters:

foregroundColor - El nuevo color para el texto

---

setHighLight

```
public void setHighLight(org.eclipse.swt.graphics.Color hightlight)
```

Establece un nuevo color para el resaltado "Color hightlight"

Parameters:

hightlight - El nuevo color para el resaltado

codeshine.views

Class CodeView

```
java.lang.Object
    |
    +--org.eclipse.core.commands.common.EventManager
        |
        +--org.eclipse.ui.part.WorkbenchPart
            |
            +--org.eclipse.ui.part.ViewPart
                |
                +--codeshine.views.CodeView
```

All Implemented Interfaces:

org.eclipse.core.runtime.IExecutableExtension, org.eclipse.ui.ISelectionListener, org.eclipse.ui.IViewPart, org.eclipse.ui.IWorkbenchPart3, org.eclipse.ui.part.IWorkbenchPartOrientation

---

< Constructors > < Methods >

---

public class **CodeView**
extends org.eclipse.ui.part.ViewPart
implements org.eclipse.ui.ISelectionListener

Clase que implementa los metodos necesarios para la interfaz

Constructors

CodeView

`public` **`CodeView`**`()`

El constructor.

Methods

createPartControl

`public void` **`createPartControl`**`(org.eclipse.swt.widgets.Composite parent)`

Esto es un callback que nos permitira crear el viewer e inicializarlo

Overrides:

createPartControl in class org.eclipse.ui.part.WorkbenchPart

---

dispose

```
public void dispose()
```

Para la ejecucion del reconocedor de Iatros y del listener

Overrides:

dispose in class org.eclipse.ui.part.WorkbenchPart

---

fillTextWidget

```
public void fillTextWidget()
```

Se guarda el texto que hay en un widget

---

init

```
public void init(org.eclipse.ui.IViewSite site)
```

Inicializa el visor

Parameters:

site - La interfaz del visor

Overrides:

init in class org.eclipse.ui.part.ViewPart

---

selectionChanged

```
public void selectionChanged(org.eclipse.ui.IWorkbenchPart part,
                             org.eclipse.jface.viewers.ISelection selection)
```

Notifica al listener cuando se ha cambiado la seleccion Este metodo se llama cuando la selecciÃ³n cambia de uno a un valor no null, pero no cuando la seleccion cambia a null.

Parameters:

part - El espacio de trabajo
selection - La seleccion

---

setFocus

```
public void setFocus()
```

Establece que el receptor tenga el foco del teclado, de modo que cualquier tecla que se pulse la recibe

Overrides:

setFocus in class org.eclipse.ui.part.WorkbenchPart

codeshine.views

Class EscribirEnFichero1

```
java.lang.Object
    |
    +--codeshine.views.EscribirEnFichero1
```

< Constructors > < Methods >

public class **EscribirEnFichero1**
extends java.lang.Object

Convierte ficheros de texto en voz

Constructors

EscribirEnFichero1

```
public  EscribirEnFichero1(java.lang.String voiceName)
        throws java.lang.Exception
```

Constructor

Parameters:

voiceName - La voz en la que se quiere que se hable.

Throws:

java.lang.Exception - no se especifica que tipo de excepcion puede saltar

Methods

close

```
public void close()
            throws java.lang.Exception
```

Cancela la ejecucion del procesamiento de la voz

Throws:

java.lang.Exception - La excepcion no especificada que puede saltar

getAudioType

```
public static javax.sound.sampled.AudioFileFormat.Type
getAudioType(java.lang.String file)
```

Devuelve el tipo de archivo de audio que es un file (como wav o au)

Parameters:

file - El archivo del cual queremos saber el tipo de archivo

Returns:

El tipo de audio del file que se le pasa

---

getBasename

```
public static java.lang.String getBasename(java.lang.String path)
```

Devuelve el nombre del path que se le pasa

Parameters:

path - El nombre del fichero completo con la extension

Returns:

Devuelve un substring que esta entre la primera letra del path y el .

---

getExtension

```
public static java.lang.String getExtension(java.lang.String path)
```

Devuelve la extension del nombre del fichero que se le pasa

Parameters:

path - El nombre del fichero completo (nombre + .extension)

Returns:

Devuelve un substring con la extension del fichero

---

listAllVoices

```
public static void listAllVoices()
```

Lista todas las voces que hay disponibles

---

speak

```
public void speak(java.lang.String text)
            throws java.lang.Exception
```

Reproduce el texto que se le pasa

Parameters:

text - El texto que queremos que se reproduzca

Throws:

java.lang.Exception - La excepcion no especificada que salta

---

toFile

```
public void toFile(java.lang.String filename,
                   java.lang.String text)
            throws java.lang.Exception
```

Graba un fichero de audio

Parameters:

filename - El nombre del fichero
text - El texto que queremos grabar

Throws:

java.lang.Exception - La excepcion no especificada que puede saltar

---

codeshine.views

Class SampleView

```
java.lang.Object
    |
    +--org.eclipse.core.commands.common.EventManager
        |
        +--org.eclipse.ui.part.WorkbenchPart
            |
            +--org.eclipse.ui.part.ViewPart
                |
                +--codeshine.views.SampleView
```

All Implemented Interfaces:

org.eclipse.core.runtime.IExecutableExtension, org.eclipse.ui.IViewPart, org.eclipse.ui.IWorkbenchPart3, org.eclipse.ui.part.IWorkbenchPartOrientation

---

< Fields > < Constructors > < Methods >

---

public class **SampleView**
extends org.eclipse.ui.part.ViewPart

Esta clase sirve para conetar el plugin a la vista workbench. La vista muestra como los datos se obtienen del modelo. La clase muestra una implementacion real de como conectar el modelo disponible en este u

otro plugin. La vista se conecta al modelo usando un proveedor de contenido

Fields

ID

`public static final java.lang.String` **`ID`**

El ID del visor con la extension especificada.

Constructors

SampleView

`public` **`SampleView`**`()`

The constructor.

Methods

createPartControl

`public void` **`createPartControl`**`(org.eclipse.swt.widgets.Composite parent)`

Esto es un callback que nos permite crear el visor e inicializarlo.

Overrides:

createPartControl in class org.eclipse.ui.part.WorkbenchPart

---

setFocus

`public void` **`setFocus`**`()`

Pasa el foco de la solicitud al usuario

Overrides:

setFocus in class org.eclipse.ui.part.WorkbenchPart

INDEX

I

L

M

N

O

P

R

S

T

V

W

X