
Package
org.cdi.ppm

org.cdi.ppm Class Context

java.lang.Object

└─org.cdi.ppm.Context

All Implemented Interfaces:

java.lang.Cloneable

```
public class Context
extends java.lang.Object
implements java.lang.Cloneable
```

Representa un contexto determinado. Se parece a un `ByteBuffer`. (es decir, sencillamente almacena unos cuantos bytes de forma dinámica)

Field Summary

protected	buf Buffer con los n bytes
protected	pos Siguiete índice del array buf libre

Constructor Summary

public	Context(int maxOrder) Constructor que crea un nuevo contexto vacío
--------	---

Method Summary

void	append(byte a) Añade un byte a este contexto, aumentando su orden actual
void	clear() Vacía el contexto
Context	clone() Clona el contexto actual, devolviendo una copia independiente
void	drop() Reduce el orden actual del contexto en 1
byte	getByteAt(int p) Devuelve el byte en la posición especificada
int	indexOf(byte b) Devuelve la posición de la primera aparición del byte
int	order() Devuelve el orden actual del contexto
java.lang.String	toString() Devuelve una string con el contexto actual (para depuración)

Methods inherited from class

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

buf

protected byte **buf**

Buffer con los n bytes

pos

protected int **pos**

Siguiente índice del array buf libre

Constructors

Context

public **Context**(int maxOrder)

Constructor que crea un nuevo contexto vacio

Parameters:

maxOrder - orden máximo, para efectos de reservar memoria

Methods

order

public int **order**()

Devuelve el orden actual del contexto

Returns:

valor entre 0 y orden máximo que indica el orden actual de este contexto

drop

public void **drop**()
throws java.lang.IndexOutOfBoundsException

Reduce el orden actual del contexto en 1

Throws:

IndexOutOfBoundsException - si el contexto es de orden 0

clear

public void **clear**()

(continued on next page)

(continued from last page)

Vacía el contexto

append

```
public void append(byte a)
```

Añade un byte a este contexto, aumentando su orden actual

Parameters:

a - el byte a añadir

getBytesAt

```
public byte getBytesAt(int p)
```

Devuelve el byte en la posición especificada

Parameters:

p - el índice del byte a obtener

Returns:

el byte en la posición p

Throws:

`IndexOutOfBoundsException` - si el índice no está en el contexto

indexOf

```
public int indexOf(byte b)
```

Devuelve la posición de la primera aparición del byte

Parameters:

b - byte a buscar

Returns:

posición del byte en el contexto, o -1 si no encontrado.

clone

```
public Context clone()
```

Clona el contexto actual, devolviendo una copia independiente

Returns:

una copia del contexto actual

toString

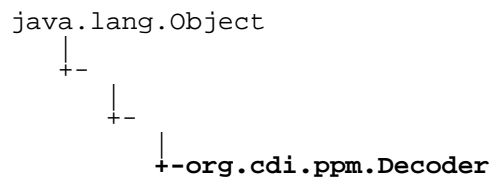
```
public java.lang.String toString()
```

Devuelve una string con el contexto actual (para depuración)

Returns:

string con el contexto actual

org.cdi.ppm Class Decoder



All Implemented Interfaces:

java.io.Closeable

Direct Known Subclasses:

HuffmanDecoder

```

public class Decoder
extends java.io.FilterInputStream
  
```

Clase base para un decompresor basado en el modelo PPM

Field Summary

protected final	currentContext El contexto actual
protected	eofReached Indica si hemos recibido un carácter EOF
protected final	maxOrder Almacena el orden máximo
protected final	model El modelo usado

Fields inherited from class

in

Constructor Summary

public	Decoder(java.io.InputStream in, int order) Crea una nueva instancia del decompresor
--------	--

Method Summary

ModelNode	decodeChar(java.util.List l) Decodifica un carácter, dada una lista con los caracteres posibles (debe ser sobrecargada por un decodificador)
int	read() Decodifica un byte

int	read(byte[] b, int off, int len) Decodifica hasta len bytes en el array especificado
-----	---

Methods inherited from class

available, close, mark, markSupported, read, read, read, reset, skip

Methods inherited from class

available, close, mark, markSupported, read, read, read, reset, skip

Methods inherited from class

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface

close

Fields

maxOrder

protected final int **maxOrder**

Almacena el orden máximo

currentContext

protected final org.cdi.ppm.Context **currentContext**

El contexto actual

model

protected final org.cdi.ppm.Model **model**

El modelo usado

eofReached

protected boolean **eofReached**

Indica si hemos recibido un carácter EOF

Constructors

Decoder

```
public Decoder(java.io.InputStream in,
               int order)
```

Crea una nueva instancia del decompresor

(continued from last page)

Parameters:

`in` - La stream donde está el archivo comprimido
`order` - Orden máximo de la predicción

Methods

read

```
public int read()  
    throws java.io.IOException
```

Decodifica un byte

Returns:

byte decodificado

Throws:

`IOException` - si ocurre un error de entrada/salida

read

```
public int read(byte[] b,  
               int off,  
               int len)  
    throws java.io.IOException
```

Decodifica hasta `len` bytes en el array especificado

Parameters:

`b` - el array de bytes
`off` - el índice del primer byte usable del búffer
`len` - la longitud del buffer

Returns:

el número de bytes decodificados

Throws:

`IOException` - si ocurre un error de entrada/salida

decodeChar

```
protected ModelNode decodeChar(java.util.List l)  
    throws java.io.IOException
```

Decodifica un carácter, dada una lista con los caracteres posibles (debe ser sobrecargada por un decodificador)

Parameters:

`l` - la lista con caracteres posibles

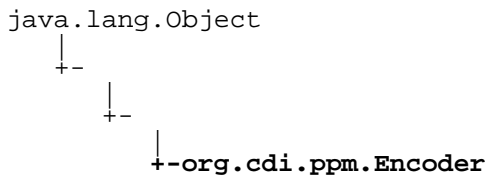
Returns:

el carácter encontrado

Throws:

`IOException` - si hay algun error de e/s

org.cdi.ppm Class Encoder



All Implemented Interfaces:

java.io.Flushable, java.io.Closeable

Direct Known Subclasses:

HuffmanEncoder

```
public class Encoder
extends java.io.FilterOutputStream
```

Clase base para un compresor basado en el modelo PPM

Field Summary

protected final	currentContext El contexto actual
protected final	maxOrder Almacena el orden máximo
protected final	model El modelo usado

Fields inherited from class

out

Constructor Summary

public	Encoder(java.io.OutputStream out, int order) Crea una nueva instancia del compresor PPM
--------	--

Method Summary

void	close() Termina la compresión, enviando el carácter de fin de archivo (no cierra el fichero de salida, debe hacerlo la subclase)
void	encodeChar(java.util.List l, ModelNode encode) Codifica un carácter dada una tabla de probabilidades (debe ser sobrecargada)
void	write(byte[] b, int off, int len) Codifica hasta len bytes del array especificado

void	write(int b) Codifica el byte especificado
------	---

Methods inherited from class

close, flush, write, write, write

Methods inherited from class

close, flush, write, write, write

Methods inherited from class

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface

close

Methods inherited from interface

flush

Fields

maxOrder

protected final int **maxOrder**

Almacena el orden máximo

currentContext

protected final org.cdi.ppm.Context **currentContext**

El contexto actual

model

protected final org.cdi.ppm.Model **model**

El modelo usado

Constructors

Encoder

public **Encoder**(java.io.OutputStream out,
int order)

Crea una nueva instancia del compresor PPM

Parameters:

out - La stream donde se escribirá el archivo resultante

order - Orden máximo la predicción

Methods

write

```
public void write(int b)
    throws java.io.IOException
```

Codifica el byte especificado

Parameters:

b - el byte

Throws:

IOException - si ocurre un error de entrada/salida

write

```
public void write(byte[] b,
    int off,
    int len)
    throws java.io.IOException
```

Codifica hasta len bytes del array especificado

Parameters:

b - el array de bytes

off - el índice del primer byte del array a codificar

len - el número de bytes a codificar

Throws:

IOException - si ocurre un error de entrada/salida

encodeChar

```
protected void encodeChar(java.util.List l,
    ModelNode encode)
    throws java.io.IOException
```

Codifica un carácter dada una tabla de probabilidades (debe ser sobrecargada)

Parameters:

l - la tabla/lista de probabilidades

encode - el carácter a codificar

Throws:

IOException - si hay un error de salida al escribir

close

```
public void close()
    throws java.io.IOException
```

Termina la compresión, enviando el carácter de fin de archivo (no cierra el fichero de salida, debe hacerlo la subclase)

org.cdi.ppm Class Excluder

```
java.lang.Object
└─org.cdi.ppm.Excluder
```

```
public class Excluder
extends java.lang.Object
```

Mantiene una lista de bytes excluidos (es decir, una variable booleana para cada byte posible [0..255])

Field Summary

protected static final	byteRange El número de valores máximos posibles para un tipo Byte. Value: 256
protected	excluded Un array de booleanos.
protected	exclusionsCount Cuenta el total de caracteres excluidos

Constructor Summary

public	Excluder()
--------	------------

Method Summary

void	clear() Limpia la lista de excluidos, olvidándose de ellos
void	exclude(byte b) Excluye un byte
boolean	isExcluded(byte b) Determina si un byte está excluido

Methods inherited from class

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

byteRange

```
protected static final int byteRange
```

El número de valores máximos posibles para un tipo Byte.
Constant value: **256**

excluded

protected boolean **excluded**

Un array de booleanos. `excluded[i]` es cierto si el byte `I` está excluido

exclusionsCount

protected int **exclusionsCount**

Cuenta el total de caracteres excluidos

Constructors

Excluder

public **Excluder**()

Methods

clear

public void **clear**()

Limpia la lista de excluidos, olvidándose de ellos

exclude

public void **exclude**(byte b)

Excluye un byte

Parameters:

b - byte a excluir

isExcluded

public boolean **isExcluded**(byte b)

Determina si un byte está excluido

Parameters:

b - byte a determinar

Returns:

true si está excluido

org.cdi.ppm

Class HuffmanDecoder

```
java.lang.Object
```

```
├─
```

```
├─
```

```
├─
```

```
├─org.cdi.ppm.HuffmanDecoder
```

All Implemented Interfaces:

```
java.io.Closeable
```

```
public class HuffmanDecoder
extends Decoder
```

Implementa un decodificador Huffman para el modelo PPM

Field Summary

protected	inb
	La fuente de entrada de bits

Fields inherited from class

```
currentContext, eofReached, maxOrder, model
```

Fields inherited from class

```
in
```

Constructor Summary

public	HuffmanDecoder(java.io.InputStream in, int order)
	Crea un nuevo decodificador

Method Summary

void	close()
	Cierra el decodificador.
ModelNode	decodeChar(java.util.List l)
	Decodifica un carácter, dada una lista con los caracteres posibles
HuffmanNode	descender(HuffmanNode padre)
	Desciende por el árbol de Huffman segun los bits de la entrada
void	encapsular(java.util.List l, java.util.PriorityQueue c)
	Ordena los elementos de una lista en una cola de prioridad

Methods inherited from class

```
decodeChar, read, read
```

Methods inherited from class

available, close, mark, markSupported, read, read, read, reset, skip

Methods inherited from class

available, close, mark, markSupported, read, read, read, reset, skip

Methods inherited from class

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface

close

Fields

inbprotected de.tivano.flash.swf.common.BitInputStream **inb**

La fuente de entrada de bits

Constructors

HuffmanDecoder

```
public HuffmanDecoder(java.io.InputStream in,
                      int order)
```

Crea un nuevo decodificador

Parameters:

in - de donde leer los datos codificados
 order - orden del modelo PPM

Methods

decodeChar

```
public ModelNode decodeChar(java.util.List l)
    throws java.io.IOException
```

Decodifica un carácter, dada una lista con los caracteres posibles

Parameters:

l - la lista con caracteres posibles

Returns:

el carácter encontrado

Throws:

IOException - si hay algun error de e/s

encapsular

```
protected void encapsular(java.util.List l,  
                           java.util.PriorityQueue c)
```

Ordena los elementos de una lista en una cola de prioridad

Parameters:

- l - lista de ModelNode
- c - cola donde se añaden los HuffmanNode

descender

```
protected HuffmanNode descender(HuffmanNode padre)  
    throws java.io.IOException
```

Desciende por el árbol de Huffman segun los bits de la entrada

Parameters:

- padre - el nodo desde el que comenzar a descender

Throws:

- IOException - si hay algun problema de e/s

close

```
public void close()  
    throws java.io.IOException
```

Cierra el decodificador.

org.cdi.ppm

Class HuffmanEncoder

java.lang.Object

└─

└─

└─

└─

└─org.cdi.ppm.HuffmanEncoder

All Implemented Interfaces:

java.io.Flushable, java.io.Closeable

public class **HuffmanEncoder**
extends Encoder

Implementa un codificador Huffman para el modelo PPM

Field Summary

protected	outb Salida de bits
-----------	------------------------

Fields inherited from class

currentContext, maxOrder, model

Fields inherited from class

out

Constructor Summary

public	HuffmanEncoder(java.io.OutputStream out, int order) Crea un nuevo codificador
--------	--

Method Summary

void	ascender(HuffmanNode node) Asciende por el árbol de Huffman, y escribe los bits que tocan
void	close() Cierra el codificador, enviando los últimos bits.
HuffmanNode	encapsular(java.util.List l, java.util.PriorityQueue c, ModelNode encode) Ordena los elementos de una lista en una cola de prioridad
void	encodeChar(java.util.List l, ModelNode encode) Codifica un carácter dada una tabla de probabilidades (debe ser sobrecargada)

Methods inherited from class


```
close, encodeChar, write, write
```

Methods inherited from class

```
close, flush, write, write, write
```

Methods inherited from class

```
close, flush, write, write, write
```

Methods inherited from class

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

Methods inherited from interface

```
close
```

Methods inherited from interface

```
flush
```

Fields

outb

```
protected de.tivano.flash.swf.common.BitOutputStream outb
```

Salida de bits

Constructors

HuffmanEncoder

```
public HuffmanEncoder(java.io.OutputStream out,
                      int order)
```

Crea un nuevo codificador

Parameters:

out - donde guardar los datos codificados
order - orden del modelo PPM

Methods

encodeChar

```
protected void encodeChar(java.util.List l,
                          ModelNode encode)
    throws java.io.IOException
```

Codifica un carácter dada una tabla de probabilidades (debe ser sobrecargada)

Parameters:

(continued from last page)

l - lista/tabla de probabilidades
encode - el carácter a codificar

Throws:

`IOException` - si hay un error de salida al escribir

encapsular

```
protected HuffmanNode encapsular(java.util.List l,  
    java.util.PriorityQueue c,  
    ModelNode encode)
```

Ordena los elementos de una lista en una cola de prioridad

Parameters:

l - lista de `ModelNode`
c - cola donde se añaden los `HuffmanNode`
encode - el nodo del model que representa el carácter a codificar

Returns:

el nodo de Huffman que representa el caracter a codificar

ascender

```
protected void ascender(HuffmanNode node)  
    throws java.io.IOException
```

Asciende por el árbol de Huffman, y escribe los bits que tocan

Parameters:

node - el nodo desde el que comenzar la ascensión

Throws:

`IOException` - si hay algun problema de e/s

close

```
public void close()  
    throws java.io.IOException
```

Cierra el codificador, enviando los últimos bits.

org.cdi.ppm

Class HuffmanNode

java.lang.Object

└--org.cdi.ppm.HuffmanNode

All Implemented Interfaces:

java.lang.Comparable

```
public class HuffmanNode
  extends java.lang.Object
  implements java.lang.Comparable
```

Encapsula un nodo del modelo PPM en el árbol de Huffman (tambien puede ser un nodo sin modelo)

Field Summary

	count La cuenta de apariciones del nodo en el modelo PPM
	der El hijo derecho
	izq El hijo izquierdo
	node El nodo del modelo PPM que encapsula
	parent El nodo Huffman padre a éste

Constructor Summary

public	HuffmanNode(HuffmanNode izq, HuffmanNode der) Crea un nuevo nodo del árbol de Huffman que es padre de dos nodos existentes
public	HuffmanNode(ModelNode node) Crea un nuevo nodo del árbol de Huffman envolviendo a un nodo del modelo PPM

Method Summary

int	compareTo(java.lang.Object arg) Compara dos nodos según su número de ocurrencias
-----	---

Methods inherited from class

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface

compareTo

Fields

node

```
org.cdi.ppm.ModelNode node
```

El nodo del modelo PPM que encapsula

count

```
int count
```

La cuenta de apariciones del nodo en el modelo PPM

parent

```
org.cdi.ppm.HuffmanNode parent
```

El nodo Huffman padre a éste

izq

```
org.cdi.ppm.HuffmanNode izq
```

El hijo izquierdo

der

```
org.cdi.ppm.HuffmanNode der
```

El hijo derecho

Constructors

HuffmanNode

```
public HuffmanNode(HuffmanNode izq,  
                    HuffmanNode der)
```

Crea un nuevo nodo del árbol de Huffman que es padre de dos nodos existentes

Parameters:

izq - el hijo izquierdo

der - el hijo derecho

HuffmanNode

```
public HuffmanNode(ModelNode node)
```

Crea un nuevo nodo del árbol de Huffman envolviendo a un nodo del modelo PPM

Parameters:

node - el nodo a envolver

Methods

(continued from last page)

compareTo

```
public int compareTo(java.lang.Object arg)
```

Compara dos nodos según su número de ocurrencias

Parameters:

arg - la clase a comparar

Returns:

0 si tienen las mismas apariciones

org.cdi.ppm Class Model

```
java.lang.Object
├--org.cdi.ppm.Model
```

```
public class Model
extends java.lang.Object
```

Implementa el árbol de contextos y conteo de caracteres

Field Summary

protected final	eof Una referencia al nodo que representa el carácter de fin de archivo
protected final	escape Una referencia al nodo que representa el carácter de escape
protected final	trie La raíz del arbol de búsqueda

Constructor Summary

public	Model() Crea un nuevo modelo con el árbol vacío (salvo escape y EOF)
--------	---

Method Summary

void	add(Context context, byte car) Tener en cuenta la aparición del carácter indicado en el contexto especificado
ModelNode	addSonNodeTo(ModelNode parent, byte car) Añadir un nodo hijo con el carácter especificado al nodo especificado.
void	addSubContext(Context context, int off, int len, byte car) (Interno) Tener en cuenta la aparición del contexto especificado <i>solo</i> en el orden en el que aparece tal contexto
void	excludeAll(java.util.List l, Excluder exclusions) Excluye todos los bytes de una lista de caracteres
java.util.List	getAllCharsList(Excluder exclusions) Obtiene una lista con todos los caracteres posibles, independientemente del contexto.
java.util.List	getListFromSonsOf(ModelNode parent, Excluder exclusions) Devuelve una lista con los hijos del nodo parent, incluye el nodo de escape
java.util.List	getNodeListFor(Context context, Excluder exclusions) Devolver la tabla de nodos para un contexto determinado
boolean	isEOF(ModelNode node) Indica si un nodo es el nodo de fin de archivo

boolean	<code>isEscape(ModelNode node)</code> Indica si un nodo es el nodo de escape
ModelNode	<code>searchEOFNode(java.util.List l)</code> Busca, entre los nodos de una lista, el carácter de EOF
ModelNode	<code>searchEscapeNode(java.util.List l)</code> Busca, entre los nodos de una lista, el carácter de escape
ModelNode	<code>searchNode(java.util.List l, byte car)</code> Busca, entre los nodos de una lista, el carácter especificado
ModelNode	<code>searchSonsOf(ModelNode parent, byte car)</code> Buscar un hijo del nodo especificado, con el carácter especificado

Methods inherited from class

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Fields

trie

`protected final org.cdi.ppm.ModelNode trie`

La raíz del arbol de búsqueda

escape

`protected final org.cdi.ppm.ModelNode escape`

Una referencia al nodo que representa el carácter de escape

eof

`protected final org.cdi.ppm.ModelNode eof`

Una referencia al nodo que representa el carácter de fin de archivo

Constructors

Model

`public Model()`

Crea un nuevo modelo con el árbol vacío (salvo escape y EOF)

Methods

add

`public void add(Context context,
byte car)`

Tener en cuenta la aparición del carácter indicado en el contexto especificado

(continued from last page)

Parameters:

context - el contexto que ha aparecido

addSubContext

```
protected void addSubContext(Context context,  
    int off,  
    int len,  
    byte car)
```

(Interno) Tener en cuenta la aparición del contexto especificado *solo* en el orden en el que aparece tal contexto

Parameters:

context - el contexto que ha aparecido

off - cuantos caracteres reducir el contexto para llegar al orden deseado

len - interpretar como contexto de qué orden

See Also:[add](#)

getAllCharsList

```
public java.util.List getAllCharsList(Excluder exclusions)
```

Obtiene una lista con todos los caracteres posibles, independientemente del contexto. La probabilidad será 1/(num caracteres posibles).

Parameters:

exclusions - una lista de exclusiones que no se devolverán en la lista

Returns:

la tabla de probabilidades de todos los caracteres

getNodeListFor

```
public java.util.List getNodeListFor(Context context,  
    Excluder exclusions)
```

Devolver la tabla de nodos para un contexto determinado

Parameters:

context - el contexto donde buscar

exclusions - una lista de exclusiones que no se devolverán en la lista

Returns:

La tabla de nodos, o null si el carácter no estaba.

searchSonsOf

```
protected ModelNode searchSonsOf(ModelNode parent,  
    byte car)
```

Buscar un hijo del nodo especificado, con el carácter especificado

Parameters:

parent - El nodo entre cuyos hijos buscar

car - El carácter a buscar

Returns:

(continued from last page)

El nodo que tiene el carácter especificado, o null si no se encuentra

addSonNodeTo

```
protected ModelNode addSonNodeTo(ModelNode parent,  
    byte car)
```

Añadir un nodo hijo con el carácter especificado al nodo especificado. Se supone que el nodo no está ya presente

Parameters:

`parent` - El nodo entre cuyos hermanos buscar
`car` - El carácter a buscar

Returns:

El nodo añadido

getListFromSonsOf

```
protected java.util.List getListFromSonsOf(ModelNode parent,  
    Excluder exclusions)
```

Devuelve una lista con los hijos del nodo parent, incluye el nodo de escape

Parameters:

`parent` - el nodo del cual obtener los hijos
`exclusions` - un contexto con bytes a excluir

Returns:

la lista de los hijos de parent sin incluir los bytes en exclusions

searchNode

```
public ModelNode searchNode(java.util.List l,  
    byte car)
```

Busca, entre los nodos de una lista, el carácter especificado

Parameters:

`l` - la lista donde buscar
`car` - carácter a buscar

Returns:

el nodo del modelo que tiene el carácter especificado

searchEscapeNode

```
public ModelNode searchEscapeNode(java.util.List l)
```

Busca, entre los nodos de una lista, el carácter de escape

Parameters:

`l` - la lista donde buscar

Returns:

el nodo del modelo que tiene el carácter especificado

searchEOFNode

```
public ModelNode searchEOFNode(java.util.List l)
```

Busca, entre los nodos de una lista, el carácter de EOF

Parameters:

l - la lista donde buscar

Returns:

el nodo del modelo que tiene el carácter especificado

excludeAll

```
public void excludeAll(java.util.List l,  
    Excluder exclusions)
```

Excluye todos los bytes de una lista de caracteres

Parameters:

l - la lista de nodos a excluir

exclusions - donde poner las exclusiones

isEscape

```
public boolean isEscape(ModelNode node)
```

Indica si un nodo es el nodo de escape

Parameters:

node - el nodo a comprobar

Returns:

true si node es nodo de escape

isEOF

```
public boolean isEOF(ModelNode node)
```

Indica si un nodo es el nodo de fin de archivo

Parameters:

node - el nodo a comprobar

Returns:

true si node es nodo de fin de archivo

org.cdi.ppm

Class ModelNode

java.lang.Object

└--org.cdi.ppm.ModelNode

```
public class ModelNode
    extends java.lang.Object
```

Un nodo para el árbol de contextos PPM

Field Summary

final	car El carácter que representa este nodo
	count La cuenta de veces que se ha dado este contexto.
	firstSon El contexto hijo (con un carácter más)
	nextBrother El contexto hermano (con el mismo número de caracteres, variando el último)
	parent El contexto padre (con un carácter menos)

Constructor Summary

public	ModelNode(byte car) Construye un nuevo nodo para el carácter señalado
--------	--

Methods inherited from class

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

car

final byte **car**

El carácter que representa este nodo

parent

org.cdi.ppm.ModelNode **parent**

El contexto padre (con un carácter menos)

nextBrother

`org.cdi.ppm.ModelNode` **nextBrother**

El contexto hermano (con el mismo número de caracteres, variando el último)

firstSon

`org.cdi.ppm.ModelNode` **firstSon**

El contexto hijo (con un carácter más)

count

`int` **count**

La cuenta de veces que se ha dado este contexto.

Constructors

ModelNode

`public` **ModelNode**(byte car)

Construye un nuevo nodo para el carácter señalado

Parameters:

car - el carácter indicado

org.cdi.ppm

Class PPM

```
java.lang.Object
|
+-org.cdi.ppm.PPM
```

```
public class PPM
extends java.lang.Object
```

Gestiona la comunicación con el usuario y llama al Encoder y al Decoder

Constructor Summary

public	PPM()
--------	-------

Method Summary

static void	compress(java.io.InputStream in, java.io.OutputStream out, int order) Comprime
static void	decompress(java.io.InputStream in, java.io.OutputStream out, int order) Descomprime
static void	leer_parametros(java.lang.String[] args) Lee los parámetros enviados al programa (desde línea de comandos o modo interactivo)
static void	main(java.lang.String[] args) Punto de entrada del programa PPM

Methods inherited from class

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

PPM

```
public PPM()
```

Methods

main

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

Punto de entrada del programa PPM

Parameters:

(continued from last page)

args - parámetros de línea de comandos

leer_parametros

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

Lee los parámetros enviados al programa (desde línea de comandos o modo interactivo)

Parameters:

args - parámetros de línea de comandos

compress

```
public static void compress(java.io.InputStream in,  
                             java.io.OutputStream out,  
                             int order)  
throws java.io.IOException
```

Comprime

Parameters:

in - desde donde

out - hacia donde

order - orden máximo

Throws:

IOException - si hay un problema de entrada/salida o defecto con el archivo

decompress

```
public static void decompress(java.io.InputStream in,  
                                java.io.OutputStream out,  
                                int order)  
throws java.io.IOException
```

Descomprime

Parameters:

in - desde donde

out - hacia donde

order - orden máximo

Throws:

IOException - si hay un problema de entrada/salida o defecto con el archivo

Index

A

add 23
addSonNodeTo 25
addSubContext 24
append 4
ascender 18

B

buf 3
byteRange 11

C

car 27
clear 3, 12
clone 4
close 10, 15, 18
compareTo 20
compress 30
Context 3
count 20, 28
currentContext 6, 9

D

decodeChar 7, 14
Decoder 6
decompress 30
der 20
descender 15
drop 3

E

encapsular 15, 18
encodeChar 10, 17
Encoder 9
eof 23
eofReached 6
escape 23
exclude 12

excludeAll 26
excluded 12
Excluder 12
exclusionsCount 12

F

firstSon 28

G

getAllCharsList 24
getBytesAt 4
getListFromSonsOf 25
getNodeListFor 24

H

HuffmanDecoder 14
HuffmanEncoder 17
HuffmanNode 20

I

inb 14
indexOf 4
isEOF 26
isEscape 26
isExcluded 12
izq 20

L

leer_parametros 30

M

main 29
maxOrder 6, 9
Model 23
model 6, 9
ModelNode 28

N

nextBrother 28

node 20

O

order 3

outb 17

P

parent 20, 27

pos 3

PPM 29

R

read 7

S

searchEOFNode 25

searchEscapeNode 25

searchNode 25

searchSonsOf 24

T

toString 4

trie 23

W

write 10