

MANUEL TÉCNICO

Objetos del paquete proyectocompiladores:

A continuación, listaremos los objetos que se utilizan en el proyecto

1. FrameCompilador.java
2. Lexer.flex
3. Lexer.java
4. LexerCup.flex
5. LexerCup.java
6. Pcompiladores.java
7. Sintaxis.cup
8. Sintaxis.java
9. Tokens.java
10. Sym.java

Librerías utilizadas en el proyecto:

Una vez entendido las opciones de carga, nos dirigimos a la opción de abrir archivo, del cual seguiremos tres pasos para cargar el archivo.

1. Java-cup-11^a.jar.
2. Java-cup.jar.
3. Jflex.jar.
4. Jdk 1.8 (Default).

CLASE FrameCompilador:

Lista de los objetos que contiene FrameCompilador.java

- ✓ java.lang.Object
- ✓ java.awt.Component
- ✓ java.awt.Container
- ✓ java.awt.Window
- ✓ java.awt.Frame
- ✓ javax.swing.JFrame
- ✓ proyectocompiladores.FrameCompilador

Todas las interfaces implementadas:

- ✓ java.awt.image.ImageObserver,
- ✓ java.awt.MenuContainer
- ✓ java.io.Serializable
- ✓ javax.accessibility.Accessible
- ✓ javax.swing.RootPaneContainer
- ✓ javax.swing.WindowConstants

Resumen del constructor

- ✓ Frame Compilador()
- ✓ Crea nuevo formulario FrameCompilador
- ✓ public static void main(java.lang.String[] args)

Clase proyectocompiladores.FrameCompilador extiende javax.swing.JFrame
implementa Serializable Campos serializados

- ✓ ButtonAnalizar1
- ✓ javax.swing.JButton ButtonAnalizar1

- ✓ ButtonAnalizarSintactico
- ✓ javax.swing.JButton ButtonAnalizarSintactico

- ✓ ButtonBorrarAnalizar
- ✓ javax.swing.JButton ButtonBorrarAnalizar

- ✓ ButtonBorrarDatosArchivo
- ✓ javax.swing.JButton ButtonBorrarDatosArchivo

- ✓ ButtonBorrarSintaxis

- ✓ javax.swing.JButton ButtonBorrarSintaxis
- ✓ ButtonLeerArchivo
- ✓ javax.swing.JButton ButtonLeerArchivo
- ✓ IngresarDatos
- ✓ javax.swing.JTextArea IngresarDatos
- ✓ ResultadoSintacrico
- ✓ javax.swing.JTextArea ResultadoSintacrico
- ✓ ResultadoanalizadorLexico
- ✓ javax.swing.JTextArea ResultadoanalizadorLexico
- ✓ jPanel1
- ✓ javax.swing.JPanel jPanel1
- ✓ jScrollPane1
- ✓ javax.swing.JScrollPane jScrollPane1
- ✓ jScrollPane2
- ✓ javax.swing.JScrollPane jScrollPane2
- ✓ jScrollPane3
- ✓ javax.swing.JScrollPane jScrollPane3

CLASE Pcompiladores:

Lista de los objetos que contiene Pcompiladores.java

- ✓ java.lang.Object
- ✓ proyectocompiladores.Pcompiladores

Resumen del constructo

- ✓ public Pcompiladores()
- ✓ public static void main(java.lang.String[] args) throws java.lang.Exception

Jerarquía de clases

- ✓ java.lang.Object
- ✓ java.awt.Component (implements java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable)
- ✓ java.awt.Container
- ✓ java.awt.Window (implements javax.accessibility.Accessible)
- ✓ java.awt.Frame (implements java.awt.MenuContainer)
- ✓ javax.swing.JFrame (implements javax.accessibility.Accessible, javax.swing.RootPaneContainer, javax.swing.WindowConstants)
- ✓ proyectocompiladores.FrameCompilador
- ✓ java_cup.runtime.lr_parser
- ✓ proyectocompiladores.Sintaxis
- ✓ proyectocompiladores.Pcompiladores
- ✓ proyectocompiladores.sym

CLASE Sintaxis:

Lista de los objetos que contiene Sintaxis.java

- ✓ java.lang.Object
- ✓ java_cup.runtime.lr_parser
- ✓ proyectocompiladores.Sintaxis
- ✓ public class Sintaxis
- ✓ extends java_cup.runtime.lr_parser

Resumen del constructor

- ✓ Sintaxis()
- ✓ Sintaxis(java_cup.runtime.Scanner s)
- ✓ Sintaxis(java_cup.runtime.Scanner s,java_cup.runtime.SymbolFactory sf)

Resumen del método

- ✓ short[][]: action_table() Access to parse-action table.
- ✓ java_cup.runtime.Symbol: do_action(int act_num,java_cup.runtime.lr_parser parser,java.util.Stack stack,int top) Invoke a user supplied parse action.
- ✓ Int: EOF_sym() EOF Symbol index.
 - ✓ Int: error_sym() error Symbol index.
 - ✓ java_cup.runtime.Symbol: getS()
 - ✓ protected void: init_actions() Action encapsulation object initializer.
 - ✓ short[][]: production_table() Access to production table.
 - ✓ short[][]: reduce_table() Access to reduce_goto table.
 - ✓ Int: start_production() Indicates start production.
 - ✓ Int: start_state() Indicates start state.
- ✓ Void: syntax_error(java_cup.runtime.Symbol s)

Jerarquía de clases

- ✓ java.lang.Object
- ✓ java.awt.Component (implements java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable)
- ✓ java.awt.Container ◦ java.awt.Window (implements javax.accessibility.Accessible) java.awt.Frame (implements java.awt.MenuContainer)
- ✓ javax.swing.JFrame (implements javax.accessibility.Accessible, javax.swing.RootPaneContainer, javax.swing.WindowConstants)
- ✓ proyectocompiladores.FrameCompilador
- ✓ java_cup.runtime.lr_parser
- ✓ proyectocompiladores.Sintaxis

- ✓ proyectocompiladores.Pcompiladores
- ✓ proyectocompiladores.sym

CLASE sym:

Resumen de campos

- ✓ static int Cadena
- ✓ static int Comillas
- ✓ static int CorcheteApertura
- ✓ static int CorcheteCierre
- ✓ static int Defecto
- ✓ static int Do
- ✓ static int Dolar
- ✓ static int Else
- ✓ static int EOF
- ✓ static int error
- ✓ static int ERROR
- ✓ static int Estructura
- ✓ static int Exponente
- ✓ static int For
- ✓ static int Identificador
- ✓ static int If
- ✓ static int Igual
- ✓ static int Int
- ✓ static int LlaveApertura
- ✓ static int LlaveCierre
- ✓ static int Main
- ✓ static int Numero
- ✓ static int OperadoresAritmeticos
- ✓ static int OperadoresAtribucion
- ✓ static int OperadoresBooleano
- ✓ static int OperadoresDeCadenas
- ✓ static int OperadoresLogicos
- ✓ static int OperadoresMasMenos

- ✓ static int OperadoresRelacionales
- ✓ static int PalabraReservada
- ✓ static int ParentesisApertura
- ✓ static int ParentesisCierre
- ✓ static int PuntoComa
- ✓ static int SwitchCase
- ✓ static int TipoDeDatos
- ✓ static int While

resumen del constructor

- ✓ sym()

Detalle de campos

PuntoComa

public static final int PuntoComa

See Also:Constant Field Values

Exponente

public static final int Exponente

See Also:Constant Field Values

If

public static final int If

See Also:Constant Field Values

CorcheteCierre

public static final int CorcheteCierre

See Also:Constant Field Values

CorcheteApertura

public static final int CorcheteApertura

See Also:Constant Field Values

Else

public static final int Else

See Also:Constant Field Values

LlaveCierre

public static final int LlaveCierre

See Also:Constant Field Values

PalabraReservada

public static final int PalabraReservada

See Also:Constant Field Values

OperadoresAritmeticos

public static final int OperadoresAritmeticos

See Also:Constant Field Values

Defecto

public static final int Defecto

See Also:Constant Field Values

OperadoresLogicos

public static final int OperadoresLogicos

See Also:Constant Field Values

ParentesisCierre

public static final int ParentesisCierre

See Also:Constant Field Values

OperadoresBooleano

public static final int OperadoresBooleano

See Also:Constant Field Values

While

public static final int While

See Also:Constant Field Values

Int

public static final int Int

See Also:Constant Field Values

Cadena

public static final int Cadena

See Also:Constant Field Values

For

public static final int For

See Also:Constant Field Values

EOF

public static final int EOF

See Also:Constant Field Values

LlaveApertura

public static final int LlaveApertura

See Also:Constant Field Values

TipoDeDatos

public static final int TipoDeDatos

See Also:Constant Field Values

error

public static final int error

See Also:Constant Field Values

OperadoresAtribucion

public static final int OperadoresAtribucion

See Also:Constant Field Values

Dolar

public static final int Dolar

See Also:Constant Field Values

ERROR

public static final int ERROR

See Also:Constant Field Values

Do

public static final int Do

See Also:Constant Field Values

OperadoresRelacionales

public static final int OperadoresRelacionales

See Also:Constant Field Values

SwitchCase

public static final int SwitchCase

See Also:Constant Field Values

Identificador

public static final int Identificador

See Also:Constant Field Values

Estructura

public static final int Estructura

See Also:Constant Field Values

OperadoresDeCadenas

public static final int OperadoresDeCadenas

See Also:Constant Field Values

OperadoresMasMenos

public static final int OperadoresMasMenos

See Also:Constant Field Values

Numero

public static final int Numero

See Also:Constant Field Values

ParentesisApertura

public static final int ParentesisApertura

See Also:Constant Field Values

Main

public static final int Main

See Also:Constant Field Values

Igual

public static final int Igual

See Also:Constant Field Values

Comillas

public static final int Comillas

See Also:Constant Field Values

Jerarquía de clases

- ✓ java.lang.Object
- ✓ java.awt.Component (implements java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable)
- ✓ java.awt.Container ◊ java.awt.Window (implements javax.accessibility.Accessible) java.awt.Frame (implements java.awt.MenuContainer)
- ✓ javax.swing.JFrame (implements javax.accessibility.Accessible, javax.swing.RootPaneContainer, javax.swing.WindowConstants)
- projectocompiladores.FrameCompilador
- ✓ java_cup.runtime.lr_parser
- ✓ proyectocompiladores.Sintaxis
- ✓ proyectocompiladores.Pcompiladores
- ✓ proyectocompiladores.sym

CLASE Tokens:

Resumen de constantes de enumeración

- ✓ Cadena
- ✓ Comillas
- ✓ CorcheteApertura
- ✓ CorcheteCierre
- ✓ Correo
- ✓ Defecto

- ✓ Do
- ✓ Dolar
- ✓ Else
- ✓ ERROR
- ✓ Estructura
- ✓ Exponente
- ✓ For
- ✓ Identificador
- ✓ If
- ✓ Igual
- ✓ Int
- ✓ Linea
- ✓ LlaveApertura
- ✓ LlaveCierre
- ✓ Main
- ✓ Numero
- ✓ OperadoresAritmeticos
- ✓ OperadoresAtribucion
- ✓ OperadoresBooleano
- ✓ OperadoresDeCadenas
- ✓ OperadoresLogicos
- ✓ OperadoresMasMenos
- ✓ OperadoresRelacionales
- ✓ PalabraReservada
- ✓ ParentesisApertura
- ✓ ParentesisCierre
- ✓ PuntoComa
- ✓ SwitchCase
- ✓ TipoDeDatos
- ✓ While

Metodos

static Tokens: valueOf(java.lang.String name) Returns the enum constant of this type with the specified name

static Tokens[]: values() Returns an array containing the constants of this enum type, in the order they are declared.

Detalle constante de enumeración

- ✓ PalabraReservada
public static final Tokens PalabraReservada
- ✓ If
public static final Tokens If
- ✓ Else
public static final Tokens Else
- ✓ Do
public static final Tokens Do
- ✓ While
public static final Tokens While
- ✓ For
public static final Tokens For
- ✓ Cadena
public static final Tokens Cadena
- ✓ SwitchCase
public static final Tokens SwitchCase
- ✓ Defecto
public static final Tokens Defecto
- ✓ CorcheteApertura
public static final Tokens CorcheteApertura

- ✓ CorcheteCierre
public static final Tokens CorcheteCierre
- ✓ LlaveApertura
public static final Tokens LlaveApertura
- ✓ LlaveCierre
public static final Tokens LlaveCierre
- ✓ ParentesisApertura
public static final Tokens ParentesisApertura
- ✓ ParentesisCierre
public static final Tokens ParentesisCierre
- ✓ OperadoresRelacionales
public static final Tokens OperadoresRelacionales
- ✓ Identificador
public static final Tokens Identificador
- ✓ Numero
public static final Tokens Numero
- ✓ TipoDeDatos
public static final Tokens TipoDeDatos
- ✓ Int
public static final Tokens Int
- ✓ OperadoresAritmeticos
public static final Tokens OperadoresAritmeticos
- ✓ OperadoresLogicos
public static final Tokens OperadoresLogicos

- ✓ OperadoresAtribucion
public static final Tokens OperadoresAtribucion
- ✓ Exponente
public static final Tokens Exponente
- ✓ OperadoresBooleano
public static final Tokens OperadoresBooleano
- ✓ OperadoresDeCadenas
public static final Tokens OperadoresDeCadenas
- ✓ OperadoresMasMenos
public static final Tokens OperadoresMasMenos
- ✓ Main
public static final Tokens Main
- ✓ PuntoComa
public static final Tokens PuntoComa
- ✓ Estructura
public static final Tokens Estructura
- ✓ Comillas
public static final Tokens Comillas
- ✓ Linea
public static final Tokens Linea
- ✓ Igual
public static final Tokens Igual
- ✓ Correo
public static final Tokens Correo

- ✓ Dolar
public static final Tokens Dólar
- ✓ ERROR
public static final Tokens ERROR

CLASE Lexer:

Declaraciones utilizadas para la evolución de la sintaxis escrita en el aplicativo.

```
/* PALABRAS RESERVADAS QUE UTILIZAREMOS */
```

```
/* Para asignar un valor a una variable se utiliza la palabra reservada */
```

```
( Set | set | SET | PutS | PUTS | puts | expr | break | continue) {lexeme=yytext();  
return PalabraReservada;}
```

```
( if ) {lexeme=yytext(); return If;}
```

```
( else ) {lexeme=yytext(); return Else;}
```

```
( do ) {lexeme=yytext(); return Do;}
```

```
( while ) {lexeme=yytext(); return While;}
```

```
( for ) {lexeme=yytext(); return For;}
```

```
( String ) {lexeme=yytext(); return Cadena;}
```

```
( switch ) {lexeme=yytext(); return SwitchCase;}
```

```
( default ) {lexeme=yytext(); return Defecto;}
```

```
( int ) {lexeme=yytext(); return Int;}
```

```
( main ) {lexeme=yytext(); return Main;}
```

```
/* CORCHETES */
```

```
( "[" ) {lexeme = yytext(); return CorcheteApertura;}
```

```
( "]" ) {lexeme = yytext(); return CorcheteCierre;}
```

```
/*LLAVES*/
```

```
( "{" ) {lexeme=yytext(); return LlaveApertura;}
```

```
( "}" ) {lexeme=yytext(); return LlaveCierre;}
```

```
/*PARENTESIS DE APERTURA*/
```

```
( "(" ) {lexeme=yytext(); return ParentesisApertura;}
```

```
( ")" ) {lexeme=yytext(); return ParentesisCierre;}
```

```
/*OPERADORES RELACIONALES */
```

```
( ">" | "<" | "==" | "!=" | ">=" | "<=" ) {lexeme = yytext(); return  
OperadoresRelacionales;}
```

```
/* Identificador y números */
```

```
{L}({L}|{D})* {lexeme=yytext(); return Identificador;}
```

```
("(-{D}+")){D}+ {lexeme=yytext(); return Numero;}
```

```
/* Operadores Aritméticos */
```

```
("*" | "/" | "%" | "+" | "-") {lexeme = yytext(); return OperadoresAritmeticos;}
```

```
( "=" ) {lexeme=yytext(); return Igual;}
```

```
( "$" ) {lexeme=yytext(); return Dolar;}
```

```
( "\"" ) {lexeme=yytext(); return Comillas;}
```

```
( ";" ) {lexeme=yytext(); return PuntoComa;}
```

```
/* por ejemplo 2**2 dará como resultado 4. */
```

```
( "**" ) {lexeme=yytext(); return Exponente;}
```

```
/* Operadores logicos y Operadores Atribución */
```

```
( "&&" | "||" | "!" | "&" | "|" | "#" ) {lexeme=yytext(); return OperadoresLogicos;}
```

```
( "+=" | "-=" | "*=" | "/=" | "%=" ) {lexeme = yytext(); return OperadoresAtribucion;}
```

```
/*OPERADORES BOOLEANOS*/
```

```
( true | false ) {lexeme = yytext(); return OperadoresBooleano;}
```

```
/*OPERADORES RELACIONALES PARA CADENAS*/
```

```
( eq | ne | in | ni ) {lexeme = yytext(); return OperadoresDeCadenas;}
```

```
/* Comentarios */
```

```
( "//"(.)* ) {/*Ignore*/}
```

```
{espacio} {/*Ignore*/}
```

```
( "\n" ) {return Linea;}
```

```
/* Error de analisis */
```

```
. {return ERROR;}
```

CLASE LexerCup:

Declaraciones definidas para la evolución de la sintaxis

```
/* PALABRAS RESERVADAS QUE UTILIZAREMOS */

/* Para asignar un valor a una variable se utiliza la palabra reservada */
(Set | set| PutS | puts | expr | break | continue | incr) {return new
Symbol(sym.PalabraReservada, yychar, yyline, yytext());}

( if ) {return new Symbol(sym.If, yychar, yyline, yytext());}
( else ) {return new Symbol(sym.Else, yychar, yyline, yytext());}
( do ) {return new Symbol(sym.Do, yychar, yyline, yytext());}
( while ) {return new Symbol(sym.While, yychar, yyline, yytext());}
( for ) {return new Symbol(sym.For, yychar, yyline, yytext());}
( String ) {return new Symbol(sym.Cadena, yychar, yyline, yytext());}
( switch ) {return new Symbol(sym.SwitchCase, yychar, yyline, yytext());}
( default ) {return new Symbol(sym.Defecto, yychar, yyline, yytext());}
( int ) {return new Symbol(sym.Int, yychar, yyline, yytext());}
( main ) {return new Symbol(sym.Main, yychar, yyline, yytext());}

/*CORCHETES*/
( "[" ) {return new Symbol(sym.CorcheteApertura, yychar, yyline, yytext());}
( "]" ) {return new Symbol(sym.CorcheteCierre, yychar, yyline, yytext());}

/* LLAVES */
( "{" ) {return new Symbol(sym.LlaveApertura, yychar, yyline, yytext());}
```

```
( "}" ) {return new Symbol(sym.LlaveCierre, yychar, yyline, yytext());}
```

```
/* PARENTESIS DE APERTURA*/
```

```
( "(" ) {return new Symbol(sym.ParentesisApertura, yychar, yyline, yytext());}
```

```
( ")" ) {return new Symbol(sym.ParentesisCierre, yychar, yyline, yytext());}
```

```
/*OPERADORES RELACIONALES */
```

```
( ">" | "<" | "==" | "!=" | ">=" | "<=" ) {return new  
Symbol(sym.OperadoresRelacionales, yychar, yyline, yytext());}
```

```
/* Identificador y números */
```

```
{L}({L}|{D})* {return new Symbol(sym.Identificador, yychar, yyline, yytext());}
```

```
("-{D}+"){D}+ {return new Symbol(sym.Numero, yychar, yyline, yytext());}
```

```
/* Operadores Aritméticos */
```

```
( "*" | "/" | "%" | "+" | "-" ) {return new Symbol(sym.OperadoresAritmeticos, yychar,  
yyline, yytext());}
```

```
( "=" ) {return new Symbol(sym.Igual, yychar, yyline, yytext());}
```

```
( "$" ) {return new Symbol(sym.Dolar, yychar, yyline, yytext());}
```

```
( "\"" ) {return new Symbol(sym.Comillas, yychar, yyline, yytext());}
```

```
( ";" ) {return new Symbol(sym.PuntoComa, yychar, yyline, yytext());}
```

```
/* por ejemplo 2**2 dará como resultado 4. */
```

```
( "**" ) {return new Symbol(sym.Exponente, yychar, yyline, yytext());}
```

```
/* Operadores logicos Operadores Atribución*/  
( "&&" | "||" | "!" | "&" | "|" | "#" ) {return new Symbol(sym.OperadoresLogicos,  
yychar, yyline, yytext());}  
  
( "+=" | "-=" | "*=" | "/=" | "%=" ) {return new Symbol(sym.OperadoresAtribucion,  
yychar, yyline, yytext());}  
  
/*OPERADORES BOOLEANOS*/  
  
( true | false ) {return new Symbol(sym.OperadoresBooleano, yychar, yyline,  
yytext());}  
  
/*OPERADORES RELACIONALES PARA CADENAS*/  
  
( eq | ne | in | ni ) {return new Symbol(sym.OperadoresDeCadenas, yychar, yyline,  
yytext());}  
  
/* Comentarios */  
  
( "//"(.)* ) {/*Ignore*/}  
  
{espacio} {/*Ignore*/}  
  
( "\\n" ) {return Linea;}  
  
/* Error de analisis */  
  
. {return new Symbol(sym.ERROR, yychar, yyline, yytext());}
```