

Proyecto II : H.U.L.K.

Generado por Doxygen 1.9.8

1 Índice de espacios de nombres	1
1.1 Lista de paquetes	1
2 Índice jerárquico	3
2.1 Jerarquía de clases	3
3 Índice de clases	5
3.1 Lista de clases	5
4 Documentación de espacios de nombres	7
4.1 Referencia del espacio de nombres HULK	7
5 Documentación de clases	9
5.1 Referencia de la clase HULK.ArgumentsCountError	9
5.1.1 Descripción detallada	10
5.2 Referencia de la clase HULK.ArgumentTypeError	10
5.2.1 Descripción detallada	10
5.3 Referencia de la clase HULK.Atom	11
5.3.1 Descripción detallada	12
5.3.2 Documentación de funciones miembro	12
5.3.2.1 Analize()	12
5.3.2.2 Evaluate()	12
5.4 Referencia de la clase HULK.Binary_Exprpressions	12
5.4.1 Descripción detallada	13
5.4.2 Documentación de funciones miembro	13
5.4.2.1 CatchArgumentTypeError()	13
5.5 Referencia de la clase HULK.BooleanOperator	14
5.5.1 Descripción detallada	15
5.5.2 Documentación de funciones miembro	15
5.5.2.1 Analize()	15
5.5.2.2 Evaluate()	16
5.5.2.3 Operation()	16
5.6 Referencia de la clase HULK.Comparison	16
5.6.1 Descripción detallada	18
5.6.2 Documentación de funciones miembro	18
5.6.2.1 Analize()	18
5.6.2.2 Evaluate()	18
5.6.2.3 Operation()	19
5.7 Referencia de la clase HULK.Conditional	19
5.7.1 Descripción detallada	20
5.7.2 Documentación de funciones miembro	20
5.7.2.1 Analize()	20
5.7.2.2 Evaluate()	20
5.8 Referencia de la clase HULK.ConditionalErrors	21

5.8.1 Descripción detallada	22
5.9 Referencia de la clase HULK.DefaultError	22
5.9.1 Descripción detallada	22
5.10 Referencia de la clase HULK.DuplicateArgument	23
5.10.1 Descripción detallada	23
5.11 Referencia de la clase HULK.Expression	24
5.11.1 Descripción detallada	25
5.11.2 Documentación de funciones miembro	25
5.11.2.1 ActualToken()	25
5.12 Referencia de la clase HULK.Function	25
5.12.1 Descripción detallada	26
5.12.2 Documentación de funciones miembro	26
5.12.2.1 Analyze()	26
5.12.2.2 CheckfunctionCall()	27
5.12.2.3 Evaluate()	27
5.13 Referencia de la clase HULK.FunctionDeclaration	27
5.13.1 Descripción detallada	28
5.13.2 Documentación de funciones miembro	28
5.13.2.1 Analyze()	28
5.13.2.2 Evaluate()	29
5.14 Referencia de la clase HULK.HulkErrors	29
5.14.1 Descripción detallada	29
5.15 Referencia de la clase HULK.HulkExpression	30
5.15.1 Descripción detallada	31
5.16 Referencia de la clase HULK.IncorrectBinaryExpression	31
5.16.1 Descripción detallada	32
5.17 Referencia de la clase HULK.IncorrectOperator	32
5.17.1 Descripción detallada	33
5.18 Referencia de la clase HULK.Let_in	33
5.18.1 Descripción detallada	34
5.18.2 Documentación de funciones miembro	34
5.18.2.1 Analyze()	34
5.18.2.2 Evaluate()	34
5.19 Referencia de la clase HULK.Lexer	35
5.19.1 Descripción detallada	35
5.19.2 Documentación de funciones miembro	35
5.19.2.1 GetIncorrectToken()	35
5.19.2.2 IsBoolean()	36
5.19.2.3 IsID()	36
5.19.2.4 IsNumber()	36
5.19.2.5 IsString()	37
5.19.2.6 TokenizeInput()	37

5.19.2.7 TokenType()	37
5.19.3 Documentación de datos miembro	38
5.19.3.1 Key_Words	38
5.20 Referencia de la clase HULK.LexicalError	38
5.20.1 Descripción detallada	39
5.21 Referencia de la clase HULK.MathExpressions	39
5.21.1 Descripción detallada	40
5.21.2 Documentación de funciones miembro	40
5.21.2.1 Analyze()	40
5.21.2.2 Evaluate()	40
5.22 Referencia de la clase HULK.MultiplyExpression	41
5.22.1 Descripción detallada	42
5.22.2 Documentación de funciones miembro	42
5.22.2.1 Analyze()	42
5.22.2.2 Evaluate()	43
5.22.2.3 Operation()	43
5.23 Referencia de la clase HULK.PowerExpression	43
5.23.1 Descripción detallada	45
5.23.2 Documentación de funciones miembro	45
5.23.2.1 Analyze()	45
5.23.2.2 Evaluate()	45
5.23.2.3 Operation()	46
5.24 Referencia de la clase HULK.Print	46
5.24.1 Descripción detallada	47
5.24.2 Documentación de funciones miembro	47
5.24.2.1 Analyze()	47
5.24.2.2 Evaluate()	47
5.25 Referencia de la clase HULK.Program	48
5.26 Referencia de la clase HULK.SemanticError	48
5.26.1 Descripción detallada	48
5.27 Referencia de la clase HULK.SumExpression	49
5.27.1 Descripción detallada	50
5.27.2 Documentación de funciones miembro	50
5.27.2.1 Analyze()	50
5.27.2.2 Evaluate()	50
5.27.2.3 Operation()	51
5.28 Referencia de la clase HULK.SyntaxError	51
5.28.1 Descripción detallada	52
5.29 Referencia de la clase HULK.UnExpectedToken	52
5.29.1 Descripción detallada	53
5.30 Referencia de la clase HULK.Union	53
5.30.1 Descripción detallada	54

5.30.2 Documentación de funciones miembro	54
5.30.2.1 Analyze()	54
5.30.2.2 Evaluate()	55
5.30.2.3 Operation()	55

Capítulo 1

Índice de espacios de nombres

1.1. Lista de paquetes

Estos son los paquetes con breves descripciones (si están disponibles):

HULK	7
--------------------------------	-------------------

Capítulo 2

Índice jerárquico

2.1. Jerarquía de clases

Este listado de herencia está ordenado de forma general pero no está en orden alfabético estricto:

Exception	
HULK.HulkErrors	29
HULK.DefaultError	22
HULK.LexicalError	38
HULK.SemanticError	48
HULK.ArgumentTypeError	10
HULK.ArgumentsCountError	9
HULK.ConditionalErrors	21
HULK.DuplicateArgument	23
HULK.IncorrectBinaryExpression	31
HULK.IncorrectOperator	32
HULK.SyntaxError	51
HULK.UnExpectedToken	52
HULK.Expression	24
HULK.Atom	11
HULK.Binary_Exprsions	12
HULK.BooleanOperator	14
HULK.Comparison	16
HULK.MultiplyExpression	41
HULK.PowerExpression	43
HULK.SumExpression	49
HULK.Union	53
HULK.Conditional	19
HULK.Function	25
HULK.FunctionDeclaration	27
HULK.HulkExpression	30
HULK.Let_in	33
HULK.MathExpressions	39
HULK.Print	46
HULK.Lexer	35
HULK.Program	48

Capítulo 3

Índice de clases

3.1. Lista de clases

Lista de clases, estructuras, uniones e interfaces con breves descripciones:

HULK.ArgumentsCountError	Representa el error de cantidad de argumentos incorrectos	9
HULK.ArgumentTypeError	Representa el error de tipo de argumento incorrecto	10
HULK.Atom	Representa las expresiones atómicas	11
HULK.Binary_Exprsions	Clase abstracta de expresiones binarias	12
HULK.BooleanOperator	Representa las expresiones binarias booleanas de & y 	14
HULK.Comparison	Representa las expresiones binarias de (< , > , <= , >= , == , !=)	16
HULK.Conditional	Representa las expresiones condicionales if-else	19
HULK.ConditionalErrors	Representa el error de expresiones condicionales	21
HULK.DefaultError	Otros errores (Stack overflow , Division by zero , missing ;)	22
HULK.DuplicateArgument	Representa el error de argumentos duplicados	23
HULK.Expression	Clase abstracta de expresión	24
HULK.Function	Representa las expresiones de llamado de función	25
HULK.FunctionDeclaration	Representa las expresiones de declaraciones de funciones	27
HULK.HulkErrors	Representa la clase abstracta de errores en HULK	29
HULK.HulkExpression	Clase inicializadora , evalúa todo tipo de expresiones HULK	30
HULK.IncorrectBinaryExpression	Representa el error de operadores binarios	31
HULK.IncorrectOperator	Representa el error de operador unario incorrecto	32
HULK.Let_in	Representa las expresiones let-in	33

HULK.Lexer	
Representa la clase que tokeniza la expresión	35
HULK.LexicalError	
Representa la clase de errores léxicos	38
HULK.MathExpressions	
Representa las expresiones matemáticas básicas	39
HULK.MultiplyExpression	
Representa las expresiones binarias de $*$, $/$, $%$	41
HULK.PowerExpression	
Representa las expresiones binarias de $^$	43
HULK.Print	
Representa la función de Print	46
HULK.Program	
.	48
HULK.SemanticError	
Representa la clase abstracta de errores semánticos	48
HULK.SumExpression	
Representa las expresiones binarias de $+$ y $-$	49
HULK.SyntaxError	
Representa la clase de errores sintácticos	51
HULK.UnExpectedToken	
Representa el error de tokens inesperados	52
HULK.Union	
Representa las expresiones binarias de $@$	53

Capítulo 4

Documentación de espacios de nombres

4.1. Referencia del espacio de nombres HULK

Clases

- class [ArgumentsCountError](#)
Representa el error de cantidad de argumentos incorrectos.
- class [ArgumentTypeError](#)
Representa el error de tipo de argumento incorrecto.
- class [Atom](#)
Representa las expresiones atómicas.
- class [Binary_Exprpressions](#)
Clase abstracta de expresiones binarias.
- class [BooleanOperator](#)
Representa las expresiones binarias booleanas de & y | .
- class [Comparison](#)
Representa las expresiones binarias de (< , > , <= , >= , == , !=)
- class [Conditional](#)
Representa las expresiones condicionales if-else.
- class [ConditionalErrors](#)
Representa el error de expresiones condicionales.
- class [DefaultError](#)
Otros errores (Stack overflow , Division by zero , missing ;)
- class [DuplicateArgument](#)
Representa el error de argumentos duplicados.
- class [Expression](#)
Clase abstracta de expresión.
- class [Function](#)
Representa las expresiones de llamado de función.
- class [FunctionDeclaration](#)
Representa las expresiones de declaraciones de funciones.
- class [HulkErrors](#)
Representa la clase abstracta de errores en [HULK](#).
- class [HulkExpression](#)
Clase inicializadora , evalúa todo tipo de expresiones [HULK](#) .
- class [IncorrectBinaryExpression](#)

- Representa el error de operadores binarios.*
- class [IncorrectOperator](#)
 - Representa el error de operador unario incorrecto.*
- class [Let_in](#)
 - Representa las expresiones let-in.*
- class [Lexer](#)
 - Representa la clase que tokeniza la expresión.*
- class [LexicalError](#)
 - Representa la clase de errores léxicos.*
- class [MathExpressions](#)
 - Representa las expresiones matemáticas básicas.*
- class [MultiplyExpression](#)
 - Representa las expresiones binarias de * , / , % .*
- class [PowerExpression](#)
 - Representa las expresiones binarias de ^ .*
- class [Print](#)
 - Representa la función de Print .*
- class [Program](#)
- class [SemanticError](#)
 - Representa la clase abstracta de errores semánticos.*
- class [SumExpression](#)
 - Representa las expresiones binarias de + y - .*
- class [SyntaxError](#)
 - Representa la clase de errores sintácticos.*
- class [UnexpectedToken](#)
 - Representa el error de tokens inesperados.*
- class [Union](#)
 - Representa las expresiones binarias de .*

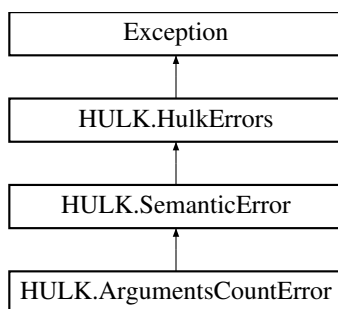
Capítulo 5

Documentación de clases

5.1. Referencia de la clase HULK.ArgumentsCountError

Representa el error de cantidad de argumentos incorrectos.

Diagrama de herencia de HULK.ArgumentsCountError



Métodos públicos

- **ArgumentsCountError** (string functionName, int argumentsIdCount, int argumentsValueCount)
- override void **PrintError** ()

Métodos públicos heredados de [HULK.HulkErrors](#)

- void **PrintError** ()

Otros miembros heredados

Atributos públicos heredados de [HULK.SemanticError](#)

- string? **ProblemType**

5.1.1. Descripción detallada

Representa el error de cantidad de argumentos incorrectos.

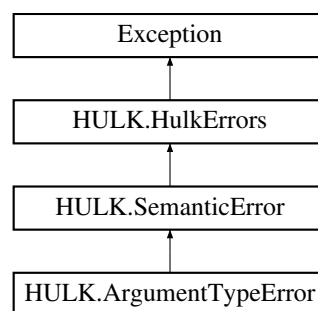
La documentación de esta clase está generada del siguiente archivo:

- Errors.cs

5.2. Referencia de la clase HULK.ArgumentTypeError

Representa el error de tipo de argumento incorrecto.

Diagrama de herencia de HULK.ArgumentTypeError



Métodos públicos

- **ArgumentTypeError** (string expectedToken, string badToken, string? functionName=null)
- override void **PrintError** ()

Métodos públicos heredados de [HULK.HulkErrors](#)

- void **PrintError** ()

Atributos públicos

- string **expectedToken**
- string **badToken**

Atributos públicos heredados de [HULK.SemanticError](#)

- string? **ProblemType**

5.2.1. Descripción detallada

Representa el error de tipo de argumento incorrecto.

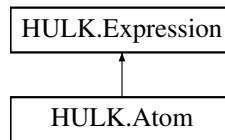
La documentación de esta clase está generada del siguiente archivo:

- Errors.cs

5.3. Referencia de la clase HULK.Atom

Representa las expresiones atómicas.

Diagrama de herencia de HULK.Atom



Métodos públicos

- override void **Analyze** ()
Analiza el tipo de la expresión atómica .
- override void **Evaluate** ()
Evalúa la expresión atómica .

Métodos públicos heredados de HULK.Expression

- object **GetValue** ()
- string **GetExpType** ()
- void **Evaluate** ()
- void **Analyze** ()

Otros miembros heredados

Métodos públicos estáticos heredados de HULK.Expression

- static void **Next** ()
Mueve el índice al próximo token.
- static string **ActualToken** ()
Retorna el token actual.

Atributos públicos heredados de HULK.Expression

- object? **value**
- string? **type**

Atributos públicos estáticos heredados de HULK.Expression

- const string **NumberType** = "number"
- const string **StringType** = "string"
- const string **BooleanType** = "boolean"
- const string **InferenceType** = "inference"

5.3.1. Descripción detallada

Representa las expresiones atómicas.

5.3.2. Documentación de funciones miembro

5.3.2.1. Analyze()

```
override void HULK.Atom.Analyze ( )
```

Analiza el tipo de la expresión atómica .

Excepciones

<i>ArgumentTypeError</i>	Lanza error si left o right no son del tipo correcto y son argumentos de función
<i>IncorrectOperator</i>	Lanza error si el operador unario (! , -) no es válido para el tipo de la expresión
<i>SyntaxError</i>	Lanza error de sintaxis
<i>UnexpectedToken</i>	Lanza error si no encuentra un siguiente de la expresión o es un token no válido

5.3.2.2. Evaluate()

```
override void HULK.Atom.Evaluate ( )
```

Evalúa la expresión atómica .

Excepciones

<i>ArgumentTypeError</i>	Lanza error si left o right no son del tipo correcto y son argumentos de función
<i>IncorrectOperator</i>	Lanza error si el operador unario (! , -) no es válido para el tipo de la expresión
<i>SyntaxError</i>	Lanza error de sintaxis
<i>UnexpectedToken</i>	Lanza error si no encuentra un siguiente de la expresión o es un token no válido

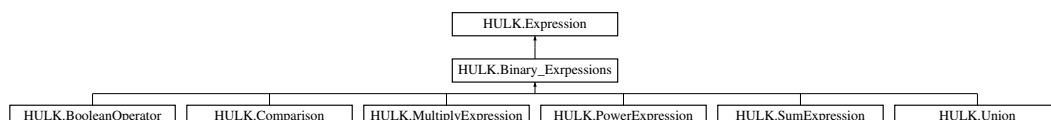
La documentación de esta clase está generada del siguiente archivo:

- Atoms.cs

5.4. Referencia de la clase HULK.Binary_Exprpressions

Clase abstracta de expresiones binarias.

Diagrama de herencia de HULK.Binary_Exprpressions



Métodos públicos

- object **Operation** (object left, string operatorToken, object right)
- void **CatchArgumentTypeError** (string idLeft, string leftType, string idRight, string rightType, string expectedType)

Encuentra errores de argumentos de funciones.

Métodos públicos heredados de HULK.Expression

- object **GetValue** ()
- string **GetExpType** ()
- void **Evaluate** ()
- void **Analyze** ()

Atributos públicos

- string? **idLeft**
- string? **idRight**

Atributos públicos heredados de HULK.Expression

- object? **value**
- string? **type**

Otros miembros heredados

Métodos públicos estáticos heredados de HULK.Expression

- static void **Next** ()
Mueve el índice al próximo token.
- static string **ActualToken** ()
Retorna el token actual.

Atributos públicos estáticos heredados de HULK.Expression

- const string **NumberType** = "number"
- const string **StringType** = "string"
- const string **BooleanType** = "boolean"
- const string **InferenceType** = "inference"

5.4.1. Descripción detallada

Clase abstracta de expresiones binarias.

5.4.2. Documentación de funciones miembro

5.4.2.1. CatchArgumentTypeError()

```
void HULK.Binary_Exprpressions.CatchArgumentTypeError (
    string idLeft,
    string leftType,
    string idRight,
    string rightType,
    string expectedType )
```

Encuentra errores de argumentos de funciones.

Parámetros

<i>idLeft</i>	Token que representa al valor de la izquierda
<i>leftType</i>	Tipo del valor de la expresión de la izquierda
<i>iDRight</i>	Token que representa al valor de la derecha
<i>rightType</i>	Tipo del valor de la expresión de la derecha
<i>expectedType</i>	Tipo de valor esperado

Excepciones

<i>ArgumentTypeError</i>	Lanza error si left o right no son del tipo correcto y son argumentos de función
--------------------------	--

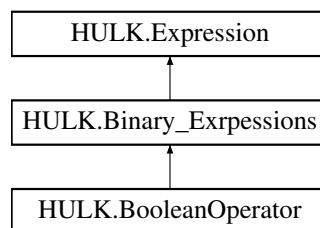
La documentación de esta clase está generada del siguiente archivo:

- Expressions.cs

5.5. Referencia de la clase HULK.BooleanOperator

Representa las expresiones binarias booleanas de & y | .

Diagrama de herencia de HULK.BooleanOperator



Métodos públicos

- override object **Operation** (object left, string operatorToken, object right)
Operación de la clase BooleanOperator.
- override void **Analyze** ()
Analiza la expresión , si hay un token de (& , |) , verifica que left y right sean de tipo boolean y define el tipo de la expresión actualizando left hasta que encuentre un siguiente de la expresión .
- override void **Evaluate** ()
Evalúa la expresión left , si hay un operador (& , |) , Evalúa la expresión right y actualiza el valor de left con el valor de la operación entre el valor de left y right.

Métodos públicos heredados de HULK.Binary_Expressions

- object **Operation** (object left, string operatorToken, object right)
- void **CatchArgumentTypeError** (string idLeft, string leftType, string iDRight, string rightType, string expectedType)
Encuentra errores de argumentos de funciones.

Métodos públicos heredados de [HULK.Expression](#)

- object **GetValue** ()
- string **GetExpType** ()
- void **Evaluate** ()
- void **Analyze** ()

Atributos públicos

- [Expression](#) **left** = new [Comparison](#)()
- [Expression](#) **right** = new [Comparison](#)()

Atributos públicos heredados de [HULK.Binary_Expressions](#)

- string? **iDLeft**
- string? **iDRight**

Atributos públicos heredados de [HULK.Expression](#)

- object? **value**
- string? **type**

Otros miembros heredados

Métodos públicos estáticos heredados de [HULK.Expression](#)

- static void **Next** ()
Mueve el índice al próximo token.
- static string [ActualToken](#) ()
Retorna el token actual.

Atributos públicos estáticos heredados de [HULK.Expression](#)

- const string **NumberType** = "number"
- const string **StringType** = "string"
- const string **BooleanType** = "boolean"
- const string **InferenceType** = "inference"

5.5.1. Descripción detallada

Representa las expresiones binarias booleanas de & y | .

5.5.2. Documentación de funciones miembro

5.5.2.1. [Analyze](#)()

```
override void HULK.BooleanOperator.Analyze ( )
```

Analiza la expresión , si hay un token de (& , |) , verifica que left y right sean de tipo boolean y define el tipo de la expresión actualizando left hasta que encuentre un siguiente de la expresión .

Excepciones

<i>IncorrectBinaryExpression</i>	Lanza la excepción si left o right no son de tipo boolean
<i>UnexpectedToken</i>	Lanza la excepción si no encuentra un siguiente de la expresión

5.5.2.2. Evaluate()

```
override void HULK.BooleanOperator.Evaluate ( )
```

Evalúa la expresión left , si hay un operador (& , |) , Evalúa la expresión right y actualiza el valor de left con el valor de la operación entre el valor de left y right.

Excepciones

<i>IncorrectBinaryExpression</i>	Lanza la excepción si left o right no son de tipo boolean en tiempo de ejecución
<i>UnexpectedToken</i>	Lanza la excepción si no encuentra un siguiente de la expresión

5.5.2.3. Operation()

```
override object HULK.BooleanOperator.Operation (
    object left,
    string operatorToken,
    object right )
```

Operación de la clase BooleanOperator.

Parámetros

<i>left</i>	Representa el valor de la expresión de la izquierda
<i>operatorToken</i>	Operador de la expresión binaria (& ,)
<i>right</i>	Representa el valor de la expresión de la derecha

Devuelve

Retorna la operación entre left y right (bool)

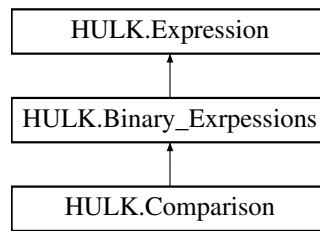
La documentación de esta clase está generada del siguiente archivo:

- Booleans.cs

5.6. Referencia de la clase HULK.Comparison

Representa las expresiones binarias de (< , > , <= , >= , == , !=)

Diagrama de herencia de HULK.Comparison



Métodos públicos

- override void [Analyze](#) ()
Analiza la expresión , si hay un token de (< , > , <= , >= , == , !=) , verifica que left y right sean de tipo number (< , > , <= , >=) o del mismo tipo (== , !=) y define el tipo de la expresión actualizando left hasta que encuentre un siguiente de la expresión .
- override object [Operation](#) (object left, string operatorToken, object right)
Operación de la clase Comparison.
- override void [Evaluate](#) ()
Evalúa la expresión left , si hay un operador (< , > , <= , >= , == , !=) , Evalúa la expresión right y actualiza el valor de left con el valor de la operación entre el valor de left y right.

Métodos públicos heredados de [HULK.Binary_Expressions](#)

- object **Operation** (object left, string operatorToken, object right)
- void [CatchArgumentTypeError](#) (string idLeft, string leftType, string idRight, string rightType, string expectedType)
Encuentra errores de argumentos de funciones.

Métodos públicos heredados de [HULK.Expression](#)

- object **GetValue** ()
- string **GetExpType** ()
- void **Evaluate** ()
- void **Analyze** ()

Atributos públicos

- [Expression](#) **left** = new [SumExpression](#)()
- [Expression](#) **right** = new [SumExpression](#)()

Atributos públicos heredados de [HULK.Binary_Expressions](#)

- string? **idLeft**
- string? **idRight**

Atributos públicos heredados de [HULK.Expression](#)

- object? **value**
- string? **type**

Otros miembros heredados

Métodos públicos estáticos heredados de **HULK.Expression**

- static void **Next** ()
Mueve el índice al próximo token.
- static string **ActualToken** ()
Retorna el token actual.

Atributos públicos estáticos heredados de **HULK.Expression**

- const string **NumberType** = "number"
- const string **StringType** = "string"
- const string **BooleanType** = "boolean"
- const string **InferenceType** = "inference"

5.6.1. Descripción detallada

Representa las expresiones binarias de (< , > , <= , >= , == , !=)

5.6.2. Documentación de funciones miembro

5.6.2.1. Analyze()

```
override void HULK.Comparison.Analyze ( )
```

Analiza la expresión , si hay un token de (< , > , <= , >= , == , !=) , verifica que left y right sean de tipo number (< , > , <= , >=) o del mismo tipo (== , !=) y define el tipo de la expresión actualizando left hasta que encuentre un siguiente de la expresión .

Excepciones

<i>IncorrectBinaryExpression</i>	Lanza la excepción si left o right no son del mismo tipo en tiempo de ejecución
<i>ArgumentTypeError</i>	Lanza error si left o right no son del tipo correcto y son argumentos de función
<i>UnexpectedToken</i>	Lanza la excepción si no encuentra un siguiente de la expresión

5.6.2.2. Evaluate()

```
override void HULK.Comparison.Evaluate ( )
```

Evalúa la expresión left , si hay un operador (< , > , <= , >= , == , !=) , Evalúa la expresión right y actualiza el valor de left con el valor de la operación entre el valor de left y right.

Excepciones

<i>IncorrectBinaryExpression</i>	Lanza la excepción si left o right no son del mismo tipo en tiempo de ejecución
<i>ArgumentTypeError</i>	Lanza error si left o right no son del tipo correcto y son argumentos de función
<i>UnexpectedToken</i>	Lanza la excepción si no encuentra un siguiente de la expresión

5.6.2.3. Operation()

```
override object HULK.Comparison.Operation (
    object left,
    string operatorToken,
    object right )
```

Operación de la clase Comparison.

Parámetros

<i>left</i>	Representa el valor de la expresión de la izquierda
<i>operatorToken</i>	Operador de la expresión binaria (< , > , <= , >= , == , !=)
<i>right</i>	Representa el valor de la expresión de la derecha

Devuelve

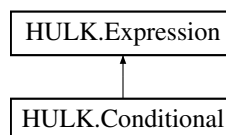
La documentación de esta clase está generada del siguiente archivo:

- Booleans.cs

5.7. Referencia de la clase HULK.Conditional

Representa las expresiones condicionales if-else.

Diagrama de herencia de HULK.Conditional



Métodos públicos

- override void [Analyze](#) ()
Analiza que sea correcta la sintaxis de if-else.
- override void [Evaluate](#) ()
Evalúa la expresión if-else , dándole valor a la expresión.

Métodos públicos heredados de [HULK.Expression](#)

- object **GetValue** ()
- string **GetExpType** ()
- void **Evaluate** ()
- void **Analyze** ()

Otros miembros heredados

Métodos públicos estáticos heredados de [HULK.Expression](#)

- static void **Next** ()
Mueve el índice al próximo token.
- static string **ActualToken** ()
Retorna el token actual.

Atributos públicos heredados de [HULK.Expression](#)

- object? **value**
- string? **type**

Atributos públicos estáticos heredados de [HULK.Expression](#)

- const string **NumberType** = "number"
- const string **StringType** = "string"
- const string **BooleanType** = "boolean"
- const string **InferenceType** = "inference"

5.7.1. Descripción detallada

Representa las expresiones condicionales if-else.

5.7.2. Documentación de funciones miembro

5.7.2.1. Analyze()

```
override void HULK.Conditional.Analyze ( )
```

Analiza que sea correcta la sintaxis de if-else.

Excepciones

<i>ConditionalErrors</i>	Lanza error si la expresión de condición no es booleana
<i>SyntaxError</i>	Error en la sintaxis en la expresión

5.7.2.2. Evaluate()

```
override void HULK.Conditional.Evaluate ( )
```

Evalúa la expresión if-else , dándole valor a la expresión.

Excepciones

<i>ConditionalErrors</i>	Lanza error si la expresión de condición no es booleana
<i>SyntaxError</i>	Error de sintaxis en la expresión

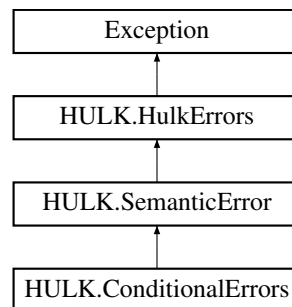
La documentación de esta clase está generada del siguiente archivo:

- Conditionals.cs

5.8. Referencia de la clase HULK.ConditionalErrors

Representa el error de expresiones condicionales.

Diagrama de herencia de HULK.ConditionalErrors

**Métodos públicos**

- **ConditionalErrors** (string badToken)
- override void **PrintError** ()

Métodos públicos heredados de [HULK.HulkErrors](#)

- void **PrintError** ()

Atributos públicos

- string **badToken**

Atributos públicos heredados de [HULK.SemanticError](#)

- string? **ProblemType**

5.8.1. Descripción detallada

Representa el error de expresiones condicionales.

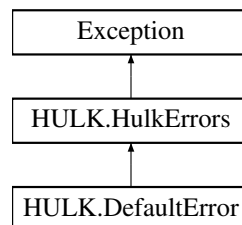
La documentación de esta clase está generada del siguiente archivo:

- Errors.cs

5.9. Referencia de la clase HULK.DefaultError

Otros errores (Stack overflow , Division by zero , missing ;)

Diagrama de herencia de HULK.DefaultError



Métodos públicos

- **DefaultError** (string ProblemType)
- **DefaultError** (string ProblemType, string functionName)
- override void **PrintError** ()

Métodos públicos heredados de [HULK.HulkErrors](#)

- void **PrintError** ()

Atributos públicos

- string **ProblemType**
- string? **functionName**

5.9.1. Descripción detallada

Otros errores (Stack overflow , Division by zero , missing ;)

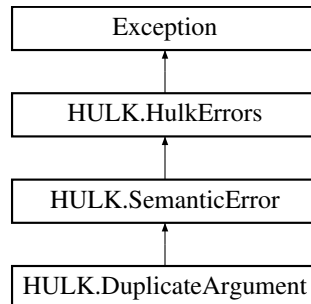
La documentación de esta clase está generada del siguiente archivo:

- Errors.cs

5.10. Referencia de la clase HULK.DuplicateArgument

Representa el error de argumentos duplicados.

Diagrama de herencia de HULK.DuplicateArgument



Métodos públicos

- **DuplicateArgument** (string badToken)
- override void **PrintError** ()

Métodos públicos heredados de [HULK.HulkErrors](#)

- void **PrintError** ()

Otros miembros heredados

Atributos públicos heredados de [HULK.SemanticError](#)

- string? **ProblemType**

5.10.1. Descripción detallada

Representa el error de argumentos duplicados.

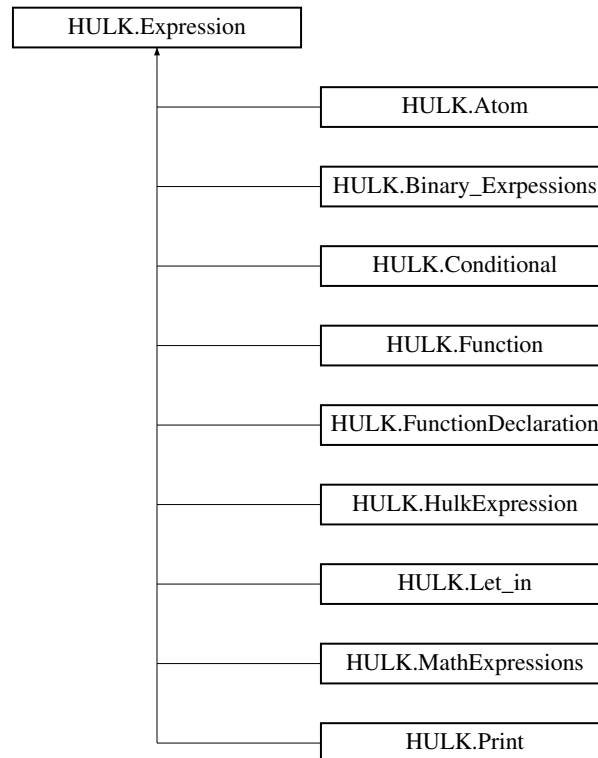
La documentación de esta clase está generada del siguiente archivo:

- Errors.cs

5.11. Referencia de la clase HULK.Expression

Clase abstracta de expresión.

Diagrama de herencia de HULK.Expression



Métodos públicos

- object **GetValue** ()
- string **GetExpType** ()
- void **Evaluate** ()
- void **Analyze** ()

Métodos públicos estáticos

- static void **Next** ()
Mueve el índice al próximo token.
- static string **ActualToken** ()
Retorna el token actual.

Atributos públicos

- object? **value**
- string? **type**

Atributos públicos estáticos

- const string **NumberType** = "number"
- const string **StringType** = "string"
- const string **BooleanType** = "boolean"
- const string **InferenceType** = "inference"

5.11.1. Descripción detallada

Clase abstracta de expresión.

5.11.2. Documentación de funciones miembro

5.11.2.1. ActualToken()

```
static string HULK.Expression.ActualToken ( ) [static]
```

Retorna el token actual.

Devuelve

Token actual

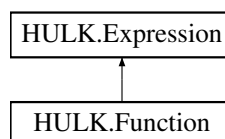
La documentación de esta clase está generada del siguiente archivo:

- Expressions.cs

5.12. Referencia de la clase HULK.Function

Representa las expresiones de llamado de función.

Diagrama de herencia de HULK.Function



Métodos públicos

- **Function** (List< string > argumentsId, List< string > functionExpression, string functionName)
- override void [Analyze](#) ()
Analiza que el llamado de la función sea correcto.
- override void [Evaluate](#) ()
Evalúa la función , devolviendo el valor de su cuerpo con los argumentos dados.

Métodos públicos heredados de [HULK.Expression](#)

- object **GetValue** ()
- string **GetExpType** ()
- void **Evaluate** ()
- void **Analyze** ()

Métodos públicos estáticos

- static void [CheckfunctionCall](#) (string functionName)
Analiza que el llamado de la función sea correcto sin evaluarla.

Métodos públicos estáticos heredados de [HULK.Expression](#)

- static void **Next** ()
Mueve el índice al próximo token.
- static string [ActualToken](#) ()
Retorna el token actual.

Atributos públicos

- List< string > **argumentsId** = new List<string>()
- List< object > **argumentsValue** = new List<object>()
- List< string > **functionExpression** = new List<string>()

Atributos públicos heredados de [HULK.Expression](#)

- object? **value**
- string? **type**

Atributos públicos estáticos

- static Dictionary< string, object > **functionsId** = new Dictionary<string, object>()

Atributos públicos estáticos heredados de [HULK.Expression](#)

- const string **NumberType** = "number"
- const string **StringType** = "string"
- const string **BooleanType** = "boolean"
- const string **InferenceType** = "inference"

5.12.1. Descripción detallada

Representa las expresiones de llamado de función.

5.12.2. Documentación de funciones miembro

5.12.2.1. Analyze()

```
override void HULK.Function.Analyze ( )
```

Analiza que el llamado de la función sea correcto.

Excepciones

<i>SyntaxError</i>	Lanza error de sintaxis
<i>ArgumentsCountError</i>	Lanza error si la cantidad de argumentos esperados son incorrectos

5.12.2.2. CheckfunctionCall()

```
static void HULK.Function.CheckfunctionCall (
    string functionName ) [static]
```

Analiza que el llamado de la función sea correcto sin evaluarla.

Parámetros

<i>functionName</i>	Nombre de la función
---------------------	----------------------

Excepciones

<i>ArgumentsCountError</i>	Lanza error si la cantidad de argumentos esperados son incorrectos
----------------------------	--

5.12.2.3. Evaluate()

```
override void HULK.Function.Evaluate ( )
```

Evalúa la función , devolviendo el valor de su cuerpo con los argumentos dados.

Excepciones

<i>DefaultError</i>	Stack Overflow
<i>ArgumentsCountError</i>	La cantidad de argumentos esperados son incorrectos
<i>ArgumentTypeError</i>	
<i>SyntaxError</i>	Lanza error de sintaxis

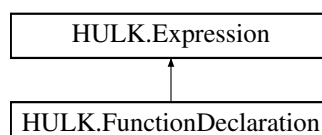
La documentación de esta clase está generada del siguiente archivo:

- Functions.cs

5.13. Referencia de la clase HULK.FunctionDeclaration

Representa las expresiones de declaraciones de funciones.

Diagrama de herencia de HULK.FunctionDeclaration



Métodos públicos

- override void [Analyze](#) ()
Analiza que sea correcta la declaración de la función .
- override void [Evaluate](#) ()
Una vez analizada la expresión , guarda la función en functionStore.

Métodos públicos heredados de [HULK.Expression](#)

- object [GetValue](#) ()
- string [GetExpType](#) ()
- void [Evaluate](#) ()
- void [Analyze](#) ()

Atributos públicos

- List< string > **functionArguments** = new List<string>()

Atributos públicos heredados de [HULK.Expression](#)

- object? **value**
- string? **type**

Atributos públicos estáticos

- static Dictionary< string, [Function](#) > **functionStore** = new Dictionary<string, [Function](#)>()
- static Dictionary< string, int > **functionStack** = new Dictionary<string, int>()
- static int **functionArgumentsCount** = 0
- static Dictionary< string, string > **functionSIdInference** = new Dictionary<string, string>()

Atributos públicos estáticos heredados de [HULK.Expression](#)

- const string **NumberType** = "number"
- const string **StringType** = "string"
- const string **BooleanType** = "boolean"
- const string **InferenceType** = "inference"

Otros miembros heredados

Métodos públicos estáticos heredados de [HULK.Expression](#)

- static void [Next](#) ()
Mueve el índice al próximo token.
- static string [ActualToken](#) ()
Retorna el token actual.

5.13.1. Descripción detallada

Representa las expresiones de declaraciones de funciones.

5.13.2. Documentación de funciones miembro

5.13.2.1. [Analyze](#)()

```
override void HULK.FunctionDeclaration.Analyze ( )
```

Analiza que sea correcta la declaración de la función .

Excepciones

<i>SyntaxError</i>	Lanza error de sintaxis
<i>DuplicateArgument</i>	Lanza error si existen argumentos de la función duplicados

5.13.2.2. Evaluate()

```
override void HULK.FunctionDeclaration.Evaluate ( )
```

Una vez analizada la expresión , guarda la función en functionStore.

Excepciones

<i>SyntaxError</i>	Lanza error de sintaxis
<i>DuplicateArgument</i>	Lanza error si existen argumentos de la función duplicados

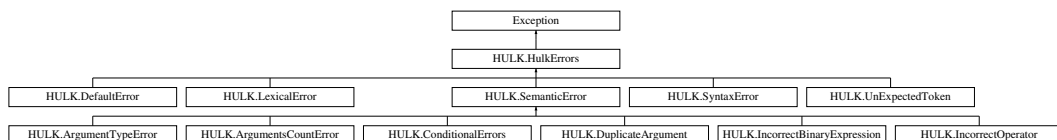
La documentación de esta clase está generada del siguiente archivo:

- Functions.cs

5.14. Referencia de la clase HULK.HulkErrors

Representa la clase abstracta de errores en [HULK](#).

Diagrama de herencia de HULK.HulkErrors



Métodos públicos

- void **PrintError** ()

5.14.1. Descripción detallada

Representa la clase abstracta de errores en [HULK](#).

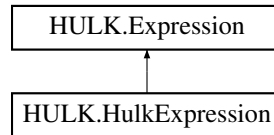
La documentación de esta clase está generada del siguiente archivo:

- Errors.cs

5.15. Referencia de la clase HULK.HulkExpression

Clase inicializadora , evalúa todo tipo de expresiones [HULK](#) .

Diagrama de herencia de HULK.HulkExpression



Métodos públicos

- override void **Analyze** ()
- override void **Evaluate** ()

Métodos públicos heredados de [HULK.Expression](#)

- object **GetValue** ()
- string **GetExpType** ()
- void **Evaluate** ()
- void **Analyze** ()

Otros miembros heredados

Métodos públicos estáticos heredados de [HULK.Expression](#)

- static void **Next** ()
Mueve el índice al próximo token.
- static string **ActualToken** ()
Retorna el token actual.

Atributos públicos heredados de [HULK.Expression](#)

- object? **value**
- string? **type**

Atributos públicos estáticos heredados de [HULK.Expression](#)

- const string **NumberType** = "number"
- const string **StringType** = "string"
- const string **BooleanType** = "boolean"
- const string **InferenceType** = "inference"

5.15.1. Descripción detallada

Clase inicializadora , evalúa todo tipo de expresiones [HULK](#) .

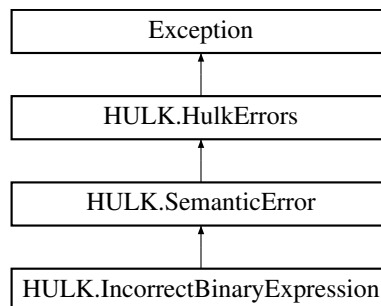
La documentación de esta clase está generada del siguiente archivo:

- Expressions.cs

5.16. Referencia de la clase HULK.IncorrectBinaryExpression

Representa el error de operadores binarios.

Diagrama de herencia de HULK.IncorrectBinaryExpression



Métodos públicos

- **IncorrectBinaryExpression** (string operatorProblem, string leftTokenType, string rightTokenType)
- override void **PrintError** ()

Métodos públicos heredados de [HULK.HulkErrors](#)

- void **PrintError** ()

Atributos públicos

- string **operatorProblem**
- string **leftTokenType**
- string **rightTokenType**

Atributos públicos heredados de [HULK.SemanticError](#)

- string? **ProblemType**

5.16.1. Descripción detallada

Representa el error de operadores binarios.

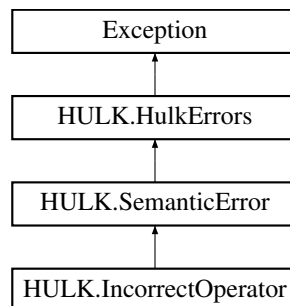
La documentación de esta clase está generada del siguiente archivo:

- Errors.cs

5.17. Referencia de la clase HULK.IncorrectOperator

Representa el error de operador unario incorrecto.

Diagrama de herencia de HULK.IncorrectOperator



Métodos públicos

- **IncorrectOperator** (string badToken, string operatorProblem, string expectedToken)
- override void **PrintError** ()

Métodos públicos heredados de [HULK.HulkErrors](#)

- void **PrintError** ()

Atributos públicos

- string **operatorProblem**
- string **badToken**
- string **expectedToken**

Atributos públicos heredados de [HULK.SemanticError](#)

- string? **ProblemType**

5.17.1. Descripción detallada

Representa el error de operador unario incorrecto.

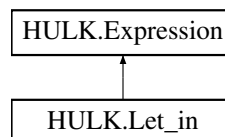
La documentación de esta clase está generada del siguiente archivo:

- Errors.cs

5.18. Referencia de la clase HULK.Let_in

Representa las expresiones let-in.

Diagrama de herencia de HULK.Let_in



Métodos públicos

- override void [Analyze](#) ()
Analiza que no hayan errores en la expresión.
- override void [Evaluate](#) ()
Evalúa la expresión let-in , dándole valor a la expresión .

Métodos públicos heredados de [HULK.Expression](#)

- object **GetValue** ()
- string **GetExpType** ()
- void **Evaluate** ()
- void **Analyze** ()

Atributos públicos estáticos

- static Dictionary< string, object > **idStore** = new Dictionary<string, object>()

Atributos públicos estáticos heredados de [HULK.Expression](#)

- const string **NumberType** = "number"
- const string **StringType** = "string"
- const string **BooleanType** = "boolean"
- const string **InferenceType** = "inference"

Otros miembros heredados

Métodos públicos estáticos heredados de [HULK.Expression](#)

- static void **Next** ()
Mueve el índice al próximo token.
- static string **ActualToken** ()
Retorna el token actual.

Atributos públicos heredados de [HULK.Expression](#)

- object? **value**
- string? **type**

5.18.1. Descripción detallada

Representa las expresiones let-in.

5.18.2. Documentación de funciones miembro

5.18.2.1. Analyze()

```
override void HULK.Let_in.Analyze ( )
```

Analiza que no hayan errores en la expresión.

Excepciones

<i>SyntaxError</i>	Lanza error de sintaxis
--------------------	-------------------------

5.18.2.2. Evaluate()

```
override void HULK.Let_in.Evaluate ( )
```

Evalúa la expresión let-in , dándole valor a la expresión .

Excepciones

<i>SyntaxError</i>	Lanza error de sintaxis
--------------------	-------------------------

La documentación de esta clase está generada del siguiente archivo:

- Let-in.cs

5.19. Referencia de la clase HULK.Lexer

Representa la clase que tokeniza la expresión.

Métodos públicos estáticos

- static void **Restart** ()
Reinicia las variables para una nueva línea.
- static void **TokenizeInput** (string input)
Tokeniza la expresión.
- static bool **IsNumber** (string Token)
Determina si la expresión es un número.
- static bool **IsString** (string Token)
Determina si la expresión es un string.
- static bool **IsBoolean** (string Token)
Determina si la expresión es un boolean.
- static bool **IsID** (string Token)
Determina si la expresión es un ID válido.
- static string **TokenType** (object Token)
Determina el tipo de la expresión.
- static string **GetIncorrectToken** (string a, string b, string expectedToken)
Encuentra cual es el token incorrecto de una expresión binaria.

Atributos públicos estáticos

- static List< string > **Tokens** = new List<string>()
- static List< object > **ConsolePrints** = new List<object>()
- static string **decimalSeparator** = nfi.NumberDecimalSeparator
- static int **index** = 0
- static List< string > **Key_Words**

5.19.1. Descripción detallada

Representa la clase que tokeniza la expresión.

5.19.2. Documentación de funciones miembro

5.19.2.1. GetIncorrectToken()

```
static string HULK.Lexer.GetIncorrectToken (
    string a,
    string b,
    string expectedToken ) [static]
```

Encuentra cual es el token incorrecto de una expresión binaria.

Parámetros

<i>a</i>	Tipo del valor de la expresión izquierda
<i>b</i>	Tipo del valor de la expresión derecha
<i>expectedToken</i>	Tipo de valor esperado

Devuelve

Tipo de expresión incorrecta

5.19.2.2. IsBoolean()

```
static bool HULK.Lexer.IsBoolean (  
    string Token ) [static]
```

Determina si la expresión es un boolean.

Parámetros

<i>Token</i>	Token actual
--------------	--------------

Devuelve

Es o no un boolean

5.19.2.3. IsID()

```
static bool HULK.Lexer.IsID (  
    string Token ) [static]
```

Determina si la expresión es un ID válido.

Parámetros

<i>Token</i>	Token actual
--------------	--------------

Devuelve

Es o no un ID válido

5.19.2.4. IsNumber()

```
static bool HULK.Lexer.IsNumber (  
    string Token ) [static]
```

Determina si la expresión es un número.

Parámetros

<i>Token</i>	Token actual
--------------	--------------

Devuelve

Es o no un número

5.19.2.5. IsString()

```
static bool HULK.Lexer.IsString (  
    string Token ) [static]
```

Determina si la expresión es un string.

Parámetros

<i>Token</i>	Token actual
--------------	--------------

Devuelve

Es o no un string

5.19.2.6. TokenizeInput()

```
static void HULK.Lexer.TokenizeInput (  
    string input ) [static]
```

Tokeniza la expresión.

Parámetros

<i>input</i>	Input del usuario
--------------	-------------------

Excepciones

<i>LexicalError</i>	Lanza error léxico
---------------------	--------------------

5.19.2.7. TokenType()

```
static string HULK.Lexer.TokenType (  
    object Token ) [static]
```

Determina el tipo de la expresión.

Parámetros

<i>Token</i>	Token actual
--------------	--------------

Devuelve

Tipo de la expresión

5.19.3. Documentación de datos miembro**5.19.3.1. Key_Words**

```
List<string> HULK.Lexer.Key_Words [static]
```

Valor inicial:

```
= new List<string>()
{ "print" , "let " , "in" , "function" , "if" , "else" , "true" , "false" , "sin" , "cos" , "sqrt" ,
  "rand" , "exp" , "log" , "PI" , "E" }
```

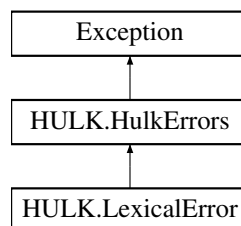
La documentación de esta clase está generada del siguiente archivo:

- Lexer.cs

5.20. Referencia de la clase HULK.LexicalError

Representa la clase de errores léxicos.

Diagrama de herencia de HULK.LexicalError

**Métodos públicos**

- **LexicalError** (string BadToken)
- override void **PrintError** ()

Métodos públicos heredados de [HULK.HulkErrors](#)

- void **PrintError** ()

Atributos públicos

- string **BadToken**

5.20.1. Descripción detallada

Representa la clase de errores léxicos.

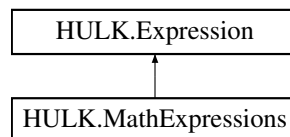
La documentación de esta clase está generada del siguiente archivo:

- Errors.cs

5.21. Referencia de la clase HULK.MathExpressions

Representa las expresiones matemáticas básicas.

Diagrama de herencia de HULK.MathExpressions



Métodos públicos

- **MathExpressions** (string mathExp)
- override void **Analyze** ()
Analiza que el llamado de función sea correcto.
- override void **Evaluate** ()
Evalúa el llamado de la función.
- void **sqrt** ()
- void **sin** ()
- void **cos** ()
- void **exp** ()
- void **log** ()
- void **rand** ()

Métodos públicos heredados de [HULK.Expression](#)

- object **GetValue** ()
- string **GetExpType** ()
- void **Evaluate** ()
- void **Analyze** ()

Atributos públicos

- string **mathExp**

Atributos públicos heredados de [HULK.Expression](#)

- object? **value**
- string? **type**

Atributos públicos estáticos

- static List< string > **MathFunctions** = new List<string>(){ "sin" , "cos" , "sqrt" , "rand" , "exp" , "log" , "PI" , "E" }

Atributos públicos estáticos heredados de [HULK.Expression](#)

- const string **NumberType** = "number"
- const string **StringType** = "string"
- const string **BooleanType** = "boolean"
- const string **InferenceType** = "inference"

Otros miembros heredados

Métodos públicos estáticos heredados de [HULK.Expression](#)

- static void **Next** ()
Mueve el índice al próximo token.
- static string **ActualToken** ()
Retorna el token actual.

5.21.1. Descripción detallada

Representa las expresiones matemáticas básicas.

5.21.2. Documentación de funciones miembro

5.21.2.1. Analyze()

```
override void HULK.MathExpressions.Analyze ( )
```

Analiza que el llamado de función sea correcto.

Excepciones

<i>ArgumentTypeError</i>	Lanza error si el tipo del argumento no es correcto
<i>SyntaxError</i>	Lanza error de syntaxs
<i>ArgumentsCountError</i>	Lanza error si la cantidad de argumentos esperados son incorrectos

5.21.2.2. Evaluate()

```
override void HULK.MathExpressions.Evaluate ( )
```

Evalúa el llamado de la función.

Excepciones

<code>SyntaxError</code>	Lanza error de sintaxis
--------------------------	-------------------------

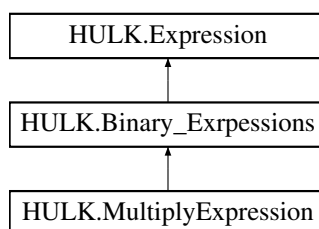
La documentación de esta clase está generada del siguiente archivo:

- MathClass.cs

5.22. Referencia de la clase HULK.MultiplyExpression

Representa las expresiones binarias de $*$, $/$, $%$.

Diagrama de herencia de HULK.MultiplyExpression



Métodos públicos

- override object **Operation** (object left, string operatorToken, object right)
Operación de la clase MultiplyExpression.
- override void **Analyze** ()
Analiza la expresión, si hay un token de ($$, $/$, $%$), verifica que left y right sean de tipo number y define el tipo de la expresión actualizando left hasta que encuentre un siguiente de la expresión ;.*
- override void **Evaluate** ()
Evalúa la expresión left, si hay un operador ($$, $/$, $%$), Evalúa la expresión right y actualiza el valor de left con el valor de la operación entre el valor de left y right.*

Métodos públicos heredados de HULK.Binary_Expressions

- object **Operation** (object left, string operatorToken, object right)
- void **CatchArgumentTypeError** (string idLeft, string leftType, string idRight, string rightType, string expectedType)
Encuentra errores de argumentos de funciones.

Métodos públicos heredados de HULK.Expression

- object **GetValue** ()
- string **GetExpType** ()
- void **Evaluate** ()
- void **Analyze** ()

Atributos públicos

- `Expression left` = new `PowerExpression()`
- `Expression right` = new `PowerExpression()`

Atributos públicos heredados de `HULK.Binary_Expressions`

- `string? iDLeft`
- `string? iDRight`

Atributos públicos heredados de `HULK.Expression`

- `object? value`
- `string? type`

Otros miembros heredados

Métodos públicos estáticos heredados de `HULK.Expression`

- `static void Next ()`
Mueve el índice al próximo token.
- `static string ActualToken ()`
Retorna el token actual.

Atributos públicos estáticos heredados de `HULK.Expression`

- `const string NumberType` = "number"
- `const string StringType` = "string"
- `const string BooleanType` = "boolean"
- `const string InferenceType` = "inference"

5.22.1. Descripción detallada

Representa las expresiones binarias de `*`, `/`, `%`.

5.22.2. Documentación de funciones miembro

5.22.2.1. `Analyze()`

```
override void HULK.MultiplyExpression.Analyze ( )
```

Analiza la expresión, si hay un token de (`*`, `/`, `%`), verifica que `left` y `right` sean de tipo `number` y define el tipo de la expresión actualizando `left` hasta que encuentre un siguiente de la expresión ;.

Excepciones

<i>IncorrectBinaryExpression</i>	Lanza la excepción si left o right no son de tipo number
<i>UnexpectedToken</i>	Lanza la excepción si no encuentra un siguiente de la expresión

5.22.2.2. Evaluate()

```
override void HULK.MultiplyExpression.Evaluate ( )
```

Evalúa la expresión left , si hay un operador (* , / , %) , Evalúa la expresión right y actualiza el valor de left con el valor de la operación entre el valor de left y right.

Excepciones

<i>IncorrectBinaryExpression</i>	Lanza la excepción si left o right no son de tipo number en tiempo de ejecución
<i>UnexpectedToken</i>	Lanza la excepción si no encuentra un siguiente de la expresión

5.22.2.3. Operation()

```
override object HULK.MultiplyExpression.Operation (
    object left,
    string operatorToken,
    object right )
```

Operación de la clase MultiplyExpresión.

Parámetros

<i>left</i>	Representa el valor de la expresión de la izquierda
<i>operatorToken</i>	Operador de la expresión binaria (* , / , %)
<i>right</i>	Representa el valor de la expresión de la derecha

Devuelve

Retorna la operación entre left y right (number)

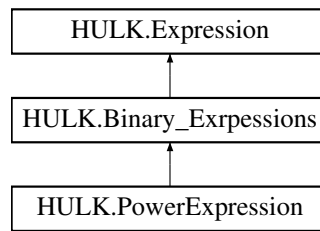
La documentación de esta clase está generada del siguiente archivo:

- Numbers.cs

5.23. Referencia de la clase HULK.PowerExpression

Representa las expresiones binarias de $^$.

Diagrama de herencia de HULK.PowerExpression



Métodos públicos

- override object [Operation](#) (object left, string operatorToken, object right)
- override void [Analyze](#) ()

Analiza la expresión , si hay un token de (^) , verifica que left y right sean de tipo number y define el tipo de la expresión actualizando left hasta que encuentre un siguiente de la expresión .
- override void [Evaluate](#) ()

Evalúa la expresión left , si hay un operador (^) , Evalúa la expresión right y actualiza el valor de left con el valor de la operación entre el valor de left y right.

Métodos públicos heredados de [HULK.Binary_Exprpressions](#)

- object **Operation** (object left, string operatorToken, object right)
- void [CatchArgumentTypeError](#) (string idLeft, string leftType, string idRight, string rightType, string expectedType)

Encuentra errores de argumentos de funciones.

Métodos públicos heredados de [HULK.Expression](#)

- object **GetValue** ()
- string **GetExpType** ()
- void **Evaluate** ()
- void **Analyze** ()

Atributos públicos

- [Expression](#) **left** = new [Atom](#)()
- [Expression](#) **right** = new [Atom](#)()

Atributos públicos heredados de [HULK.Binary_Exprpressions](#)

- string? **idLeft**
- string? **idRight**

Atributos públicos heredados de [HULK.Expression](#)

- object? **value**
- string? **type**

Otros miembros heredados

Métodos públicos estáticos heredados de [HULK.Expression](#)

- static void **Next** ()
Mueve el índice al próximo token.
- static string **ActualToken** ()
Retorna el token actual.

Atributos públicos estáticos heredados de [HULK.Expression](#)

- const string **NumberType** = "number"
- const string **StringType** = "string"
- const string **BooleanType** = "boolean"
- const string **InferenceType** = "inference"

5.23.1. Descripción detallada

Representa las expresiones binarias de \wedge .

5.23.2. Documentación de funciones miembro

5.23.2.1. Analyze()

```
override void HULK.PowerExpression.Analyze ( )
```

Analiza la expresión, si hay un token de (\wedge), verifica que left y right sean de tipo number y define el tipo de la expresión actualizando left hasta que encuentre un siguiente de la expresión.

Excepciones

<i>IncorrectBinaryExpression</i>	Lanza la excepción si left o right no son de tipo number
<i>UnexpectedToken</i>	Lanza la excepción si no encuentra un siguiente de la expresión

5.23.2.2. Evaluate()

```
override void HULK.PowerExpression.Evaluate ( )
```

Evalúa la expresión left, si hay un operador (\wedge), Evalúa la expresión right y actualiza el valor de left con el valor de la operación entre el valor de left y right.

Excepciones

<i>IncorrectBinaryExpression</i>	Lanza la excepción si left o right no son de tipo number en tiempo de ejecución
<i>UnexpectedToken</i>	Lanza la excepción si no encuentra un siguiente de la expresión

5.23.2.3. Operation()

```
override object HULK.PowerExpression.Operation (
    object left,
    string operatorToken,
    object right )
```

Operación de la clase PowerExpression

Parámetros

<i>left</i>	Representa el valor de la expresión de la izquierda
<i>operatorToken</i>	Operador de la expresión binaria (^)
<i>right</i>	Representa el valor de la expresión de la derecha

Devuelve

Retorna la operación entre left y right (number)/returns>

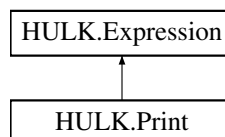
La documentación de esta clase está generada del siguiente archivo:

- Numbers.cs

5.24. Referencia de la clase HULK.Print

Representa la función de Print .

Diagrama de herencia de HULK.Print



Métodos públicos

- override void [Analyze](#) ()
Analiza que no hayan errores de sintaxis en la expresión.
- override void [Evaluate](#) ()
Evalúa la expresión , imprimiendo en pantalla el valor de la expresión.

Métodos públicos heredados de [HULK.Expression](#)

- object [GetValue](#) ()
- string [GetExpType](#) ()
- void [Evaluate](#) ()
- void [Analyze](#) ()

Otros miembros heredados

Métodos públicos estáticos heredados de [HULK.Expression](#)

- static void **Next** ()
Mueve el índice al próximo token.
- static string **ActualToken** ()
Retorna el token actual.

Atributos públicos heredados de [HULK.Expression](#)

- object? **value**
- string? **type**

Atributos públicos estáticos heredados de [HULK.Expression](#)

- const string **NumberType** = "number"
- const string **StringType** = "string"
- const string **BooleanType** = "boolean"
- const string **InferenceType** = "inference"

5.24.1. Descripción detallada

Representa la función de Print .

5.24.2. Documentación de funciones miembro

5.24.2.1. Analyze()

```
override void HULK.Print.Analyze ( )
```

Analiza que no hayan errores de sintaxis en la expresión.

Excepciones

<i>SyntaxError</i>	Lanza error de sintaxis
--------------------	-------------------------

5.24.2.2. Evaluate()

```
override void HULK.Print.Evaluate ( )
```

Evalúa la expresión , imprimiendo en pantalla el valor de la expresión.

Excepciones

<i>SyntaxError</i>	
--------------------	--

La documentación de esta clase está generada del siguiente archivo:

- Functions.cs

5.25. Referencia de la clase HULK.Program

Métodos públicos estáticos

- static void **Main** (string[] args)

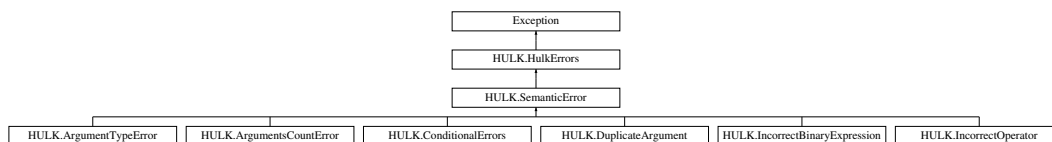
La documentación de esta clase está generada del siguiente archivo:

- Program.cs

5.26. Referencia de la clase HULK.SemanticError

Representa la clase abstracta de errores semánticos.

Diagrama de herencia de HULK.SemanticError



Atributos públicos

- string? **ProblemType**

Otros miembros heredados

Métodos públicos heredados de [HULK.HulkErrors](#)

- void **PrintError** ()

5.26.1. Descripción detallada

Representa la clase abstracta de errores semánticos.

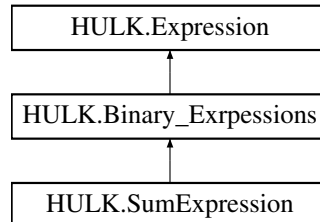
La documentación de esta clase está generada del siguiente archivo:

- Errors.cs

5.27. Referencia de la clase HULK.SumExpression

Representa las expresiones binarias de + y - .

Diagrama de herencia de HULK.SumExpression



Métodos públicos

- override object [Operation](#) (object left, string operatorToken, object right)
Operación de la clase SumExpression.
- override void [Analyze](#) ()
Analiza la expresión , si hay un token de + o - , verifica que left y right sean de tipo number y define el tipo de la expresión actualizando left hasta que encuentre un siguiente de la expresión ;.
- override void [Evaluate](#) ()
Evalúa la expresión left , si hay un operador (+ , -) , Evalúa la expresión right y actualiza el valor de left con el valor de la operación entre el valor de left y right.

Métodos públicos heredados de [HULK.Binary_Exprpressions](#)

- object **Operation** (object left, string operatorToken, object right)
- void [CatchArgumentTypeError](#) (string idLeft, string leftType, string idRight, string rightType, string expectedType)
Encuentra errores de argumentos de funciones.

Métodos públicos heredados de [HULK.Expression](#)

- object **GetValue** ()
- string **GetExpType** ()
- void **Evaluate** ()
- void **Analyze** ()

Atributos públicos

- [Expression](#) left = new [MultiplyExpression](#)()
- [Expression](#) right = new [MultiplyExpression](#)()

Atributos públicos heredados de [HULK.Binary_Exprpressions](#)

- string? **idLeft**
- string? **idRight**

Atributos públicos heredados de [HULK.Expression](#)

- object? **value**
- string? **type**

Otros miembros heredados

Métodos públicos estáticos heredados de [HULK.Expression](#)

- static void **Next** ()
Mueve el índice al próximo token.
- static string **ActualToken** ()
Retorna el token actual.

Atributos públicos estáticos heredados de [HULK.Expression](#)

- const string **NumberType** = "number"
- const string **StringType** = "string"
- const string **BooleanType** = "boolean"
- const string **InferenceType** = "inference"

5.27.1. Descripción detallada

Representa las expresiones binarias de + y - .

5.27.2. Documentación de funciones miembro

5.27.2.1. Analyze()

```
override void HULK.SumExpression.Analyze ( )
```

Analiza la expresión , si hay un token de + o - , verifica que left y right sean de tipo number y define el tipo de la expresión actualizando left hasta que encuentre un siguiente de la expresión ;.

Excepciones

<i>IncorrectBinaryExpression</i>	Lanza la excepción si left o right no son de tipo number
<i>UnexpectedToken</i>	Lanza la excepción si no encuentra un siguiente de la expresión

5.27.2.2. Evaluate()

```
override void HULK.SumExpression.Evaluate ( )
```

Evalúa la expresión left , si hay un operador (+ , -) , Evalúa la expresión right y actualiza el valor de left con el valor de la operación entre el valor de left y right.

Excepciones

<i>IncorrectBinaryExpression</i>	Lanza la excepción si left o right no son de tipo number en tiempo de ejecución
<i>UnexpectedToken</i>	Lanza la excepción si no encuentra un siguiente de la expresión

5.27.2.3. Operation()

```
override object HULK.SumExpression.Operation (
    object left,
    string operatorToken,
    object right )
```

Operación de la clase SumExpression.

Parámetros

<i>left</i>	Representa el valor de la expresión de la izquierda
<i>operatorToken</i>	Operador de la expresión binaria (+ , -)
<i>right</i>	Representa el valor de la expresión de la derecha

Devuelve

Retorna la operación entre left y right (number)

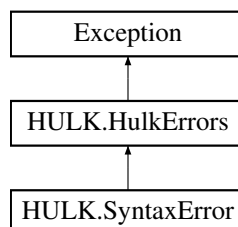
La documentación de esta clase está generada del siguiente archivo:

- Numbers.cs

5.28. Referencia de la clase HULK.SyntaxError

Representa la clase de errores sintácticos.

Diagrama de herencia de HULK.SyntaxError



Métodos públicos

- **SyntaxError** (string Problem, string ProblemType, string ExpressionType, string Token)
- **SyntaxError** (string Token, string ProblemType)
- override void **PrintError** ()

Métodos públicos heredados de [HULK.HulkErrors](#)

- void **PrintError** ()

Atributos públicos

- string? **Problem**
- string **ProblemType**
- string? **ExpressionType**
- string **Token**

5.28.1. Descripción detallada

Representa la clase de errores sintácticos.

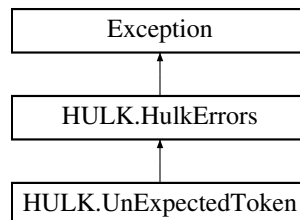
La documentación de esta clase está generada del siguiente archivo:

- Errors.cs

5.29. Referencia de la clase [HULK.UnExpectedToken](#)

Representa el error de tokens inesperados.

Diagrama de herencia de [HULK.UnExpectedToken](#)



Métodos públicos

- **UnExpectedToken** (string BadToken)
- override void **PrintError** ()

Métodos públicos heredados de [HULK.HulkErrors](#)

- void **PrintError** ()

Atributos públicos

- string **BadToken**

5.29.1. Descripción detallada

Representa el error de tokens inesperados.

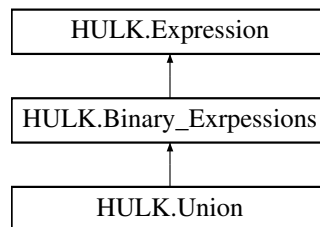
La documentación de esta clase está generada del siguiente archivo:

- Errors.cs

5.30. Referencia de la clase HULK.Union

Representa las expresiones binarias de .

Diagrama de herencia de HULK.Union



Métodos públicos

- override object [Operation](#) (object left, string operatorToken, object right)
Operación de la clase Union (Une como string dos valores)
- override void [Analyze](#) ()
Analiza la expresión left , si hay un token de @ analiza la expresión right y define el tipo como string si no define el tipo de la expresión con el tipo de left.
- override void [Evaluate](#) ()
Evalúa la expresión left , si hay un token de @ evalúa la expresión right y une los valores de left y right como string , definiendo el valor de la expresión.

Métodos públicos heredados de [HULK.Binary_Expressions](#)

- object **Operation** (object left, string operatorToken, object right)
- void [CatchArgumentTypeError](#) (string idLeft, string leftType, string idRight, string rightType, string expectedType)
Encuentra errores de argumentos de funciones.

Métodos públicos heredados de [HULK.Expression](#)

- object **GetValue** ()
- string **GetExpType** ()
- void **Evaluate** ()
- void **Analyze** ()

Atributos públicos

- `Expression left` = new `BooleanOperator()`
- `Expression right` = new `BooleanOperator()`

Atributos públicos heredados de `HULK.Binary_Expressions`

- `string? iDLeft`
- `string? iDRight`

Atributos públicos heredados de `HULK.Expression`

- `object? value`
- `string? type`

Otros miembros heredados

Métodos públicos estáticos heredados de `HULK.Expression`

- `static void Next ()`
Mueve el índice al próximo token.
- `static string ActualToken ()`
Retorna el token actual.

Atributos públicos estáticos heredados de `HULK.Expression`

- `const string NumberType` = "number"
- `const string StringType` = "string"
- `const string BooleanType` = "boolean"
- `const string InferenceType` = "inference"

5.30.1. Descripción detallada

Representa las expresiones binarias de .

5.30.2. Documentación de funciones miembro

5.30.2.1. `Analyze()`

```
override void HULK.Union.Analyze ( )
```

Analiza la expresión `left` , si hay un token de `@` analiza la expresión `right` y define el tipo como `string` si no define el tipo de la expresión con el tipo de `left`.

Excepciones

<i>UnexpectedToken</i>	Lanza error si no encuentra un siguiente de la expresión
------------------------	--

5.30.2.2. Evaluate()

```
override void HULK.Union.Evaluate ( )
```

Evalúa la expresión *left* , si hay un token de *@* evalúa la expresión *right* y une los valores de *left* y *right* como string , definiendo el valor de la expresión.

Excepciones

<i>UnexpectedToken</i>	Lanza error si no encuentra un siguiente de la expresión
------------------------	--

5.30.2.3. Operation()

```
override object HULK.Union.Operation (
    object left,
    string operatorToken,
    object right )
```

Operación de la clase Union (Une como string dos valores)

Parámetros

<i>left</i>	Representa la expresión de la izquierda
<i>operatorToken</i>	Operador de la expresión binaria (<i>@</i>)
<i>right</i>	Representa la expresión de la derecha

Devuelve

Retorna la unión de los valores de *left* y *right* (string)

La documentación de esta clase está generada del siguiente archivo:

- Union.cs

