

Project_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	9
4.1 Referencia del espacio de nombres Hulk	9
4.1.1 Documentación de funciones	11
4.1.1.1 Print()	11
5 Documentación de clases	13
5.1 Referencia de la clase Hulk.Addition	13
5.1.1 Descripción detallada	14
5.2 Referencia de la clase Hulk.Asignment	14
5.2.1 Descripción detallada	15
5.2.2 Documentación de constructores y destructores	15
5.2.2.1 Asignment()	15
5.3 Referencia de la clase Hulk.BinaryFunction	15
5.3.1 Descripción detallada	17
5.3.2 Documentación de constructores y destructores	17
5.3.2.1 BinaryFunction()	17
5.3.3 Documentación de funciones miembro	17
5.3.3.1 Evaluate()	17
5.4 Referencia de la clase Hulk.Conjunction	19
5.4.1 Descripción detallada	20
5.5 Referencia de la clase Hulk.Cosine	20
5.5.1 Descripción detallada	21
5.6 Referencia de la clase Hulk.DefaultError	22
5.6.1 Descripción detallada	22
5.7 Referencia de la clase Hulk.Disjunction	22
5.7.1 Descripción detallada	23
5.8 Referencia de la clase Hulk.Division	24
5.8.1 Descripción detallada	25
5.9 Referencia de la clase Hulk.EmptyReturn	25
5.9.1 Descripción detallada	25
5.10 Referencia de la clase Hulk.Equal	25
5.10.1 Descripción detallada	26
5.11 Referencia de la clase Hulk.ERaised	27
5.11.1 Descripción detallada	28
5.12 Referencia de la clase Hulk.FunctionCall	28

5.12.1 Descripción detallada	29
5.12.2 Documentación de constructores y destructores	29
5.12.2.1 FunctionCall()	29
5.13 Referencia de la clase Hulk.FunctionDeclaration	29
5.13.1 Descripción detallada	30
5.13.2 Documentación de constructores y destructores	30
5.13.2.1 FunctionDeclaration()	30
5.13.3 Documentación de funciones miembro	31
5.13.3.1 AddToMemory()	31
5.13.3.2 CheckDefinition()	31
5.13.3.3 Define()	31
5.13.3.4 Evaluate()	32
5.14 Referencia de la clase Hulk.GreaterEqualThan	32
5.14.1 Descripción detallada	33
5.15 Referencia de la clase Hulk.GreaterThan	34
5.15.1 Descripción detallada	35
5.16 Referencia de la clase Hulk.HulkException	35
5.16.1 Descripción detallada	35
5.17 Referencia de la clase Hulk.HulkExpression	36
5.17.1 Descripción detallada	36
5.17.2 Documentación de funciones miembro	37
5.17.2.1 CheckType()	37
5.17.2.2 GetValue()	37
5.18 Referencia de la clase Hulk.HulkMemory	37
5.18.1 Descripción detallada	38
5.18.2 Documentación de funciones miembro	38
5.18.2.1 AddNewFunction()	38
5.19 Referencia de la clase Hulk.HulkParser	38
5.19.1 Descripción detallada	39
5.19.2 Documentación de constructores y destructores	39
5.19.2.1 HulkParser()	39
5.19.3 Documentación de funciones miembro	39
5.19.3.1 Parse()	39
5.19.3.2 ParseFunctionDeclaration()	39
5.20 Referencia de la clase Hulk.IfElseStatement	40
5.20.1 Descripción detallada	41
5.20.2 Documentación de constructores y destructores	41
5.20.2.1 IfElseStatement()	41
5.21 Referencia de la clase Hulk.InstrucciónError	41
5.21.1 Descripción detallada	42
5.22 Referencia de la clase Hulk.Kompiler	42
5.22.1 Descripción detallada	42

5.22.2 Documentación de constructores y destructores	42
5.22.2.1 Kompiler()	42
5.22.3 Documentación de funciones miembro	43
5.22.3.1 Compile()	43
5.23 Referencia de la clase Hulk.LetInStatement	43
5.23.1 Descripción detallada	44
5.23.2 Documentación de constructores y destructores	44
5.23.2.1 LetInStatement()	44
5.23.3 Documentación de funciones miembro	44
5.23.3.1 Define()	44
5.24 Referencia de la clase Hulk.LexicalError	45
5.24.1 Descripción detallada	45
5.25 Referencia de la clase Hulk.Logarithm	46
5.25.1 Descripción detallada	47
5.26 Referencia de la clase Hulk.LowerEqualThan	47
5.26.1 Descripción detallada	48
5.27 Referencia de la clase Hulk.LowerThan	48
5.27.1 Descripción detallada	49
5.28 Referencia de la clase Hulk.Module	50
5.28.1 Descripción detallada	51
5.29 Referencia de la clase Hulk.Multiplication	51
5.29.1 Descripción detallada	52
5.30 Referencia de la clase Hulk.Negation	52
5.30.1 Descripción detallada	53
5.31 Referencia de la clase Hulk.Negative	54
5.31.1 Descripción detallada	55
5.32 Referencia de la clase Hulk.OverflowError	55
5.32.1 Descripción detallada	56
5.33 Referencia de la clase Hulk.Positive	56
5.33.1 Descripción detallada	57
5.34 Referencia de la clase Hulk.Power	57
5.34.1 Descripción detallada	58
5.35 Referencia de la clase Hulk.PrintFunc	59
5.35.1 Descripción detallada	59
5.35.2 Documentación de constructores y destructores	59
5.35.2.1 PrintFunc()	59
5.36 Referencia de la clase Hulk.Rand	60
5.36.1 Descripción detallada	60
5.37 Referencia de la clase Hulk.SemanticError	61
5.37.1 Descripción detallada	61
5.38 Referencia de la clase Hulk.SimpleConcatenation	61
5.38.1 Descripción detallada	62

5.39 Referencia de la clase Hulk.Sine	63
5.39.1 Descripción detallada	64
5.40 Referencia de la clase Hulk.SquaredRoot	64
5.40.1 Descripción detallada	65
5.41 Referencia de la clase Hulk.Subtraction	65
5.41.1 Descripción detallada	66
5.42 Referencia de la clase Hulk.SyntaxError	67
5.42.1 Descripción detallada	67
5.43 Referencia de la clase Hulk.UnaryFunction	67
5.43.1 Descripción detallada	68
5.43.2 Documentación de constructores y destructores	68
5.43.2.1 UnaryFunction()	68
5.43.3 Documentación de funciones miembro	69
5.43.3.1 Evaluate()	69
5.43.3.2 UnaryOperation()	69
5.44 Referencia de la clase Hulk.UnEqual	70
5.44.1 Descripción detallada	71
5.45 Referencia de la clase Hulk.Variable	71
5.45.1 Descripción detallada	72
5.45.2 Documentación de constructores y destructores	72
5.45.2.1 Variable() [1/3]	72
5.45.2.2 Variable() [2/3]	72
5.45.2.3 Variable() [3/3]	72
5.46 Referencia de la clase Hulk.VariableDeclaration	73
5.46.1 Descripción detallada	74
5.46.2 Documentación de constructores y destructores	74
5.46.2.1 VariableDeclaration() [1/2]	74
5.46.2.2 VariableDeclaration() [2/2]	74
5.47 Referencia de la clase Hulk.WhiteSpaceConcatenation	75
5.47.1 Descripción detallada	76

Capítulo 1

Índice de espacios de nombres

1.1. Lista de paquetes

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

Hulk	9
--------------------------------	---

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:

Hulk.EmptyReturn	25
Exception	
Hulk.HulkException	35
Hulk.DefaultError	22
Hulk.InstrucctionError	41
Hulk.LexicalError	45
Hulk.OverFlowError	55
Hulk.SemanticError	61
Hulk.SyntaxError	67
Hulk.HulkExpression	36
Hulk.Asignment	14
Hulk.BinaryFunction	15
Hulk.Addition	13
Hulk.Conjunction	19
Hulk.Disjunction	22
Hulk.Division	24
Hulk.Equal	25
Hulk.GreaterEqualThan	32
Hulk.GreaterThan	34
Hulk.Logarithm	46
Hulk.LowerEqualThan	47
Hulk.LowerThan	48
Hulk.Module	50
Hulk.Multiplication	51
Hulk.Power	57
Hulk.SimpleConcatenation	61
Hulk.Subtraction	65
Hulk.UnEqual	70
Hulk.WhiteSpaceConcatenation	75
Hulk.FunctionCall	28
Hulk.FunctionDeclaration	29
Hulk.IfElseStatement	40
Hulk.LetInStatement	43
Hulk.PrintFunc	59
Hulk.Rand	60

Hulk.UnaryFunction	67
Hulk.Cosine	20
Hulk.ERaised	27
Hulk.Negation	52
Hulk.Negative	54
Hulk.Positive	56
Hulk.Sine	63
Hulk.SquaredRoot	64
Hulk.Variable	71
Hulk.VariableDeclaration	73
Hulk.HulkMemory	37
Hulk.HulkParser	38
Hulk.Kompiler	42

Capítulo 3

Índice de clases

3.1. Lista de clases

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

Hulk.Addition	Clase que representa la adición. Expresión del tipo [expresión] + [expresión]	13
Hulk.Asignment	Clase que representa la expresión de asignación destructiva. Expresiones de la forma [variable] = [expresión]	14
Hulk.BinaryFunction	Representa una operación binaria. Expresiones del tipo [expresión] [operación] [expresión] . .	15
Hulk.Conjunction	Clase que representa la conjunción lógica. Expresión del tipo [expresión] & [expresión]	19
Hulk.Cosine	Representa el llamado a la función coseno. Expresiones del tipo cos([expresión])	20
Hulk.DefaultError	Representa un error genérico	22
Hulk.Disjunction	Clase que representa la disyunción lógica. Expresión del tipo [expresión] [expresión]	22
Hulk.Division	Clase que representa la división. Expresión del tipo [expresión] / [expresión]	24
Hulk.EmptyReturn	Objeto que representa un retorno vacío	25
Hulk.Equal	Clase que representa la operación de igualdad. Expresión del tipo [expresión] == [expresión] .	25
Hulk.ERaised	Representa el llamado a la función exp (e elevado a una potencia). Expresiones el tipo exp([expresión])	27
Hulk.FunctionCall	Representa los llamados a funciones expresiones de la forma [id] ([expresión], [expresión]...) .	28
Hulk.FunctionDeclaration	Representa una declaración de función inline. Expresión del tipo function [id] ([argumentos]) => [definición]	29
Hulk.GreaterEqualThan	Clase que representa la operación mayor igual que. Expresión del tipo [expresión] >= [expresión]	32
Hulk.GreaterThan	Clase que representa la operación mayor que. Expresión del tipo [expresión] > [expresión] . .	34
Hulk.HulkException	Representa los errores de cualquier tipo que se puedan dar en el lenguaje	35

Hulk.HulkExpression	Representa todas las expresiones del lenguaje HULK	36
Hulk.HulkMemory	Representa un memoria de HULK. Instancias de esta clase se encargaran de guardar las funciones	37
Hulk.HulkParser	Representa un objeto que controla el parseo de instrucciones de HULK	38
Hulk.IfElseStatement	Representa las expresiones condicionales. Expresiones del tipo if([condicion]) [expresion] else [expresion]	40
Hulk.InstrucctionError	Representa los errores ocurridos en una instruccion	41
Hulk.Kompiler	Representa un compilador de HULK. Instancias de esta clase se encargaran juntar los distintos procesos de compilacion	42
Hulk.LetInStatement	Representa a las expresiones let-in. Expresiones del tipo let [declaraciones de variables sepaa- radas por coma] in [expresion]	43
Hulk.LexicalError	Representa un error lexico. Se producen por la presencia de tokens invalidos	45
Hulk.Logarithm	Clase que representa el llamado a la funcion logaritmo. Expresion del tipo log([expresion],[expresion]) 46	
Hulk.LowerEqualThan	Clase que representa la operacion menor o igual que. Expresion del tipo [expresion] <= [expresion]	47
Hulk.LowerThan	Clase que representa la operacion menor que. Expresion del tipo [expresion] < [expresion] . .	48
Hulk.Module	Clase que representa la operacion modulo o resto. Expresion del tipo [expresion] % [expresion]	50
Hulk.Multiplication	Clase que representa la multiplicacion. Expresion del tipo [expresion] * [expresion]	51
Hulk.Negation	Representa la negacion logica. Expresiones del tipo ![expresion]	52
Hulk.Negative	Representa la operacion unaria negativa. Expresiones del tipo -[expresion]	54
Hulk.OverflowError	Representa un error producido por la sobrecarga de la pila de llamados	55
Hulk.Positive	Representa la operacion unaria positiva. Expresiones del tipo +[expresion]	56
Hulk.Power	Clase que representa la exponenciacion. Expresion del tipo [expresion] ^ [expresion]	57
Hulk.PrintFunc	Clase que representa a la funcion de imprimir en consola. Expresion del tipo print([expresion])	59
Hulk.Rand	Representa el llamado a la funcion random. Expresiones del tipo rand([expresion])	60
Hulk.SemanticError	Representa un error semantico. Se produce por el uso incorrecto de los tipos y argumentos . .	61
Hulk.SimpleConcatenation	Clase que representa la concatenacion de expresiones. Expresion del tipo [expresion] @ [expresion]	61
Hulk.Sine	Representa el llamado a la funcion seno. Expresiones del tipo sin([expresion])	63
Hulk.SquaredRoot	Representa el llamado a la funcion raiz cuadrada. Expresiones del tipo sqrt([expresion])	64
Hulk.Subtraction	Clase que representa la sustraccion. Expresion del tipo [expresion] - [expresion]	65

Hulk.SyntaxError	
Representa un error sintactico. Se produce por expresiones incompletas	67
Hulk.UnaryFunction	
Representa las operaciones unarias. Expresiones del tipo [operacion] [expresion]	67
Hulk.UnEqual	
Clase que representa la operacion desigualdad que. Expresion del tipo [expresion] != [expresion]	70
Hulk.Variable	
Clase que representa a una variable o literal	71
Hulk.VariableDeclaration	
Representa la declaracion de variables. Expresion del tipo [tipo?] [id] = [expresion]	73
Hulk.WhiteSpaceConcatenation	
Clase que representa la concatenacion con espacio en blanco intermedio. Expresion del tipo [expresion] @ [expresion]	75

Capítulo 4

Documentación de espacios de nombres

4.1. Referencia del espacio de nombres Hulk

Clases

- class [Addition](#)
Clase que representa la adición. Expresión del tipo $[expresion] + [expresion]$.
- class [Assignment](#)
Clase que representa la expresión de asignación destructiva. Expresiones de la forma $[variable] = [expresion]$.
- class [BinaryFunction](#)
Representa una operación binaria. Expresiones del tipo $[expresion] [operacion] [expresion]$.
- class [Conjunction](#)
Clase que representa la conjunción lógica. Expresión del tipo $[expresion] \& [expresion]$.
- class [Cosine](#)
Representa el llamado a la función coseno. Expresiones del tipo $\cos([expresion])$
- class [DefaultError](#)
Representa un error genérico.
- class [Disjunction](#)
Clase que representa la disyunción lógica. Expresión del tipo $[expresion] \mid [expresion]$.
- class [Division](#)
Clase que representa la división. Expresión del tipo $[expresion] / [expresion]$.
- class [EmptyReturn](#)
Objeto que representa un retorno vacío.
- class [Equal](#)
Clase que representa la operación de igualdad. Expresión del tipo $[expresion] == [expresion]$.
- class [ERaised](#)
Representa el llamado a la función exp (e elevado a una potencia). Expresiones el tipo $\exp([expresion])$
- class [ExtObject](#)
Método que deduce el tipo que tendría en HULK un objeto.
- class [FunctionCall](#)
Representa los llamados a funciones expresiones de la forma $[id] ([expresion], [expresion]...)$
- class [FunctionDeclaration](#)
Representa una declaración de función inline. Expresión del tipo $\text{function } [id] ([argumentos]) => [definicion]$.
- class [GreaterEqualThan](#)
Clase que representa la operación mayor igual que. Expresión del tipo $[expresion] \geq [expresion]$.
- class [GreaterThan](#)

- Clase que representa la operacion mayor que. Expresion del tipo [expresion] > [expresion].*
- class [HulkException](#)
Representa los errores de cualquier tipo que se puedan dar en el lenguaje.
 - class [HulkExpression](#)
Representa todas las expresiones del lenguaje HULK.
 - class [HulkInfo](#)
Clase estatica que contiene información sobre el lenguaje [Hulk](#). Operaciones, palabras clave, etc.
 - class [HulkMemory](#)
Representa un memoria de HULK. Instancias de esta clase se encargaran de guardar las funciones.
 - class [HulkParser](#)
Representa un objeto que controla el parseo de instrucciones de HULK.
 - class [IfElseStatement](#)
Representa las expresiones condicionales. Expresiones del tipo if([condicion]) [expresion] else [expresion].
 - class [InstrucctionError](#)
Representa los errores ocurridos en una instruccion.
 - class [Kompiler](#)
Representa un compilador de HULK. Instancias de esta clase se encargaran juntar los distintos procesos de compilacion.
 - class [LetInStatement](#)
Representa a las expresiones let-in. Expresiones del tipo let [declaraciones de variables sepaaradas por coma] in [expresion].
 - class [LexicalError](#)
Representa un error lexico. Se producen por la presencia de tokens invalidos.
 - class [Logarithm](#)
Clase que representa el llamado a la funcion logaritmo. Expresion del tipo log([expresion],[expresion])
 - class [LowerEqualThan](#)
Clase que representa la operacion menor o igual que. Expresion del tipo [expresion] <= [expresion].
 - class [LowerThan](#)
Clase que representa la operacion menor que. Expresion del tipo [expresion] < [expresion].
 - class [Module](#)
Clase que representa la operacion modulo o resto. Expresion del tipo [expresion] % [expresion].
 - class [Multiplication](#)
*Clase que representa la multiplicacion. Expresion del tipo [expresion] * [expresion].*
 - class [Negation](#)
Representa la negacion logica. Expresiones del tipo ![expresion].
 - class [Negative](#)
Representa la operacion unaria negativa. Expresiones del tipo -[expresion].
 - class [OverflowError](#)
Representa un error producido por la sobrecarga de la pila de llamados.
 - class [Positive](#)
Representa la operacion unaria positiva. Expresiones del tipo +[expresion].
 - class [Power](#)
Clase que representa la exponenciacion. Expresion del tipo [expresion] ^ [expresion].
 - class [PrintFunc](#)
Clase que representa a la funcion de imprimir en consola. Expresion del tipo print([expresion])
 - class [Rand](#)
Representa el llamado a la funcion random. Expresiones del tipo rand([expresion])
 - class [SemanticError](#)
Representa un error semantico. Se produce por el uso incorrecto de los tipos y argumentos.
 - class [SimpleConcatenation](#)
Clase que representa la concatenacion de expresiones. Expresion del tipo [expresion] @ [expresion].

- class [Sine](#)
Representa el llamado a la funcion seno. Expresiones del tipo $\sin([expresion])$
- class [SquaredRoot](#)
Representa el llamado a la funcion raiz cuadrada. Expresiones del tipo $\sqrt{[expresion]}$
- class [Subtraction](#)
Clase que representa la sustraccion. Expresion del tipo $[expresion] - [expresion]$.
- class [SyntaxError](#)
Representa un error sintactico. Se produce por expresiones incompletas.
- class **Tokenizer**
Clase estatica que contiene metodos utiles para el tratamiento de tokens.
- class [UnaryFunction](#)
Representa las operaciones unarias. Expresiones del tipo $[operacion] [expresion]$.
- class [Unequal](#)
Clase que representa la operacion desigualdad que. Expresion del tipo $[expresion] \neq [expresion]$.
- class [Variable](#)
Clase que representa a una variable o literal.
- class [VariableDeclaration](#)
Representa la declaracion de variables. Expresion del tipo $[tipo?] [id] = [expresion]$.
- class [WhiteSpaceConcatenation](#)
Clase que representa la concatenacion con espacio en blanco intermedio. Expresion del tipo $[expresion] @ [expresion]$.

Enumeraciones

- enum [Types](#) {
 Void , **number** , **boolean** , **hstring** ,
 dynamic }
Tipos de los valores que se pueden devolver en HULK.

Funciones

- delegate void [Print](#) (object input)
Funcion delegada que se encargara de imprimir en consola.

4.1.1. Documentación de funciones

4.1.1.1. Print()

```
delegate void Hulk.Print (
    object input )
```

Funcion delegada que se encargara de imprimir en consola.

Parámetros

<i>input</i>	
--------------	--

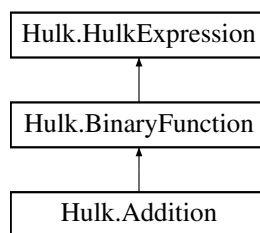
Capítulo 5

Documentación de clases

5.1. Referencia de la clase Hulk.Addition

Clase que representa la adición. Expresion del tipo `[expresion] + [expresion]`.

Diagrama de herencia de `Hulk.Addition`



Métodos públicos

- **Addition** (`HulkExpression` leftArgument, `HulkExpression` rightArgument)

Métodos públicos heredados de `Hulk.BinaryFunction`

- `BinaryFunction` (`HulkExpression` leftArgument, `HulkExpression` rightArgument)
Constructor de un objeto que representa a una funcion binaria.
- override object **GetValue** (bool execute)
- override `Types CheckType` ()
- object `Evaluate` (object left, object right)
Evalua la expresion.
- delegate object **BinaryOperation** (object left, object right)

Métodos públicos heredados de `Hulk.HulkExpression`

- object `GetValue` (bool execute)
Metodo que devuelve el valor de la expresion.
- `Types CheckType` ()
Chequea que no existan errores semanticos en la expresion y devuelve el tipo de retorno de la misma.

Otros miembros heredados

Propiedades heredados de **Hulk.BinaryFunction**

- **HulkExpression LeftArgument** [get, protected set]
Expresion que representa al miembro izquierdo de la expresion.
- **HulkExpression RightArgument** [get, protected set]
Expresion que representa al miembro derecho de la expresion.
- **Types ReturnedType** [get, protected set]
Tipo de retorno aceptado.
- **Types EnteredType** [get, protected set]
Tipo de extrada aceptado.
- **Type AcceptedType** [get, protected set]
Tipo aceptado del objeto que devuelven los argumentos.
- string **OperationToken** [get, protected set]
- BinaryOperation **Operation** [get, protected set]

Propiedades heredados de **Hulk.HulkExpression**

- bool **IsDependent** [get, protected set]
Especificacion de si la expresion depende de un parametro (argumento de funcion) o no.

5.1.1. Descripción detallada

Clase que representa la adición. Expresion del tipo [expresion] + [expresion].

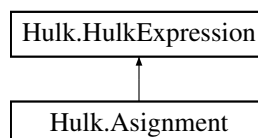
La documentación de esta clase está generada del siguiente archivo:

- C:/Users/Jossue/Cosas/proyectos y cosas/Proyectos de Programacion/H.U.L.K/Hulk/BinaryOperations.cs

5.2. Referencia de la clase Hulk.Assignment

Clase que representa la expresion de asignación destructiva. Expresiones de la forma [variable] = [expresion].

Diagrama de herencia de Hulk.Assignment



Métodos públicos

- **Assignment** (List< **Variable** > Vars, **HulkExpression** ValueExp)
Constructor de un objeto que representa a la operacion de asignación destructiva.
- override object **GetValue** (bool execute)
- override **Types CheckType** ()

Métodos públicos heredados de Hulk.HulkExpression

- object **GetValue** (bool execute)
Método que devuelve el valor de la expresion.
- **Types CheckType** ()
Chequea que no existan errores semanticos en la expresion y devuelve el tipo de retorno de la misma.

Propiedades

- List< **Variable** > **Variables** [get, protected set]
Variables a las que se les esta cambiando el valor.
- **HulkExpression ValueExpression** [get, protected set]
Expresion del valor que se va a cambiar.

Propiedades heredados de Hulk.HulkExpression

- bool **IsDependent** [get, protected set]
Especificacion de si la expresion depende de un parametro (argumento de funcion) o no.

5.2.1. Descripción detallada

Clase que representa la expresion de asignación destructiva. Expresiones de la forma [variable] = [expresion].

5.2.2. Documentación de constructores y destructores

5.2.2.1. Assignment()

```
Hulk.Assignment.Assignment (
    List< Variable > Vars,
    HulkExpression ValueExp )
```

Constructor de un objeto que representa a la operacion de asignación destructiva.

Parámetros

<i>Vars</i>	Lista de variables al las que se les van a cambiar los valores
<i>ValueExp</i>	Expresion que será asignada

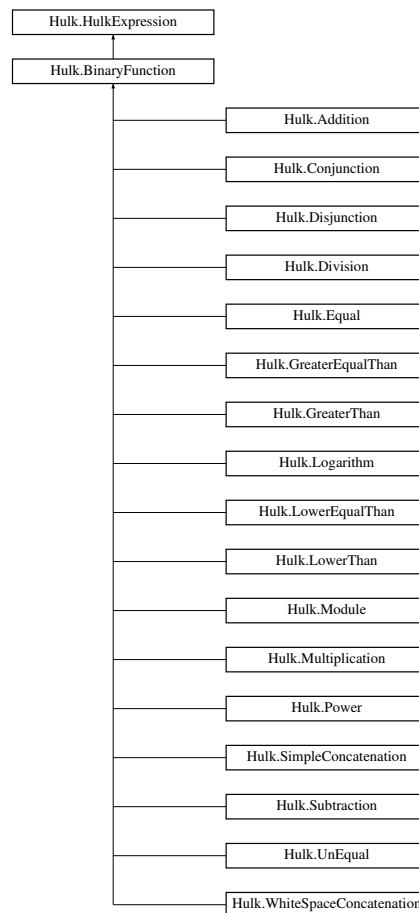
La documentación de esta clase está generada del siguiente archivo:

- C:/Users/Jossue/Cosas/proyectos y cosas/Proyectos de Programacion/H.U.L.K/Hulk/BasicExpressions.cs

5.3. Referencia de la clase Hulk.BinaryFunction

Representa una operacion binaria. Expresiones del tipo [expresion] [operacion] [expresion].

Diagrama de herencia de Hulk.BinaryFunction



Métodos públicos

- **BinaryFunction** ([HulkExpression](#) leftArgument, [HulkExpression](#) rightArgument)
Contructor de un objeto que representa a una funcion binaria.
- override object **GetValue** (bool execute)
- override [Types](#) **CheckType** ()
- object [Evaluate](#) (object left, object right)
Evalua la expresion.
- delegate object **BinaryOperation** (object left, object right)

Métodos públicos heredados de [Hulk.HulkExpression](#)

- object [GetValue](#) (bool execute)
Metodo que devuelve el valor de la expresion.
- [Types](#) **CheckType** ()
Chequea que no existan errores semanticos en la expresion y devuelve el tipo de retorno de la misma.

Propiedades

- **HulkExpression LeftArgument** [get, protected set]
Expresion que representa al miembro izquierdo de la expresion.
- **HulkExpression RightArgument** [get, protected set]
Expresion que representa al miembro derecho de la expresion.
- **Types ReturnedType** [get, protected set]
Tipo de retorno aceptado.
- **Types EnteredType** [get, protected set]
Tipo de extrada aceptado.
- **Type AcceptedType** [get, protected set]
Tipo aceptado del objeto que devuelven los argumentos.
- string **OperationToken** [get, protected set]
- BinaryOperation **Operation** [get, protected set]

Propiedades heredados de Hulk.HulkExpression

- bool **IsDependent** [get, protected set]
Especificacion de si la expresion depende de un parametro (argumento de funcion) o no.

5.3.1. Descripción detallada

Representa una operacion binaria. Expresiones del tipo [expresion] [operacion] [expresion].

5.3.2. Documentación de constructores y destructores

5.3.2.1. BinaryFunction()

```
Hulk.BinaryFunction.BinaryFunction (
    HulkExpression leftArgument,
    HulkExpression rightArgument )
```

Contructor de un objeto que representa a una funcion binaria.

Parámetros

<i>leftArgument</i>	Miembro izquierdo
<i>rightArgument</i>	Miembro derecho

5.3.3. Documentación de funciones miembro

5.3.3.1. Evaluate()

```
object Hulk.BinaryFunction.Evaluate (
    object left,
    object right )
```

Evalua la expresion.

Parámetros

<i>left</i>	Valor del miembro izquierdo
<i>right</i>	Valor del miembro derecho

Devuelve

Valor de la expresion

Excepciones

<i>SemanticError</i>	
----------------------	--

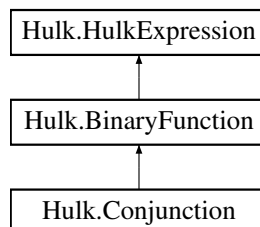
La documentación de esta clase está generada del siguiente archivo:

- C:/Users/Jossue/Cosas/proyectos y cosas/Proyectos de Programacion/H.U.L.K/Hulk/BinaryOperations.cs

5.4. Referencia de la clase Hulk.Conjunction

Clase que representa la conjuncion logica. Expresion del tipo [expresion] & [expresion].

Diagrama de herencia de Hulk.Conjunction



Métodos públicos

- **Conjunction** ([HulkExpression](#) leftArgument, [HulkExpression](#) rightArgument)

Métodos públicos heredados de [Hulk.BinaryFunction](#)

- [BinaryFunction](#) ([HulkExpression](#) leftArgument, [HulkExpression](#) rightArgument)
Contructor de un objeto que representa a una funcion binaria.
- override object **GetValue** (bool execute)
- override [Types](#) **CheckType** ()
- object [Evaluate](#) (object left, object right)
Evalua la expresion.
- delegate object **BinaryOperation** (object left, object right)

Métodos públicos heredados de **Hulk.HulkExpression**

- object **GetValue** (bool execute)
Metodo que devuelve el valor de la expresion.
- **Types CheckType** ()
Chequea que no existan errores semanticos en la expresion y devuelve el tipo de retorno de la misma.

Otros miembros heredados

Propiedades heredados de **Hulk.BinaryFunction**

- **HulkExpression LeftArgument** [get, protected set]
Expresion que representa al miembro izquierdo de la expresion.
- **HulkExpression RightArgument** [get, protected set]
Expresion que representa al miembro derecho de la expresion.
- **Types ReturnedType** [get, protected set]
Tipo de retorno aceptado.
- **Types EnteredType** [get, protected set]
Tipo de extrada aceptado.
- **Type AcceptedType** [get, protected set]
Tipo aceptado del objeto que devuelven los argumentos.
- string **OperationToken** [get, protected set]
- BinaryOperation **Operation** [get, protected set]

Propiedades heredados de **Hulk.HulkExpression**

- bool **IsDependent** [get, protected set]
Especificacion de si la expresion depende de un parametro (argumento de funcion) o no.

5.4.1. Descripción detallada

Clase que representa la conjuncion logica. Expresion del tipo [expresion] & [expresion].

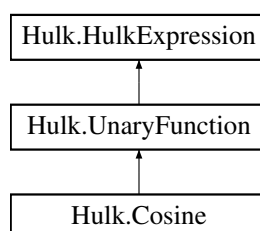
La documentación de esta clase está generada del siguiente archivo:

- C:/Users/Jossue/Cosas/proyectos y cosas/Proyectos de Programacion/H.U.L.K/Hulk/BinaryOperations.cs

5.5. Referencia de la clase **Hulk.Cosine**

Representa el llamado a la funcion coseno. Expresiones del tipo cos([expresion])

Diagrama de herencia de Hulk.Cosine



Métodos públicos

- **Cosine** ([HulkExpression](#) Arg)

Métodos públicos heredados de [Hulk.UnaryFunction](#)

- **UnaryFunction** ([HulkExpression](#) Arg)
Construye un objeto que representa una operacion unaria.
- override object **GetValue** (bool execute)
- delegate object **UnaryOperation** (object arg)
Funcion unaria delegada.
- object **Evaluate** (object arg)
Evalua la operacion unaria.
- override [Types](#) **CheckType** ()

Métodos públicos heredados de [Hulk.HulkExpression](#)

- object **GetValue** (bool execute)
Metodo que devuelve el valor de la expresion.
- [Types](#) **CheckType** ()
Chequea que no existan errores semanticos en la expresion y devuelve el tipo de retorno de la misma.

Otros miembros heredados**Propiedades heredados de [Hulk.UnaryFunction](#)**

- [HulkExpression](#) **Argument** [get, protected set]
- [Types](#) **ReturnedType** [get, protected set]
Tipo de retorno de la operacion.
- [Types](#) **EnteredType** [get, protected set]
Tipo de entrada de la operacion aceptado.
- Type **AcceptedType** [get, protected set]
Tipo de valor aceptado para el valor del argumento.
- string **OperationToken** [get, protected set]
Token de las operacion.
- [UnaryOperation](#) **Operation** [get, protected set]
Funcion que efectua la operacion unaria.

Propiedades heredados de [Hulk.HulkExpression](#)

- bool **IsDependent** [get, protected set]
Especificacion de si la expresion depende de un parametro (argumento de funcion) o no.

5.5.1. Descripción detallada

Representa el llamado a la funcion coseno. Expresiones del tipo cos([expresion])

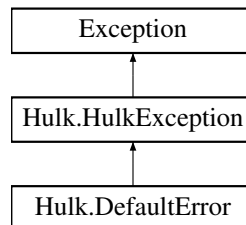
La documentación de esta clase está generada del siguiente archivo:

- C:/Users/Jossue/Cosas/proyectos y cosas/Proyectos de Programacion/H.U.L.K/Hulk/UnaryOperations.cs

5.6. Referencia de la clase Hulk.DefaultError

Representa un error generico.

Diagrama de herencia de Hulk.DefaultError



Métodos públicos

- **DefaultError** (string message)
- **DefaultError** (string message, string errorEspecificacion)

Otros miembros heredados

Propiedades heredados de **Hulk.HulkException**

- override string **Message** [get]
- string **MessageStart** [get, protected set]
Caption que inicia el mensaje de error.
- string **MessageDefinition** [get, protected set]
Descripcion del error.

5.6.1. Descripción detallada

Representa un error generico.

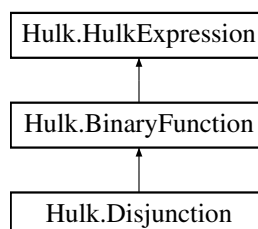
La documentación de esta clase está generada del siguiente archivo:

- C:/Users/Jossue/Cosas/proyectos y cosas/Proyectos de Programacion/H.U.L.K/Hulk/Exceptions.cs

5.7. Referencia de la clase Hulk.Disjunction

Clase que representa la disyuncion logica. Expresion del tipo [expresion] | [expresion].

Diagrama de herencia de Hulk.Disjunction



Métodos públicos

- **Disjunction** ([HulkExpression](#) leftArgument, [HulkExpression](#) rightArgument)

Métodos públicos heredados de [Hulk.BinaryFunction](#)

- **BinaryFunction** ([HulkExpression](#) leftArgument, [HulkExpression](#) rightArgument)
Constructor de un objeto que representa a una funcion binaria.
- override object **GetValue** (bool execute)
- override [Types](#) **CheckType** ()
- object **Evaluate** (object left, object right)
Evalua la expresion.
- delegate object **BinaryOperation** (object left, object right)

Métodos públicos heredados de [Hulk.HulkExpression](#)

- object **GetValue** (bool execute)
Metodo que devuelve el valor de la expresion.
- [Types](#) **CheckType** ()
Chequea que no existan errores semanticos en la expresion y devuelve el tipo de retorno de la misma.

Otros miembros heredados

Propiedades heredados de [Hulk.BinaryFunction](#)

- [HulkExpression](#) **LeftArgument** [get, protected set]
Expresion que representa al miembro izquierdo de la expresion.
- [HulkExpression](#) **RightArgument** [get, protected set]
Expresion que representa al miembro derecho de la expresion.
- [Types](#) **ReturnedType** [get, protected set]
Tipo de retorno aceptado.
- [Types](#) **EnteredType** [get, protected set]
Tipo de extrada aceptado.
- [Type](#) **AcceptedType** [get, protected set]
Tipo aceptado del objeto que devuelven los argumentos.
- string **OperationToken** [get, protected set]
- [BinaryOperation](#) **Operation** [get, protected set]

Propiedades heredados de [Hulk.HulkExpression](#)

- bool **IsDependent** [get, protected set]
Especificacion de si la expresion depende de un parametro (argumento de funcion) o no.

5.7.1. Descripción detallada

Clase que representa la disyuncion logica. Expresion del tipo [expresion] | [expresion].

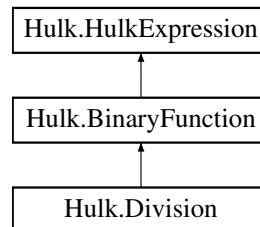
La documentación de esta clase está generada del siguiente archivo:

- C:/Users/Jossue/Cosas/proyectos y cosas/Proyectos de Programacion/H.U.L.K/Hulk/BinaryOperations.cs

5.8. Referencia de la clase Hulk.Division

Clase que representa la division. Expresion del tipo [expresion] / [expresion].

Diagrama de herencia de Hulk.Division



Métodos públicos

- **Division** ([HulkExpression](#) leftArgument, [HulkExpression](#) rightArgument)

Métodos públicos heredados de [Hulk.BinaryFunction](#)

- [BinaryFunction](#) ([HulkExpression](#) leftArgument, [HulkExpression](#) rightArgument)
Contructor de un objeto que representa a una funcion binaria.
- override object **GetValue** (bool execute)
- override [Types](#) **CheckType** ()
- object [Evaluate](#) (object left, object right)
Evalua la expresion.
- delegate object **BinaryOperation** (object left, object right)

Métodos públicos heredados de [Hulk.HulkExpression](#)

- object [GetValue](#) (bool execute)
Metodo que devuelve el valor de la expresion.
- [Types](#) **CheckType** ()
Chequea que no existan errores semanticos en la expresion y devuelve el tipo de retorno de la misma.

Otros miembros heredados

Propiedades heredados de [Hulk.BinaryFunction](#)

- [HulkExpression](#) **LeftArgument** [get, protected set]
Expresion que representa al miembro izquierdo de la expresion.
- [HulkExpression](#) **RightArgument** [get, protected set]
Expresion que representa al miembro derecho de la expresion.
- [Types](#) **ReturnedType** [get, protected set]
Tipo de retorno aceptado.
- [Types](#) **EnteredType** [get, protected set]
Tipo de extrada aceptado.
- Type **AcceptedType** [get, protected set]
Tipo aceptado del objeto que devuelven los argumentos.
- string **OperationToken** [get, protected set]
- BinaryOperation **Operation** [get, protected set]

Propiedades heredados de [Hulk.HulkExpression](#)

- **bool IsDependent** [get, protected set]

Especificacion de si la expresion depende de un parametro (argumento de funcion) o no.

5.8.1. Descripción detallada

Clase que representa la division. Expresion del tipo [expresion] / [expresion].

La documentación de esta clase está generada del siguiente archivo:

- C:/Users/Jossue/Cosas/proyectos y cosas/Proyectos de Programacion/H.U.L.K/Hulk/BinaryOperations.cs

5.9. Referencia de la clase Hulk.EmptyReturn

Objeto que representa un retorno vacio.

5.9.1. Descripción detallada

Objeto que representa un retorno vacio.

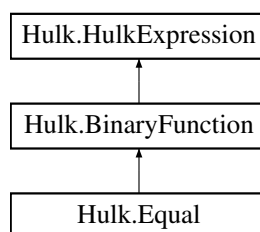
La documentación de esta clase está generada del siguiente archivo:

- C:/Users/Jossue/Cosas/proyectos y cosas/Proyectos de Programacion/H.U.L.K/Hulk/HulkEpression.cs

5.10. Referencia de la clase Hulk.Equal

Clase que representa la operacion de igualdad. Expresion del tipo [expresion] == [expresion].

Diagrama de herencia de Hulk.Equal



Métodos públicos

- **Equal** ([HulkExpression](#) leftArgument, [HulkExpression](#) rightArgument)

Métodos públicos heredados de **Hulk.BinaryFunction**

- **BinaryFunction** (**HulkExpression** leftArgument, **HulkExpression** rightArgument)
Constructor de un objeto que representa a una funcion binaria.
- override object **GetValue** (bool execute)
- override **Types CheckType** ()
- object **Evaluate** (object left, object right)
Evalua la expresion.
- delegate object **BinaryOperation** (object left, object right)

Métodos públicos heredados de **Hulk.HulkExpression**

- object **GetValue** (bool execute)
Metodo que devuelve el valor de la expresion.
- **Types CheckType** ()
Chequea que no existan errores semanticos en la expresion y devuelve el tipo de retorno de la misma.

Otros miembros heredados

Propiedades heredados de **Hulk.BinaryFunction**

- **HulkExpression LeftArgument** [get, protected set]
Expresion que representa al miembro izquierdo de la expresion.
- **HulkExpression RightArgument** [get, protected set]
Expresion que representa al miembro derecho de la expresion.
- **Types ReturnedType** [get, protected set]
Tipo de retorno aceptado.
- **Types EnteredType** [get, protected set]
Tipo de extrada aceptado.
- **Type AcceptedType** [get, protected set]
Tipo aceptado del objeto que devuelven los argumentos.
- string **OperationToken** [get, protected set]
- BinaryOperation **Operation** [get, protected set]

Propiedades heredados de **Hulk.HulkExpression**

- bool **IsDependent** [get, protected set]
Especificacion de si la expresion depende de un parametro (argumento de funcion) o no.

5.10.1. Descripción detallada

Clase que representa la operacion de igualdad. Expresion del tipo [expresion] == [expresion].

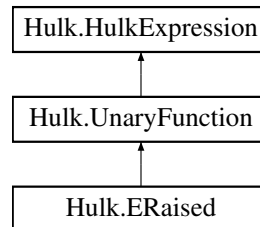
La documentación de esta clase está generada del siguiente archivo:

- C:/Users/Jossue/Cosas/proyectos y cosas/Proyectos de Programacion/H.U.L.K/Hulk/BinaryOperations.cs

5.11. Referencia de la clase Hulk.ERaised

Representa el llamado a la funcion exp (e elevado a una potencia). Expresiones el tipo exp([expresion])

Diagrama de herencia de Hulk.ERaised



Métodos públicos

- **ERaised** ([HulkExpression](#) Arg)

Métodos públicos heredados de [Hulk.UnaryFunction](#)

- **UnaryFunction** ([HulkExpression](#) Arg)
Construye un objeto que representa una operacion unaria.
- override object **GetValue** (bool execute)
- delegate object **UnaryOperation** (object arg)
Funcion unaria delegada.
- object **Evaluate** (object arg)
Evalua la operacion unaria.
- override **Types CheckType** ()

Métodos públicos heredados de [Hulk.HulkExpression](#)

- object **GetValue** (bool execute)
Metodo que devuelve el valor de la expresion.
- **Types CheckType** ()
Chequea que no existan errores semanticos en la expresion y devuelve el tipo de retorno de la misma.

Otros miembros heredados

Propiedades heredados de [Hulk.UnaryFunction](#)

- **HulkExpression Argument** [get, protected set]
- **Types ReturnedType** [get, protected set]
Tipo de retorno de la operacion.
- **Types EnteredType** [get, protected set]
Tipo de entrada de la operacion aceptado.
- **Type AcceptedType** [get, protected set]
Tipo de valor aceptado para el valor del argumento.
- string **OperationToken** [get, protected set]
Token de las operacion.
- **UnaryOperation Operation** [get, protected set]
Funcion que efectua la operacion unaria.

Propiedades heredadas de [Hulk.HulkExpression](#)

- `bool IsDependent` [get, protected set]
Especificación de si la expresión depende de un parámetro (argumento de función) o no.

5.11.1. Descripción detallada

Representa el llamado a la función exp (e elevado a una potencia). Expresiones el tipo `exp([expresion])`

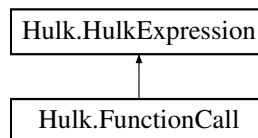
La documentación de esta clase está generada del siguiente archivo:

- `C:/Users/Jossue/Cosas/proyectos y cosas/Proyectos de Programacion/H.U.L.K/Hulk/UnaryOperations.cs`

5.12. Referencia de la clase [Hulk.FunctionCall](#)

Representa los llamados a funciones expresiones de la forma `[id] ([expresion], [expresion]...)`

Diagrama de herencia de [Hulk.FunctionCall](#)



Métodos públicos

- `FunctionCall` (string name, List< [HulkExpression](#) > Args, [FunctionDeclaration](#) Def)
Constructor de una expresión que representa un llamado a función.
- `override object GetValue` (bool execute)
- `override Types CheckType` ()

Métodos públicos heredados de [Hulk.HulkExpression](#)

- `object GetValue` (bool execute)
Método que devuelve el valor de la expresión.
- `Types CheckType` ()
Chequea que no existan errores semánticos en la expresión y devuelve el tipo de retorno de la misma.

Propiedades

- `string Name` [get, protected set]
Nombre de la función a la que sea llama.
- `List< HulkExpression > Arguments` [get, protected set]
Lista de expresiones que representan a los argumentos.
- `FunctionDeclaration Definition` [get, protected set]
Definición de la función.

Propiedades heredados de [Hulk.HulkExpression](#)

- bool **IsDependent** [get, protected set]

Especificacion de si la expresion depende de un parametro (argumento de funcion) o no.

5.12.1. Descripción detallada

Representa los llamados a funciones espresiones de la forma [id] ([expresion], [expresion]...)

5.12.2. Documentación de constructores y destructores

5.12.2.1. FunctionCall()

```
Hulk.FunctionCall.FunctionCall (
    string name,
    List< HulkExpression > Args,
    FunctionDeclaration Def )
```

Constructor de una expresion que representa un llamado a funcion.

Parámetros

<i>name</i>	Nombre de la funcion que se esta llamando
<i>Args</i>	Lista de los argumentos de la funcion
<i>Def</i>	Referencia al lugar en memoria donde se define la funcion

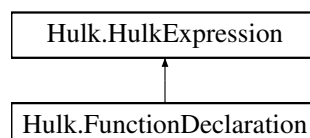
La documentación de esta clase está generada del siguiente archivo:

- C:/Users/Jossue/Cosas/proyectos y cosas/Proyectos de Programacion/H.U.L.K/Hulk/BasicExpressions.cs

5.13. Referencia de la clase Hulk.FunctionDeclaration

Representa una declaraion de funcion inline. Expresion del tipo function [id] ([argumentos]) => [definicion].

Diagrama de herencia de Hulk.FunctionDeclaration



Métodos públicos

- **FunctionDeclaration** (string name, List< string > argNames)
Construye un objeto que representa una declaracion de funcion.
- object **Evaluate** (List< **HulkExpression** > Args, bool execute)
Evalua la funcion.
- void **Define** (**HulkExpression** definition)
Asigna la expresion que define al cuerpo de la funcion.
- **Types CheckDefinition** ()
Aplica el chequeo de tipos a la expresion que define la funcion.
- void **AddToMemory** (**HulkMemory** Memoria)
Añade la funcion a una memoria.
- override object **GetValue** (bool execute)
- override **Types CheckType** ()

Métodos públicos heredados de **Hulk.HulkExpression**

- object **GetValue** (bool execute)
Metodo que devuelve el valor de la expresion.
- **Types CheckType** ()
Chequea que no existan errores semanticos en la expresion y devuelve el tipo de retorno de la misma.

Propiedades

- List< string > **ArgumentNames** [get]
Lista de nombres de los argumentos.
- string **FunctionName** [get]
Nombre de la funcion que se declara.
- Dictionary< string, **Variable** > **Arguments** [get]
Lugar donde se guardan las variables que representan los argumentos de la funcion.
- **HulkExpression Definition** [get]
Expresion que define la funcion.
- **Types ReturnedType** [get]
Tipo de retorno de la funcion.

Propiedades heredados de **Hulk.HulkExpression**

- bool **IsDependent** [get, protected set]
Especificacion de si la expresion depende de un parametro (argumento de funcion) o no.

5.13.1. Descripción detallada

Representa una declaracion de funcion inline. Expresion del tipo function [id] ([argumentos]) => [definicion].

5.13.2. Documentación de constructores y destructores

5.13.2.1. FunctionDeclaration()

```
Hulk.FunctionDeclaration.FunctionDeclaration (
    string name,
    List< string > argNames )
```

Construye un objeto que representa una declaracion de funcion.

Parámetros

<i>name</i>	Nombre de la funcion que se declara
<i>argNames</i>	Nombres de los argumentos

5.13.3. Documentación de funciones miembro

5.13.3.1. AddToMemory()

```
void Hulk.FunctionDeclaration.AddToMemory (
    HulkMemory Memoria )
```

Añade la funcion a una memoria.

Parámetros

<i>Memoria</i>	
----------------	--

5.13.3.2. CheckDefinition()

```
Types Hulk.FunctionDeclaration.CheckDefinition ( )
```

Aplica el chequeo de tipos a la expresion que define la funcion.

Devuelve

Excepciones

<i>DefaultError</i>	
---------------------	--

5.13.3.3. Define()

```
void Hulk.FunctionDeclaration.Define (
    HulkExpression definition )
```

Asigna la expresion que define al cuerpo de la funcion.

Parámetros

<i>definition</i>	
-------------------	--

5.13.3.4. Evaluate()

```
object Hulk.FunctionDeclaration.Evaluate (
    List< HulkExpression > Args,
    bool execute )
```

Evalua la funcion.

Parámetros

<i>Args</i>	Lista de las expresiones que dan valores a los argumentos
<i>execute</i>	

Devuelve

Excepciones

<i>SemanticError</i>	
<i>OverFlowError</i>	

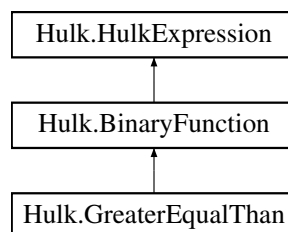
La documentación de esta clase está generada del siguiente archivo:

- C:/Users/Jossue/Cosas/proyectos y cosas/Proyectos de Programacion/H.U.L.K/Hulk/Declarations.cs

5.14. Referencia de la clase Hulk.GreaterEqualThan

Clase que representa la operacion mayor igual que. Expresion del tipo [expresion] >= [expresion].

Diagrama de herencia de Hulk.GreaterEqualThan



Métodos públicos

- **GreaterEqualThan** ([HulkExpression](#) leftArgument, [HulkExpression](#) rightArgument)

Métodos públicos heredados de Hulk.BinaryFunction

- **BinaryFunction** ([HulkExpression](#) leftArgument, [HulkExpression](#) rightArgument)
Constructor de un objeto que representa a una funcion binaria.
- override object **GetValue** (bool execute)
- override [Types](#) **CheckType** ()
- object **Evaluate** (object left, object right)
Evalua la expresion.
- delegate object **BinaryOperation** (object left, object right)

Métodos públicos heredados de Hulk.HulkExpression

- object [GetValue](#) (bool execute)
Metodo que devuelve el valor de la expresion.
- [Types](#) **CheckType** ()
Chequea que no existan errores semanticos en la expresion y devuelve el tipo de retorno de la misma.

Otros miembros heredados

Propiedades heredados de Hulk.BinaryFunction

- [HulkExpression](#) **LeftArgument** [get, protected set]
Expresion que representa al miembro izquierdo de la expresion.
- [HulkExpression](#) **RightArgument** [get, protected set]
Expresion que representa al miembro derecho de la expresion.
- [Types](#) **ReturnedType** [get, protected set]
Tipo de retorno aceptado.
- [Types](#) **EnteredType** [get, protected set]
Tipo de extrada aceptado.
- [Type](#) **AcceptedType** [get, protected set]
Tipo aceptado del objeto que devuelven los argumentos.
- string **OperationToken** [get, protected set]
- [BinaryOperation](#) **Operation** [get, protected set]

Propiedades heredados de Hulk.HulkExpression

- bool **IsDependent** [get, protected set]
Especificacion de si la expresion depende de un parametro (argumento de funcion) o no.

5.14.1. Descripción detallada

Clase que representa la operacion mayor igual que. Expresion del tipo [expresion] >= [expresion].

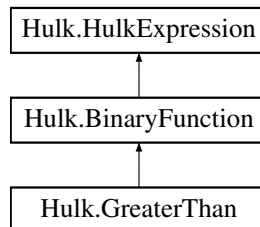
La documentación de esta clase está generada del siguiente archivo:

- C:/Users/Jossue/Cosas/proyectos y cosas/Proyectos de Programacion/H.U.L.K/Hulk/BinaryOperations.cs

5.15. Referencia de la clase Hulk.GreaterThan

Clase que representa la operacion mayor que. Expresion del tipo [expresion] > [expresion].

Diagrama de herencia de Hulk.GreaterThan



Métodos públicos

- **GreaterThan** ([HulkExpression](#) leftArgument, [HulkExpression](#) rightArgument)

Métodos públicos heredados de [Hulk.BinaryFunction](#)

- [BinaryFunction](#) ([HulkExpression](#) leftArgument, [HulkExpression](#) rightArgument)
Contructor de un objeto que representa a una funcion binaria.
- override object **GetValue** (bool execute)
- override [Types](#) **CheckType** ()
- object [Evaluate](#) (object left, object right)
Evalua la expresion.
- delegate object **BinaryOperation** (object left, object right)

Métodos públicos heredados de [Hulk.HulkExpression](#)

- object [GetValue](#) (bool execute)
Metodo que devuelve el valor de la expresion.
- [Types](#) **CheckType** ()
Chequea que no existan errores semanticos en la expresion y devuelve el tipo de retorno de la misma.

Otros miembros heredados

Propiedades heredados de [Hulk.BinaryFunction](#)

- [HulkExpression](#) **LeftArgument** [get, protected set]
Expresion que representa al miembro izquierdo de la expresion.
- [HulkExpression](#) **RightArgument** [get, protected set]
Expresion que representa al miembro derecho de la expresion.
- [Types](#) **ReturnedType** [get, protected set]
Tipo de retorno aceptado.
- [Types](#) **EnteredType** [get, protected set]
Tipo de extrada aceptado.
- Type **AcceptedType** [get, protected set]
Tipo aceptado del objeto que devuelven los argumentos.
- string **OperationToken** [get, protected set]
- BinaryOperation **Operation** [get, protected set]

Propiedades heredados de [Hulk.HulkExpression](#)

- bool **IsDependent** [get, protected set]

Especificacion de si la expresion depende de un parametro (argumento de funcion) o no.

5.15.1. Descripción detallada

Clase que representa la operacion mayor que. Expresion del tipo [expresion] > [expresion].

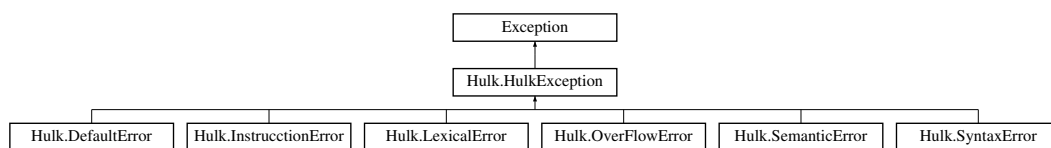
La documentación de esta clase está generada del siguiente archivo:

- C:/Users/Jossue/Cosas/proyectos y cosas/Proyectos de Programacion/H.U.L.K/Hulk/BinaryOperations.cs

5.16. Referencia de la clase Hulk.HulkException

Representa los errores de cualquier tipo que se puedan dar en el lenguaje.

Diagrama de herencia de Hulk.HulkException



Propiedades

- override string **Message** [get]
- string **MessageStart** [get, protected set]
Caption que inicia el mensaje de error.
- string **MessageDefinition** [get, protected set]
Descripcion del error.

5.16.1. Descripción detallada

Representa los errores de cualquier tipo que se puedan dar en el lenguaje.

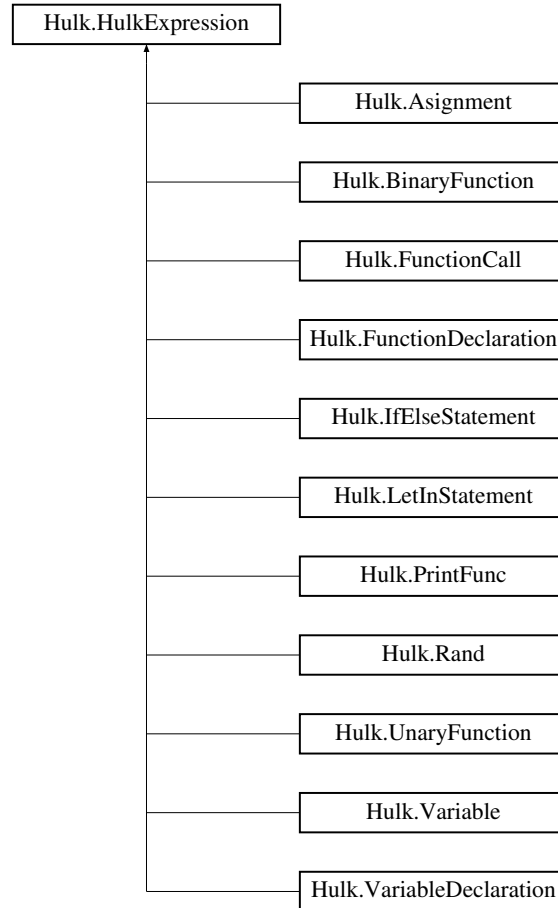
La documentación de esta clase está generada del siguiente archivo:

- C:/Users/Jossue/Cosas/proyectos y cosas/Proyectos de Programacion/H.U.L.K/Hulk/Exceptions.cs

5.17. Referencia de la clase Hulk.HulkExpression

Representa todas las expresiones del lenguaje HULK.

Diagrama de herencia de Hulk.HulkExpression



Métodos públicos

- object [GetValue](#) (bool execute)
Método que devuelve el valor de la expresión.
- [Types CheckType](#) ()
Chequea que no existan errores semánticos en la expresión y devuelve el tipo de retorno de la misma.

Propiedades

- bool **IsDependent** [get, protected set]
Especificación de si la expresión depende de un parámetro (argumento de función) o no.

5.17.1. Descripción detallada

Representa todas las expresiones del lenguaje HULK.

5.17.2. Documentación de funciones miembro

5.17.2.1. CheckType()

`Types` Hulk.HulkExpression.CheckType () [abstract]

Chequea que no existan errores semanticos en la expresion y devuelve el tipo de retorno de la misma.

Devuelve

Tipo de retorno de la expresion

5.17.2.2. GetValue()

`object` Hulk.HulkExpression.GetValue (
 `bool execute`) [abstract]

Metodo que devuelve el valor de la expresion.

Parámetros

<code>execute</code>	Indica si se debe ejecutar o no una accion al devolver el valor, como imprimir algo o cambiar algun valor
----------------------	---

Devuelve

Valor de la expresion

La documentación de esta clase está generada del siguiente archivo:

- C:/Users/Jossue/Cosas/proyectos y cosas/Proyectos de Programacion/H.U.L.K/Hulk/HulkEpression.cs

5.18. Referencia de la clase Hulk.HulkMemory

Representa un memoria de HULK. Instancias de esta clase se encargaran de guardar las funciones.

Métodos públicos

- **HulkMemory** ()
Construye un memoria de HULK.
- void **AddNewFunction** (string key, [FunctionDeclaration](#) Val)
Agrega una nueva funcion a la memoria.

Propiedades

- Dictionary< string, [FunctionDeclaration](#) > **FunctionsStorage** [get]
Lugar de la memoria donde se guardan las funciones.

5.18.1. Descripción detallada

Representa un memoria de HULK. Instancias de esta clase se encargaran de guardar las funciones.

5.18.2. Documentación de funciones miembro

5.18.2.1. AddNewFunction()

```
void Hulk.HulkMemory.AddNewFunction (
    string key,
    FunctionDeclaration Val )
```

Agrega una nueva funcion a la memoria.

Parámetros

<i>key</i>	Nombre de la funcion
<i>Val</i>	Declaracion de la funcion

Excepciones

<i>DefaultError</i>	
---------------------	--

La documentación de esta clase está generada del siguiente archivo:

- C:/Users/Jossue/Cosas/proyectos y cosas/Proyectos de Programacion/H.U.L.K/Hulk/Memory.cs

5.19. Referencia de la clase Hulk.HulkParser

Representa un objeto que controla el parseo de instrucciones de HULK.

Métodos públicos

- [HulkParser](#) ([HulkMemory](#) Mem, [Print](#) printHandler)
Construye un parseador del lenguaje HULK.
- [HulkExpression Parse](#) (string[] tokens)
Parsea una instruccion de HULK.
- [HulkExpression ParseFunctionDeclaration](#) (string[] tokens, int start, int end)
Intenta parsear una declaracion funcion.

Propiedades

- [HulkMemory Memoria](#) [get]
Memoria donde se guardaran las funciones declaradas.

5.19.1. Descripción detallada

Representa un objeto que controla el parseo de instrucciones de HULK.

5.19.2. Documentación de constructores y destructores

5.19.2.1. HulkParser()

```
Hulk.HulkParser.HulkParser (
    HulkMemory Mem,
    Print printHandler )
```

Construye un parseador del lenguaje HULK.

Parámetros

<i>Mem</i>	Memoria donde se guardaran las funciones
<i>printHandler</i>	Funcion que se encargara de imprimir

5.19.3. Documentación de funciones miembro

5.19.3.1. Parse()

```
HulkExpression Hulk.HulkParser.Parse (
    string[] tokens )
```

Parsea una instruccion de HULK.

Parámetros

<i>tokens</i>	Arreglo de tokens que representan la entrada
---------------	--

Devuelve

5.19.3.2. ParseFunctionDeclaration()

```
HulkExpression Hulk.HulkParser.ParseFunctionDeclaration (
    string[] tokens,
    int start,
    int end )
```

Intenta parsear una declaracion funcion.

Parámetros

<i>tokens</i>	Arreglo de tokens que representan la entrada
<i>start</i>	Puntero que representa el punto de inicio de la expresion a parsear
<i>end</i>	Puntero que representa el punto final de la expresion a parsear

Devuelve

Expresion que se logra parsear o null

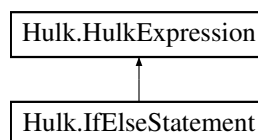
La documentación de esta clase está generada del siguiente archivo:

- C:/Users/Jossue/Cosas/proyectos y cosas/Proyectos de Programacion/H.U.L.K/Hulk/Parser.cs

5.20. Referencia de la clase Hulk.IfElseStatement

Representa las expresiones condicionales. Expresiones del tipo if([condicion]) [expresion] else [expresion].

Diagrama de herencia de Hulk.IfElseStatement

**Métodos públicos**

- **IfElseStatement** ([HulkExpression](#) Cond, [HulkExpression](#) IfExp, [HulkExpression](#) ElseExp)
Construye un objeto que representa una expresion condicional.
- override object **GetValue** (bool execute)
- override [Types](#) **CheckType** ()

Métodos públicos heredados de [Hulk.HulkExpression](#)

- object [GetValue](#) (bool execute)
Metodo que devuelve el valor de la expresion.
- [Types](#) **CheckType** ()
Chequea que no existan errores semanticos en la expresion y devuelve el tipo de retorno de la misma.

Propiedades

- [HulkExpression](#) **Condition** [get, protected set]
Expresion que representa la condicion.
- [HulkExpression](#) **IfExpression** [get, protected set]
Expresion que se ejecutara si se cumple la condicion.
- [HulkExpression](#) **ElseExpression** [get, protected set]
Expresion que se ejecutara si no se cumple la condicion.

Propiedades heredados de [Hulk.HulkExpression](#)

- **bool IsDependent** [get, protected set]

Especificacion de si la expresion depende de un parametro (argumento de funcion) o no.

5.20.1. Descripción detallada

Representa las expresiones condicionales. Expresiones del tipo if([condicion]) [expresion] else [expresion].

5.20.2. Documentación de constructores y destructores

5.20.2.1. IfElseStatement()

```
Hulk.IfElseStatement.IfElseStatement (
    HulkExpression Cond,
    HulkExpression IfExp,
    HulkExpression ElseExp )
```

Construye un objeto que representa una expresion condicional.

Parámetros

<i>Cond</i>	Expresion que representa la condicion
<i>IfExp</i>	Expresion que se ejecutara si se cumple la condicion
<i>ElseExp</i>	Expresion que se ejecutara si no se cumple la condicion

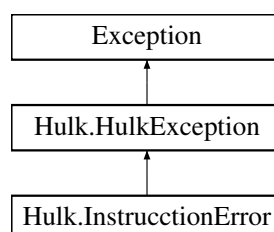
La documentación de esta clase está generada del siguiente archivo:

- C:/Users/Jossue/Cosas/proyectos y cosas/Proyectos de Programacion/H.U.L.K/Hulk/Statements.cs

5.21. Referencia de la clase Hulk.InstrucctionError

Representa los errores ocurridos en una instruccion.

Diagrama de herencia de Hulk.InstrucctionError



Métodos públicos

- **InstrucctionError** ([HulkException](#) ex, int instructionNumber, int instructionsCount)

Otros miembros heredados

Propiedades heredadas de [Hulk.HulkException](#)

- override string **Message** [get]
- string **MessageStart** [get, protected set]
Caption que inicia el mensaje de error.
- string **MessageDefinition** [get, protected set]
Descripción del error.

5.21.1. Descripción detallada

Representa los errores ocurridos en una instrucción.

La documentación de esta clase está generada del siguiente archivo:

- C:/Users/Jossue/Cosas/proyectos y cosas/Proyectos de Programacion/H.U.L.K/Hulk/Exceptions.cs

5.22. Referencia de la clase Hulk.Kompiler

Representa un compilador de HULK. Instancias de esta clase se encargaran juntar los distintos procesos de compilación.

Métodos públicos

- [Kompiler](#) ([Print](#) print)
Construye un compilador de HULK.
- void [Compile](#) (string input)
Ejecuta una entrada escrita en HULK.
- void **Clear** ()
Limpia la memoria de funciones.

Propiedades

- [HulkMemory](#) **Memory** [get]
Memoria donde se guardaran las funciones.

5.22.1. Descripción detallada

Representa un compilador de HULK. Instancias de esta clase se encargaran juntar los distintos procesos de compilación.

5.22.2. Documentación de constructores y destructores

5.22.2.1. Kompiler()

```
Hulk.Kompiler.Kompiler (
    Print print )
```

Construye un compilador de HULK.

Parámetros

<code>print</code>	Funcion que se encargara de imprimir
--------------------	--------------------------------------

5.22.3. Documentación de funciones miembro

5.22.3.1. Compile()

```
void Hulk.Kompiler.Compile (
    string input )
```

Ejecuta una entrada escrita en HULK.

Parámetros

<code>input</code>	Entrada escrita en HULK
--------------------	-------------------------

Excepciones

<code>InstrucctionError</code>	
--------------------------------	--

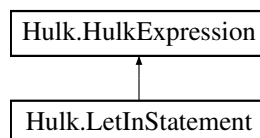
La documentación de esta clase está generada del siguiente archivo:

- C:/Users/Jossue/Cosas/proyectos y cosas/Proyectos de Programacion/H.U.L.K/Hulk/Kompiler.cs

5.23. Referencia de la clase Hulk.LetInStatement

Representa a las expresiones let-in. Expresiones del tipo let [declaraciones de variables sepaaradas por coma] in [expresion].

Diagrama de herencia de Hulk.LetInStatement



Métodos públicos

- `LetInStatement` (Dictionary< string, [Variable](#) > Variables)
Construye una expresion let-in.
- override object `GetValue` (bool execute)
- override `Types CheckType` ()
- void `Define` ([HulkExpression](#) Definition)
Asigna el cuerpo de la expresion let-in.

Métodos públicos heredados de **Hulk.HulkExpression**

- object **GetValue** (bool execute)
Método que devuelve el valor de la expresión.
- **Types CheckType** ()
Chequea que no existan errores semánticos en la expresión y devuelve el tipo de retorno de la misma.

Propiedades

- Dictionary< string, **Variable** > **StoredVariables** [get]
Variables del contexto de la expresión let-in.
- **HulkExpression Body** [get]
Cuerpo de la expresión let-in.

Propiedades heredados de **Hulk.HulkExpression**

- bool **IsDependent** [get, protected set]
Especificación de si la expresión depende de un parámetro (argumento de función) o no.

5.23.1. Descripción detallada

Representa a las expresiones let-in. Expresiones del tipo let [declaraciones de variables separadas por coma] in [expresión].

5.23.2. Documentación de constructores y destructores

5.23.2.1. LetInStatement()

```
Hulk.LetInStatement.LetInStatement (
    Dictionary< string, Variable > Variables )
```

Construye una expresión let-in.

Parámetros

<i>Variables</i>	Variables locales que se declaran en la expresión
------------------	---

5.23.3. Documentación de funciones miembro

5.23.3.1. Define()

```
void Hulk.LetInStatement.Define (
    HulkExpression Definition )
```

Asigna el cuerpo de la expresión let-in.

Parámetros

<i>Definition</i>	Cuerpo de la expresion let-in
-------------------	-------------------------------

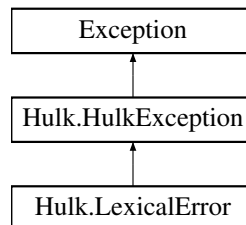
La documentación de esta clase está generada del siguiente archivo:

- C:/Users/Jossue/Cosas/proyectos y cosas/Proyectos de Programacion/H.U.L.K/Hulk/Statements.cs

5.24. Referencia de la clase Hulk.LexicalError

Representa un error lexico. Se producen por la presencia de tokens invalidos.

Diagrama de herencia de Hulk.LexicalError



Métodos públicos

- **LexicalError** (string invalidToken)
- **LexicalError** (string invalidToken, string expectedToken)

Propiedades

- string **InvalidToken** [get]
- string **ExpectedToken** [get]

Propiedades heredados de [Hulk.HulkException](#)

- override string **Message** [get]
- string **MessageStart** [get, protected set]
Caption que inicia el mensaje de error.
- string **MessageDefinition** [get, protected set]
Descripcion del error.

5.24.1. Descripción detallada

Representa un error lexico. Se producen por la presencia de tokens invalidos.

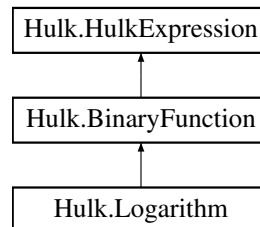
La documentación de esta clase está generada del siguiente archivo:

- C:/Users/Jossue/Cosas/proyectos y cosas/Proyectos de Programacion/H.U.L.K/Hulk/Exceptions.cs

5.25. Referencia de la clase Hulk.Logarithm

Clase que representa el llamado a la funcion logaritmo. Expresion del tipo `log([expresion],[expresion])`

Diagrama de herencia de Hulk.Logarithm



Métodos públicos

- **Logarithm** ([HulkExpression](#) leftArgument, [HulkExpression](#) rightArgument)

Métodos públicos heredados de [Hulk.BinaryFunction](#)

- **BinaryFunction** ([HulkExpression](#) leftArgument, [HulkExpression](#) rightArgument)
Contructor de un objeto que representa a una funcion binaria.
- override object **GetValue** (bool execute)
- override [Types](#) **CheckType** ()
- object [Evaluate](#) (object left, object right)
Evalua la expresion.
- delegate object **BinaryOperation** (object left, object right)

Métodos públicos heredados de [Hulk.HulkExpression](#)

- object [GetValue](#) (bool execute)
Metodo que devuelve el valor de la expresion.
- [Types](#) **CheckType** ()
Chequea que no existan errores semanticos en la expresion y devuelve el tipo de retorno de la misma.

Otros miembros heredados

Propiedades heredados de [Hulk.BinaryFunction](#)

- [HulkExpression](#) **LeftArgument** [get, protected set]
Expresion que representa al miembro izquierdo de la expresion.
- [HulkExpression](#) **RightArgument** [get, protected set]
Expresion que representa al miembro derecho de la expresion.
- [Types](#) **ReturnedType** [get, protected set]
Tipo de retorno aceptado.
- [Types](#) **EnteredType** [get, protected set]
Tipo de extrada aceptado.
- [Type](#) **AcceptedType** [get, protected set]
Tipo aceptado del objeto que devuelven los argumentos.
- string **OperationToken** [get, protected set]
- BinaryOperation **Operation** [get, protected set]

Propiedades heredados de [Hulk.HulkExpression](#)

- bool **IsDependent** [get, protected set]
Especificacion de si la expresion depende de un parametro (argumento de funcion) o no.

5.25.1. Descripción detallada

Clase que representa el llamado a la funcion logaritmo. Expresion del tipo `log([expresion],[expresion])`

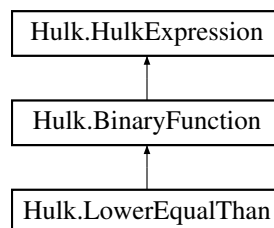
La documentación de esta clase está generada del siguiente archivo:

- `C:/Users/Jossue/Cosas/proyectos y cosas/Proyectos de Programacion/H.U.L.K/Hulk/BinaryOperations.cs`

5.26. Referencia de la clase Hulk.LowerEqualThan

Clase que representa la operacion menor o igual que. Expresion del tipo `[expresion] <= [expresion]`.

Diagrama de herencia de Hulk.LowerEqualThan



Métodos públicos

- **LowerEqualThan** ([HulkExpression](#) leftArgument, [HulkExpression](#) rightArgument)

Métodos públicos heredados de [Hulk.BinaryFunction](#)

- [BinaryFunction](#) ([HulkExpression](#) leftArgument, [HulkExpression](#) rightArgument)
Constructor de un objeto que representa a una funcion binaria.
- override object **GetValue** (bool execute)
- override [Types](#) **CheckType** ()
- object [Evaluate](#) (object left, object right)
Evalua la expresion.
- delegate object **BinaryOperation** (object left, object right)

Métodos públicos heredados de [Hulk.HulkExpression](#)

- object [GetValue](#) (bool execute)
Metodo que devuelve el valor de la expresion.
- [Types](#) **CheckType** ()
Chequea que no existan errores semanticos en la expresion y devuelve el tipo de retorno de la misma.

Otros miembros heredados

Propiedades heredados de **Hulk.BinaryFunction**

- **HulkExpression LeftArgument** [get, protected set]
Expresion que representa al miembro izquierdo de la expresion.
- **HulkExpression RightArgument** [get, protected set]
Expresion que representa al miembro derecho de la expresion.
- **Types ReturnedType** [get, protected set]
Tipo de retorno aceptado.
- **Types EnteredType** [get, protected set]
Tipo de extrada aceptado.
- **Type AcceptedType** [get, protected set]
Tipo aceptado del objeto que devuelven los argumentos.
- string **OperationToken** [get, protected set]
- BinaryOperation **Operation** [get, protected set]

Propiedades heredados de **Hulk.HulkExpression**

- bool **IsDependent** [get, protected set]
Especificacion de si la expresion depende de un parametro (argumento de funcion) o no.

5.26.1. Descripción detallada

Clase que representa la operacion menor o igual que. Expresion del tipo [expresion] <= [expresion].

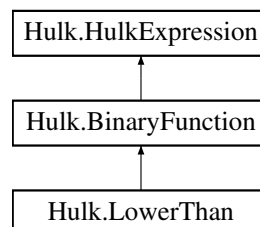
La documentación de esta clase está generada del siguiente archivo:

- C:/Users/Jossue/Cosas/proyectos y cosas/Proyectos de Programacion/H.U.L.K/Hulk/BinaryOperations.cs

5.27. Referencia de la clase **Hulk.LowerThan**

Clase que representa la operacion menor que. Expresion del tipo [expresion] < [expresion].

Diagrama de herencia de Hulk.LowerThan



Métodos públicos

- **LowerThan** (**HulkExpression** leftArgument, **HulkExpression** rightArgument)

Métodos públicos heredados de [Hulk.BinaryFunction](#)

- [BinaryFunction](#) ([HulkExpression](#) leftArgument, [HulkExpression](#) rightArgument)
Constructor de un objeto que representa a una funcion binaria.
- override object **GetValue** (bool execute)
- override [Types](#) **CheckType** ()
- object [Evaluate](#) (object left, object right)
Evalua la expresion.
- delegate object **BinaryOperation** (object left, object right)

Métodos públicos heredados de [Hulk.HulkExpression](#)

- object [GetValue](#) (bool execute)
Metodo que devuelve el valor de la expresion.
- [Types](#) **CheckType** ()
Chequea que no existan errores semanticos en la expresion y devuelve el tipo de retorno de la misma.

Otros miembros heredados

Propiedades heredados de [Hulk.BinaryFunction](#)

- [HulkExpression](#) **LeftArgument** [get, protected set]
Expresion que representa al miembro izquierdo de la expresion.
- [HulkExpression](#) **RightArgument** [get, protected set]
Expresion que representa al miembro derecho de la expresion.
- [Types](#) **ReturnedType** [get, protected set]
Tipo de retorno aceptado.
- [Types](#) **EnteredType** [get, protected set]
Tipo de extrada aceptado.
- Type **AcceptedType** [get, protected set]
Tipo aceptado del objeto que devuelven los argumentos.
- string **OperationToken** [get, protected set]
- BinaryOperation **Operation** [get, protected set]

Propiedades heredados de [Hulk.HulkExpression](#)

- bool **IsDependent** [get, protected set]
Especificacion de si la expresion depende de un parametro (argumento de funcion) o no.

5.27.1. Descripción detallada

Clase que representa la operacion menor que. Expresion del tipo [expresion] < [expresion].

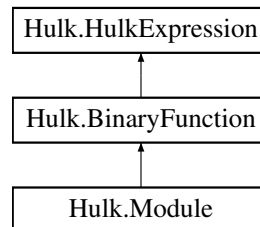
La documentación de esta clase está generada del siguiente archivo:

- C:/Users/Jossue/Cosas/proyectos y cosas/Proyectos de Programacion/H.U.L.K/Hulk/BinaryOperations.cs

5.28. Referencia de la clase Hulk.Module

Clase que representa la operacion modulo o resto. Expresion del tipo [expresion] % [expresion].

Diagrama de herencia de Hulk.Module



Métodos públicos

- **Module** ([HulkExpression](#) leftArgument, [HulkExpression](#) rightArgument)

Métodos públicos heredados de [Hulk.BinaryFunction](#)

- [BinaryFunction](#) ([HulkExpression](#) leftArgument, [HulkExpression](#) rightArgument)
Contructor de un objeto que representa a una funcion binaria.
- override object **GetValue** (bool execute)
- override [Types](#) **CheckType** ()
- object [Evaluate](#) (object left, object right)
Evalua la expresion.
- delegate object **BinaryOperation** (object left, object right)

Métodos públicos heredados de [Hulk.HulkExpression](#)

- object [GetValue](#) (bool execute)
Metodo que devuelve el valor de la expresion.
- [Types](#) **CheckType** ()
Chequea que no existan errores semanticos en la expresion y devuelve el tipo de retorno de la misma.

Otros miembros heredados

Propiedades heredados de [Hulk.BinaryFunction](#)

- [HulkExpression](#) **LeftArgument** [get, protected set]
Expresion que representa al miembro izquierdo de la expresion.
- [HulkExpression](#) **RightArgument** [get, protected set]
Expresion que representa al miembro derecho de la expresion.
- [Types](#) **ReturnedType** [get, protected set]
Tipo de retorno aceptado.
- [Types](#) **EnteredType** [get, protected set]
Tipo de extrada aceptado.
- Type **AcceptedType** [get, protected set]
Tipo aceptado del objeto que devuelven los argumentos.
- string **OperationToken** [get, protected set]
- BinaryOperation **Operation** [get, protected set]

Propiedades heredados de [Hulk.HulkExpression](#)

- bool **IsDependent** [get, protected set]
Especificacion de si la expresion depende de un parametro (argumento de funcion) o no.

5.28.1. Descripción detallada

Clase que representa la operacion modulo o resto. Expresion del tipo [expresion] % [expresion].

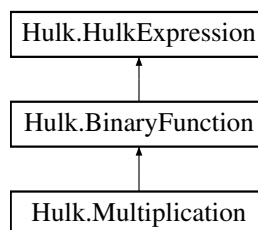
La documentación de esta clase está generada del siguiente archivo:

- C:/Users/Jossue/Cosas/proyectos y cosas/Proyectos de Programacion/H.U.L.K/Hulk/BinaryOperations.cs

5.29. Referencia de la clase Hulk.Multiplication

Clase que representa la multiplicacion. Expresion del tipo [expresion] * [expresion].

Diagrama de herencia de Hulk.Multiplication



Métodos públicos

- **Multiplication** ([HulkExpression](#) leftArgument, [HulkExpression](#) rightArgument)

Métodos públicos heredados de [Hulk.BinaryFunction](#)

- [BinaryFunction](#) ([HulkExpression](#) leftArgument, [HulkExpression](#) rightArgument)
Constructor de un objeto que representa a una funcion binaria.
- override object **GetValue** (bool execute)
- override [Types](#) **CheckType** ()
- object [Evaluate](#) (object left, object right)
Evalua la expresion.
- delegate object **BinaryOperation** (object left, object right)

Métodos públicos heredados de [Hulk.HulkExpression](#)

- object [GetValue](#) (bool execute)
Metodo que devuelve el valor de la expresion.
- [Types](#) **CheckType** ()
Chequea que no existan errores semanticos en la expresion y devuelve el tipo de retorno de la misma.

Otros miembros heredados

Propiedades heredados de **Hulk.BinaryFunction**

- **HulkExpression LeftArgument** [get, protected set]
Expresion que representa al miembro izquierdo de la expresion.
- **HulkExpression RightArgument** [get, protected set]
Expresion que representa al miembro derecho de la expresion.
- **Types ReturnedType** [get, protected set]
Tipo de retorno aceptado.
- **Types EnteredType** [get, protected set]
Tipo de extrada aceptado.
- **Type AcceptedType** [get, protected set]
Tipo aceptado del objeto que devuelven los argumentos.
- string **OperationToken** [get, protected set]
- BinaryOperation **Operation** [get, protected set]

Propiedades heredados de **Hulk.HulkExpression**

- bool **IsDependent** [get, protected set]
Especificacion de si la expresion depende de un parametro (argumento de funcion) o no.

5.29.1. Descripción detallada

Clase que representa la multiplicacion. Expresion del tipo [expresion] * [expresion].

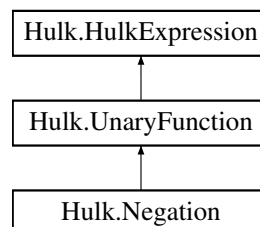
La documentación de esta clase está generada del siguiente archivo:

- C:/Users/Jossue/Cosas/proyectos y cosas/Proyectos de Programacion/H.U.L.K/Hulk/BinaryOperations.cs

5.30. Referencia de la clase **Hulk.Negation**

Representa la negacion logica. Expresiones del tipo ![expresion].

Diagrama de herencia de Hulk.Negation



Métodos públicos

- **Negation** (HulkExpression Arg)

Métodos públicos heredados de Hulk.UnaryFunction

- **UnaryFunction** (HulkExpression Arg)
Construye un objeto que representa una operacion unaria.
- override object **GetValue** (bool execute)
- delegate object **UnaryOperation** (object arg)
Funcion unaria delegada.
- object **Evaluate** (object arg)
Evalua la operacion unaria.
- override **Types CheckType** ()

Métodos públicos heredados de Hulk.HulkExpression

- object **GetValue** (bool execute)
Metodo que devuelve el valor de la expresion.
- **Types CheckType** ()
Chequea que no existan errores semanticos en la expresion y devuelve el tipo de retorno de la misma.

Otros miembros heredados

Propiedades heredados de Hulk.UnaryFunction

- **HulkExpression Argument** [get, protected set]
- **Types ReturnedType** [get, protected set]
Tipo de retorno de la operacion.
- **Types EnteredType** [get, protected set]
Tipo de entrada de la operacion aceptado.
- **Type AcceptedType** [get, protected set]
Tipo de valor aceptado para el valor del argumento.
- string **OperationToken** [get, protected set]
Token de las operacion.
- **UnaryOperation Operation** [get, protected set]
Funcion que efectua la operacion unaria.

Propiedades heredados de Hulk.HulkExpression

- bool **IsDependent** [get, protected set]
Especificacion de si la expresion depende de un parametro (argumento de funcion) o no.

5.30.1. Descripción detallada

Representa la negacion logica. Expresiones del tipo ![expresion].

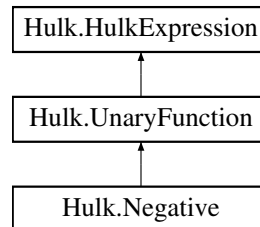
La documentación de esta clase está generada del siguiente archivo:

- C:/Users/Jossue/Cosas/proyectos y cosas/Proyectos de Programacion/H.U.L.K/Hulk/UnaryOperations.cs

5.31. Referencia de la clase Hulk.Negative

Representa la operacion unaria negativa. Expresiones del tipo `-[expresion]`.

Diagrama de herencia de Hulk.Negative



Métodos públicos

- **Negative** ([HulkExpression](#) Arg)

Métodos públicos heredados de [Hulk.UnaryFunction](#)

- **UnaryFunction** ([HulkExpression](#) Arg)
Construye un objeto que representa una operacion unaria.
- override object **GetValue** (bool execute)
- delegate object **UnaryOperation** (object arg)
Funcion unaria delegada.
- object **Evaluate** (object arg)
Evalua la operacion unaria.
- override **Types CheckType** ()

Métodos públicos heredados de [Hulk.HulkExpression](#)

- object **GetValue** (bool execute)
Metodo que devuelve el valor de la expresion.
- **Types CheckType** ()
Chequea que no existan errores semanticos en la expresion y devuelve el tipo de retorno de la misma.

Otros miembros heredados

Propiedades heredados de [Hulk.UnaryFunction](#)

- **HulkExpression Argument** [get, protected set]
- **Types ReturnedType** [get, protected set]
Tipo de retorno de la operacion.
- **Types EnteredType** [get, protected set]
Tipo de entrada de la operacion aceptado.
- **Type AcceptedType** [get, protected set]
Tipo de valor aceptado para el valor del argumento.
- string **OperationToken** [get, protected set]
Token de las operacion.
- **UnaryOperation Operation** [get, protected set]
Funcion que efectua la operacion unaria.

Propiedades heredados de [Hulk.HulkExpression](#)

- bool **IsDependent** [get, protected set]
Especificacion de si la expresion depende de un parametro (argumento de funcion) o no.

5.31.1. Descripción detallada

Representa la operacion unaria negativa. Expresiones del tipo `-[expresion]`.

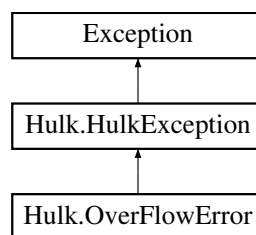
La documentación de esta clase está generada del siguiente archivo:

- `C:/Users/Jossue/Cosas/proyectos y cosas/Proyectos de Programacion/H.U.L.K/Hulk/UnaryOperations.cs`

5.32. Referencia de la clase Hulk.OverflowError

Representa un error producido por la sobrecarga de la pila de llamados.

Diagrama de herencia de Hulk.OverflowError



Métodos públicos

- **OverflowError** (string functionName)

Propiedades

- string **FunctionName** [get]

Propiedades heredados de [Hulk.HulkException](#)

- override string **Message** [get]
- string **MessageStart** [get, protected set]
Caption que inicia el mensaje de error.
- string **MessageDefinition** [get, protected set]
Descripcion del error.

5.32.1. Descripción detallada

Representa un error producido por la sobrecarga de la pila de llamados.

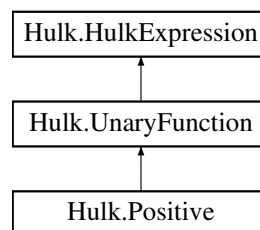
La documentación de esta clase está generada del siguiente archivo:

- C:/Users/Jossue/Cosas/proyectos y cosas/Proyectos de Programacion/H.U.L.K/Hulk/Exceptions.cs

5.33. Referencia de la clase Hulk.Positive

Representa la operacion unaria positiva. Expresiones del tipo +[expresion].

Diagrama de herencia de Hulk.Positive



Métodos públicos

- **Positive** ([HulkExpression](#) Arg)

Métodos públicos heredados de [Hulk.UnaryFunction](#)

- [UnaryFunction](#) ([HulkExpression](#) Arg)
Construye un objeto que representa una operacion unaria.
- override object **GetValue** (bool execute)
- delegate object [UnaryOperation](#) (object arg)
Funcion unaria delegada.
- object [Evaluate](#) (object arg)
Evalua la operacion unaria.
- override [Types](#) **CheckType** ()

Métodos públicos heredados de [Hulk.HulkExpression](#)

- object [GetValue](#) (bool execute)
Metodo que devuelve el valor de la expresion.
- [Types](#) **CheckType** ()
Chequea que no existan errores semanticos en la expresion y devuelve el tipo de retorno de la misma.

Otros miembros heredados

Propiedades heredados de Hulk.UnaryFunction

- **HulkExpression Argument** [get, protected set]
Tipo de retorno de la operacion.
- **Types ReturnedType** [get, protected set]
Tipo de entrada de la operacion aceptado.
- **Types EnteredType** [get, protected set]
Tipo de valor aceptado para el valor del argumento.
- **Type AcceptedType** [get, protected set]
Token de las operacion.
- **UnaryOperation Operation** [get, protected set]
Funcion que efectua la operacion unaria.

Propiedades heredados de Hulk.HulkExpression

- **bool IsDependent** [get, protected set]
Especificacion de si la expresion depende de un parametro (argumento de funcion) o no.

5.33.1. Descripción detallada

Representa la operacion unaria positiva. Expresiones del tipo $+[expresion]$.

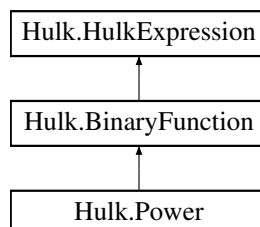
La documentación de esta clase está generada del siguiente archivo:

- C:/Users/Jossue/Cosas/proyectos y cosas/Proyectos de Programacion/H.U.L.K/Hulk/UnaryOperations.cs

5.34. Referencia de la clase Hulk.Power

Clase que representa la exponenciacion. Expresion del tipo $[expresion] ^ [expresion]$.

Diagrama de herencia de Hulk.Power



Métodos públicos

- **Power** (HulkExpression leftArgument, HulkExpression rightArgument)

Métodos públicos heredados de **Hulk.BinaryFunction**

- **BinaryFunction** (**HulkExpression** leftArgument, **HulkExpression** rightArgument)
Constructor de un objeto que representa a una funcion binaria.
- override object **GetValue** (bool execute)
- override **Types CheckType** ()
- object **Evaluate** (object left, object right)
Evalua la expresion.
- delegate object **BinaryOperation** (object left, object right)

Métodos públicos heredados de **Hulk.HulkExpression**

- object **GetValue** (bool execute)
Metodo que devuelve el valor de la expresion.
- **Types CheckType** ()
Chequea que no existan errores semanticos en la expresion y devuelve el tipo de retorno de la misma.

Otros miembros heredados

Propiedades heredados de **Hulk.BinaryFunction**

- **HulkExpression LeftArgument** [get, protected set]
Expresion que representa al miembro izquierdo de la expresion.
- **HulkExpression RightArgument** [get, protected set]
Expresion que representa al miembro derecho de la expresion.
- **Types ReturnedType** [get, protected set]
Tipo de retorno aceptado.
- **Types EnteredType** [get, protected set]
Tipo de extrada aceptado.
- **Type AcceptedType** [get, protected set]
Tipo aceptado del objeto que devuelven los argumentos.
- string **OperationToken** [get, protected set]
- BinaryOperation **Operation** [get, protected set]

Propiedades heredados de **Hulk.HulkExpression**

- bool **IsDependent** [get, protected set]
Especificacion de si la expresion depende de un parametro (argumento de funcion) o no.

5.34.1. Descripción detallada

Clase que representa la exponenciacion. Expresion del tipo [expresion] ^ [expresion].

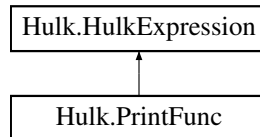
La documentación de esta clase está generada del siguiente archivo:

- C:/Users/Jossue/Cosas/proyectos y cosas/Proyectos de Programacion/H.U.L.K/Hulk/BinaryOperations.cs

5.35. Referencia de la clase Hulk.PrintFunc

Clase que representa a la funcion de imprimir en consola. Expresion del tipo `print([expresion])`

Diagrama de herencia de Hulk.PrintFunc



Métodos públicos

- `PrintFunc` (`HulkExpression` Arg, `Print` printHandler)
Constructor de un objeto que representa un llamado a la funcion print()
- override object `GetValue` (bool execute)
- override `Types CheckType` ()

Métodos públicos heredados de Hulk.HulkExpression

- object `GetValue` (bool execute)
Método que devuelve el valor de la expresion.
- `Types CheckType` ()
Chequea que no existan errores semanticos en la expresion y devuelve el tipo de retorno de la misma.

Propiedades

- `HulkExpression Argument` [get]
Expresion a imprimir.

Propiedades heredados de Hulk.HulkExpression

- bool `IsDependent` [get, protected set]
Especificacion de si la expresion depende de un parametro (argumento de funcion) o no.

5.35.1. Descripción detallada

Clase que representa a la funcion de imprimir en consola. Expresion del tipo `print([expresion])`

5.35.2. Documentación de constructores y destructores

5.35.2.1. PrintFunc()

```

Hulk.PrintFunc.PrintFunc (
    HulkExpression Arg,
    Print printHandler )
  
```

Constructor de un objeto que representa un llamado a la funcion print()

Parámetros

<i>Arg</i>	Argumento de la funcion
<i>printHandler</i>	Funcion encargada de mostrar la salida en la interfaz

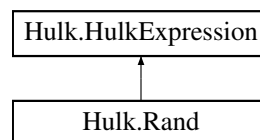
La documentación de esta clase está generada del siguiente archivo:

- C:/Users/Jossue/Cosas/proyectos y cosas/Proyectos de Programacion/H.U.L.K/Hulk/BasicExpressions.cs

5.36. Referencia de la clase Hulk.Rand

Representa el llamado a la funcion random. Expresiones del tipo rand([expresion])

Diagrama de herencia de Hulk.Rand



Métodos públicos

- override [Types CheckType](#) ()
- override object **GetValue** (bool execute)

Métodos públicos heredados de [Hulk.HulkExpression](#)

- object [GetValue](#) (bool execute)
Método que devuelve el valor de la expresion.
- [Types CheckType](#) ()
Chequea que no existan errores semanticos en la expresion y devuelve el tipo de retorno de la misma.

Otros miembros heredados

Propiedades heredados de [Hulk.HulkExpression](#)

- bool **IsDependent** [get, protected set]
Especificacion de si la expresion depende de un parametro (argumento de funcion) o no.

5.36.1. Descripción detallada

Representa el llamado a la funcion random. Expresiones del tipo rand([expresion])

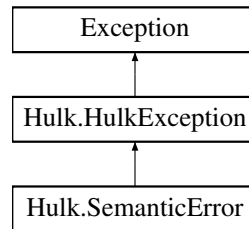
La documentación de esta clase está generada del siguiente archivo:

- C:/Users/Jossue/Cosas/proyectos y cosas/Proyectos de Programacion/H.U.L.K/Hulk/UnaryOperations.cs

5.37. Referencia de la clase Hulk.SemanticError

Representa un error semantico. Se produce por el uso incorrecto de los tipos y argumentos.

Diagrama de herencia de Hulk.SemanticError



Métodos públicos

- **SemanticError** (string expression, string expected, string received)

Propiedades

- string **Expression** [get]
- string **ExpressionReceived** [get]
- string **ExpressionExpected** [get]

Propiedades heredados de Hulk.HulkException

- override string **Message** [get]
- string **MessageStart** [get, protected set]
Caption que inicia el mensaje de error.
- string **MessageDefinition** [get, protected set]
Descripcion del error.

5.37.1. Descripción detallada

Representa un error semantico. Se produce por el uso incorrecto de los tipos y argumentos.

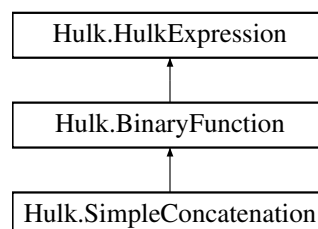
La documentación de esta clase está generada del siguiente archivo:

- C:/Users/Jossue/Cosas/proyectos y cosas/Proyectos de Programacion/H.U.L.K/Hulk/Exceptions.cs

5.38. Referencia de la clase Hulk.SimpleConcatenation

Clase que representa la concatenacion de expresiones. Expresion del tipo [expresion] @ [expresion].

Diagrama de herencia de Hulk.SimpleConcatenation



Métodos públicos

- **SimpleConcatenation** ([HulkExpression](#) leftArgument, [HulkExpression](#) rightArgument)

Métodos públicos heredados de [Hulk.BinaryFunction](#)

- **BinaryFunction** ([HulkExpression](#) leftArgument, [HulkExpression](#) rightArgument)
Constructor de un objeto que representa a una funcion binaria.
- override object **GetValue** (bool execute)
- override [Types](#) **CheckType** ()
- object **Evaluate** (object left, object right)
Evalua la expresion.
- delegate object **BinaryOperation** (object left, object right)

Métodos públicos heredados de [Hulk.HulkExpression](#)

- object **GetValue** (bool execute)
Metodo que devuelve el valor de la expresion.
- [Types](#) **CheckType** ()
Chequea que no existan errores semanticos en la expresion y devuelve el tipo de retorno de la misma.

Otros miembros heredados

Propiedades heredadas de [Hulk.BinaryFunction](#)

- [HulkExpression](#) **LeftArgument** [get, protected set]
Expresion que representa al miembro izquierdo de la expresion.
- [HulkExpression](#) **RightArgument** [get, protected set]
Expresion que representa al miembro derecho de la expresion.
- [Types](#) **ReturnedType** [get, protected set]
Tipo de retorno aceptado.
- [Types](#) **EnteredType** [get, protected set]
Tipo de extrada aceptado.
- [Type](#) **AcceptedType** [get, protected set]
Tipo aceptado del objeto que devuelven los argumentos.
- string **OperationToken** [get, protected set]
- BinaryOperation **Operation** [get, protected set]

Propiedades heredadas de [Hulk.HulkExpression](#)

- bool **IsDependent** [get, protected set]
Especificacion de si la expresion depende de un parametro (argumento de funcion) o no.

5.38.1. Descripción detallada

Clase que representa la concatenacion de expresiones. Expresion del tipo [expresion] @ [expresion].

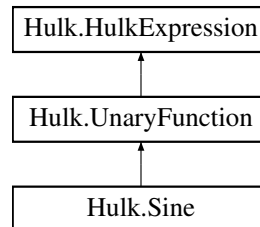
La documentación de esta clase está generada del siguiente archivo:

- C:/Users/Jossue/Cosas/proyectos y cosas/Proyectos de Programacion/H.U.L.K/Hulk/BinaryOperations.cs

5.39. Referencia de la clase Hulk.Sine

Representa el llamado a la funcion seno. Expresiones del tipo sin([expresion])

Diagrama de herencia de Hulk.Sine



Métodos públicos

- **Sine** ([HulkExpression](#) Arg)

Métodos públicos heredados de [Hulk.UnaryFunction](#)

- **UnaryFunction** ([HulkExpression](#) Arg)
Construye un objeto que representa una operacion unaria.
- override object **GetValue** (bool execute)
- delegate object **UnaryOperation** (object arg)
Funcion unaria delegada.
- object **Evaluate** (object arg)
Evalua la operacion unaria.
- override **Types CheckType** ()

Métodos públicos heredados de [Hulk.HulkExpression](#)

- object **GetValue** (bool execute)
Metodo que devuelve el valor de la expresion.
- **Types CheckType** ()
Chequea que no existan errores semanticos en la expresion y devuelve el tipo de retorno de la misma.

Otros miembros heredados

Propiedades heredados de [Hulk.UnaryFunction](#)

- **HulkExpression Argument** [get, protected set]
- **Types ReturnedType** [get, protected set]
Tipo de retorno de la operacion.
- **Types EnteredType** [get, protected set]
Tipo de entrada de la operacion aceptado.
- **Type AcceptedType** [get, protected set]
Tipo de valor aceptado para el valor del argumento.
- string **OperationToken** [get, protected set]
Token de las operacion.
- **UnaryOperation Operation** [get, protected set]
Funcion que efectua la operacion unaria.

Propiedades heredados de [Hulk.HulkExpression](#)

- bool **IsDependent** [get, protected set]
Especificacion de si la expresion depende de un parametro (argumento de funcion) o no.

5.39.1. Descripción detallada

Representa el llamado a la funcion seno. Expresiones del tipo `sin([expresion])`

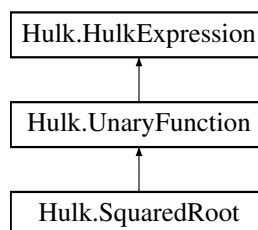
La documentación de esta clase está generada del siguiente archivo:

- `C:/Users/Jossue/Cosas/proyectos y cosas/Proyectos de Programacion/H.U.L.K/Hulk/UnaryOperations.cs`

5.40. Referencia de la clase [Hulk.SquaredRoot](#)

Representa el llamado a la funcion raiz cuadrada. Expresiones del tipo `sqrt([expresion])`

Diagrama de herencia de [Hulk.SquaredRoot](#)



Métodos públicos

- **SquaredRoot** ([HulkExpression](#) Arg)

Métodos públicos heredados de [Hulk.UnaryFunction](#)

- [UnaryFunction](#) ([HulkExpression](#) Arg)
Construye un objeto que representa una operacion unaria.
- override object **GetValue** (bool execute)
- delegate object [UnaryOperation](#) (object arg)
Funcion unaria delegada.
- object [Evaluate](#) (object arg)
Evalua la operacion unaria.
- override [Types](#) **CheckType** ()

Métodos públicos heredados de [Hulk.HulkExpression](#)

- object [GetValue](#) (bool execute)
Metodo que devuelve el valor de la expresion.
- [Types](#) **CheckType** ()
Chequea que no existan errores semanticos en la expresion y devuelve el tipo de retorno de la misma.

Otros miembros heredados

Propiedades heredados de Hulk.UnaryFunction

- **HulkExpression Argument** [get, protected set]
Tipo de retorno de la operacion.
- **Types ReturnedType** [get, protected set]
Tipo de entrada de la operacion aceptado.
- **Types EnteredType** [get, protected set]
Tipo de valor aceptado para el valor del argumento.
- **Type AcceptedType** [get, protected set]
Token de las operacion.
- **UnaryOperation Operation** [get, protected set]
Funcion que efectua la operacion unaria.

Propiedades heredados de Hulk.HulkExpression

- **bool IsDependent** [get, protected set]
Especificacion de si la expresion depende de un parametro (argumento de funcion) o no.

5.40.1. Descripción detallada

Representa el llamado a la funcion raiz cuadrada. Expresiones del tipo `sqrt([expresion])`

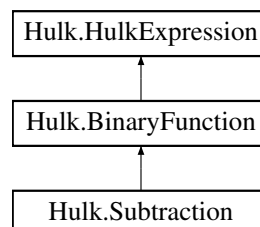
La documentación de esta clase está generada del siguiente archivo:

- `C:/Users/Jossue/Cosas/proyectos y cosas/Proyectos de Programacion/H.U.L.K/Hulk/UnaryOperations.cs`

5.41. Referencia de la clase Hulk.Subtraction

Clase que representa la sustraccion. Expresion del tipo `[expresion] - [expresion]`.

Diagrama de herencia de Hulk.Subtraction



Métodos públicos

- **Subtraction** (HulkExpression leftArgument, HulkExpression rightArgument)

Métodos públicos heredados de **Hulk.BinaryFunction**

- **BinaryFunction** (**HulkExpression** leftArgument, **HulkExpression** rightArgument)
Constructor de un objeto que representa a una funcion binaria.
- override object **GetValue** (bool execute)
- override **Types CheckType** ()
- object **Evaluate** (object left, object right)
Evalua la expresion.
- delegate object **BinaryOperation** (object left, object right)

Métodos públicos heredados de **Hulk.HulkExpression**

- object **GetValue** (bool execute)
Metodo que devuelve el valor de la expresion.
- **Types CheckType** ()
Chequea que no existan errores semanticos en la expresion y devuelve el tipo de retorno de la misma.

Otros miembros heredados

Propiedades heredados de **Hulk.BinaryFunction**

- **HulkExpression LeftArgument** [get, protected set]
Expresion que representa al miembro izquierdo de la expresion.
- **HulkExpression RightArgument** [get, protected set]
Expresion que representa al miembro derecho de la expresion.
- **Types ReturnedType** [get, protected set]
Tipo de retorno aceptado.
- **Types EnteredType** [get, protected set]
Tipo de extrada aceptado.
- **Type AcceptedType** [get, protected set]
Tipo aceptado del objeto que devuelven los argumentos.
- string **OperationToken** [get, protected set]
- BinaryOperation **Operation** [get, protected set]

Propiedades heredados de **Hulk.HulkExpression**

- bool **IsDependent** [get, protected set]
Especificacion de si la expresion depende de un parametro (argumento de funcion) o no.

5.41.1. Descripción detallada

Clase que representa la sustraccion. Expresion del tipo [expresion] - [expresion].

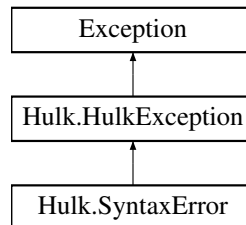
La documentación de esta clase está generada del siguiente archivo:

- C:/Users/Jossue/Cosas/proyectos y cosas/Proyectos de Programacion/H.U.L.K/Hulk/BinaryOperations.cs

5.42. Referencia de la clase Hulk.SyntaxError

Representa un error sintactico. Se produce por expresiones incompletas.

Diagrama de herencia de Hulk.SyntaxError



Métodos públicos

- **SyntaxError** (string missingPart, string place)

Propiedades

- string **MissingPart** [get]
- string **MissingPlace** [get]

Propiedades heredados de Hulk.HulkException

- override string **Message** [get]
- string **MessageStart** [get, protected set]
Caption que inicia el mensaje de error.
- string **MessageDefinition** [get, protected set]
Descripcion del error.

5.42.1. Descripción detallada

Representa un error sintactico. Se produce por expresiones incompletas.

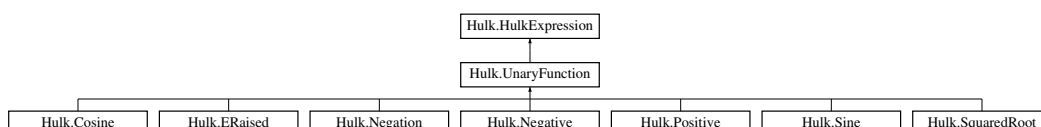
La documentación de esta clase está generada del siguiente archivo:

- C:/Users/Jossue/Cosas/proyectos y cosas/Proyectos de Programacion/H.U.L.K/Hulk/Exceptions.cs

5.43. Referencia de la clase Hulk.UnaryFunction

Representa las operaciones unarias. Expresiones del tipo [operacion] [expresion].

Diagrama de herencia de Hulk.UnaryFunction



Métodos públicos

- **UnaryFunction** (**HulkExpression** Arg)
Construye un objeto que representa una operacion unaria.
- override object **GetValue** (bool execute)
- delegate object **UnaryOperation** (object arg)
Funcion unaria delegada.
- object **Evaluate** (object arg)
Evalua la operacion unaria.
- override **Types CheckType** ()

Métodos públicos heredados de **Hulk.HulkExpression**

- object **GetValue** (bool execute)
Metodo que devuelve el valor de la expresion.
- **Types CheckType** ()
Chequea que no existan errores semanticos en la expresion y devuelve el tipo de retorno de la misma.

Propiedades

- **HulkExpression Argument** [get, protected set]
- **Types ReturnedType** [get, protected set]
Tipo de retorno de la operacion.
- **Types EnteredType** [get, protected set]
Tipo de entrada de la operacion aceptado.
- **Type AcceptedType** [get, protected set]
Tipo de valor aceptado para el valor del argumento.
- string **OperationToken** [get, protected set]
Token de las operacion.
- **UnaryOperation Operation** [get, protected set]
Funcion que efectua la operacion unaria.

Propiedades heredados de **Hulk.HulkExpression**

- bool **IsDependent** [get, protected set]
Especificacion de si la expresion depende de un parametro (argumento de funcion) o no.

5.43.1. Descripción detallada

Representa las operaciones unarias. Expresiones del tipo [operacion] [expresion].

5.43.2. Documentación de constructores y destructores

5.43.2.1. UnaryFunction()

```
Hulk.UnaryFunction.UnaryFunction (
    HulkExpression Arg )
```

Construye un objeto que representa una operacion unaria.

Parámetros

<i>Arg</i>	Argumento de la funcion
------------	-------------------------

5.43.3. Documentación de funciones miembro

5.43.3.1. Evaluate()

```
object Hulk.UnaryFunction.Evaluate (
    object arg )
```

Evalua la operacion unaria.

Parámetros

<i>arg</i>	Valor del argumento de la funcion
------------	-----------------------------------

Devuelve

Valor de retorno de la operacion unaria

Excepciones

<i>SemanticError</i>	
----------------------	--

5.43.3.2. UnaryOperation()

```
delegate object Hulk.UnaryFunction.UnaryOperation (
    object arg )
```

Funcion unaria delegada.

Parámetros

<i>arg</i>	Argumento
------------	-----------

Devuelve

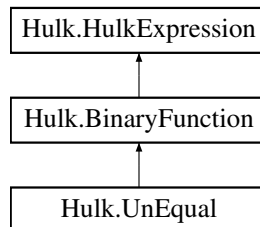
La documentación de esta clase está generada del siguiente archivo:

- C:/Users/Jossue/Cosas/proyectos y cosas/Proyectos de Programacion/H.U.L.K/Hulk/UnaryOperations.cs

5.44. Referencia de la clase Hulk.UnEqual

Clase que representa la operacion desigualdad que. Expresion del tipo [expresion] != [expresion].

Diagrama de herencia de Hulk.UnEqual



Métodos públicos

- **Unequal** ([HulkExpression](#) leftArgument, [HulkExpression](#) rightArgument)

Métodos públicos heredados de [Hulk.BinaryFunction](#)

- [BinaryFunction](#) ([HulkExpression](#) leftArgument, [HulkExpression](#) rightArgument)
Contructor de un objeto que representa a una funcion binaria.
- override object **GetValue** (bool execute)
- override [Types](#) **CheckType** ()
- object [Evaluate](#) (object left, object right)
Evalua la expresion.
- delegate object **BinaryOperation** (object left, object right)

Métodos públicos heredados de [Hulk.HulkExpression](#)

- object [GetValue](#) (bool execute)
Metodo que devuelve el valor de la expresion.
- [Types](#) **CheckType** ()
Chequea que no existan errores semanticos en la expresion y devuelve el tipo de retorno de la misma.

Otros miembros heredados

Propiedades heredados de [Hulk.BinaryFunction](#)

- [HulkExpression](#) **LeftArgument** [get, protected set]
Expresion que representa al miembro izquierdo de la expresion.
- [HulkExpression](#) **RightArgument** [get, protected set]
Expresion que representa al miembro derecho de la expresion.
- [Types](#) **ReturnedType** [get, protected set]
Tipo de retorno aceptado.
- [Types](#) **EnteredType** [get, protected set]
Tipo de extrada aceptado.
- Type **AcceptedType** [get, protected set]
Tipo aceptado del objeto que devuelven los argumentos.
- string **OperationToken** [get, protected set]
- BinaryOperation **Operation** [get, protected set]

Propiedades heredados de [Hulk.HulkExpression](#)

- bool **IsDependent** [get, protected set]
Especificacion de si la expresion depende de un parametro (argumento de funcion) o no.

5.44.1. Descripción detallada

Clase que representa la operacion desigualdad que. Expresion del tipo [expresion] != [expresion].

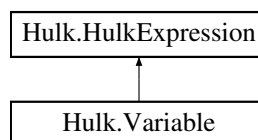
La documentación de esta clase está generada del siguiente archivo:

- C:/Users/Jossue/Cosas/proyectos y cosas/Proyectos de Programacion/H.U.L.K/Hulk/BinaryOperations.cs

5.45. Referencia de la clase Hulk.Variable

Clase que representa a una variable o literal.

Diagrama de herencia de Hulk.Variable



Tipos públicos

- enum [VariableOptions](#) {
Value , InitializedVariable , NonInitialized , FunctionArgument ,
Dependent }
Opciones para establecer características de las variable.

Métodos públicos

- [Variable](#) (object value)
- [Variable](#) (string name, object value)
Constructor para una variable declarada sin tipo.
- [Variable](#) (string name, object value, [Types](#) type, [VariableOptions](#) options)
Constructor para las variables declaradas explícitamente con el tipo.
- override object **GetValue** (bool execute)
- override [Types](#) **CheckType** ()

Métodos públicos heredados de [Hulk.HulkExpression](#)

- object [GetValue](#) (bool execute)
Metodo que devuelve el valor de la expresion.
- [Types](#) **CheckType** ()
Chequea que no existan errores semanticos en la expresion y devuelve el tipo de retorno de la misma.

Propiedades

- string? **Name** [get, protected set]
Nombre de la variable.
- **Types Type** [get, protected set]
Tipo de la variable.
- **VariableOptions Options** [get, set]
Características de la variable.
- object **Value** [get, set]
Valor de la variable.

Propiedades heredados de **Hulk.HulkExpression**

- bool **IsDependent** [get, protected set]
Especificación de si la expresión depende de un parámetro (argumento de función) o no.

5.45.1. Descripción detallada

Clase que representa a una variable o literal.

5.45.2. Documentación de constructores y destructores

5.45.2.1. **Variable()** [1/3]

```
Hulk.Variable.Variable (
    object value )
```

<summary> Constructor para literales \summary>

5.45.2.2. **Variable()** [2/3]

```
Hulk.Variable.Variable (
    string name,
    object value )
```

Constructor para una variable declarada sin tipo.

Parámetros

<i>name</i>	Nombre de la variable
<i>value</i>	Valor de la variable

5.45.2.3. **Variable()** [3/3]

```
Hulk.Variable.Variable (
    string name,
```

```

    object value,
    Types type,
    VariableOptions options )

```

Constructor para las variables declaradas explícitamente con el tipo.

Parámetros

<i>name</i>	Nombre de la variable
<i>value</i>	Valor de la variable
<i>type</i>	Tipo de la variable
<i>options</i>	Opciones para la declaración de la variable

Excepciones

<i>SemanticError</i>	
----------------------	--

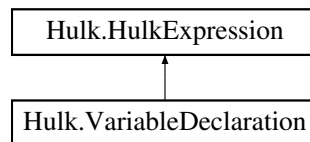
La documentación de esta clase está generada del siguiente archivo:

- C:/Users/Jossue/Cosas/proyectos y cosas/Proyectos de Programacion/H.U.L.K/Hulk/BasicExpressions.cs

5.46. Referencia de la clase Hulk.VariableDeclaration

Representa la declaración de variables. Expresión del tipo [tipo?] [id] = [expresión].

Diagrama de herencia de Hulk.VariableDeclaration



Métodos públicos

- **VariableDeclaration** (List< string > names, string type, **HulkExpression** ValueExp)
Constructor para una declaración de variable con tipo explícito.
- **VariableDeclaration** (List< string > names, **HulkExpression** ValueExp)
Constructor para una declaración de variable con tipo explícito.
- override object **GetValue** (bool execute)
- override **Types CheckType** ()

Métodos públicos heredados de **Hulk.HulkExpression**

- object **GetValue** (bool execute)
Método que devuelve el valor de la expresión.
- **Types CheckType** ()
Chequea que no existan errores semánticos en la expresión y devuelve el tipo de retorno de la misma.

Propiedades

- `List< string > Names` [get]
Nombres de las variables.
- `HulkExpression ValueExpression` [get]
Expresion que define el valor de las variables.
- `object Value` [get, set]
Valor de las variables.
- `Types Type` [get]
Tipo de las variables que se declaran.

Propiedades heredados de `Hulk.HulkExpression`

- `bool IsDependent` [get, protected set]
Especificacion de si la expresion depende de un parametro (argumento de funcion) o no.

5.46.1. Descripción detallada

Representa la declaracion de variables. Expresion del tipo `[tipo?] [id] = [expresion]`.

5.46.2. Documentación de constructores y destructores

5.46.2.1. `VariableDeclaration()` [1/2]

```
Hulk.VariableDeclaration.VariableDeclaration (
    List< string > names,
    string type,
    HulkExpression ValueExp )
```

Constructor para una declaracion de variable con tipo explicito.

Parámetros

<i>names</i>	Nombre de las variables que se van a declarar
<i>type</i>	Tipo de las variables que se van a declarar
<i>ValueExp</i>	Valor que tomaran las variables

Excepciones

<i>SemanticError</i>	
----------------------	--

5.46.2.2. `VariableDeclaration()` [2/2]

```
Hulk.VariableDeclaration.VariableDeclaration (
    List< string > names,
    HulkExpression ValueExp )
```


Constructor para una declaracion de variable con tipo explicito.

Parámetros

<i>names</i>	Nombre de las variables que se van a declarar
<i>ValueExp</i>	Valor que tomaran las variables

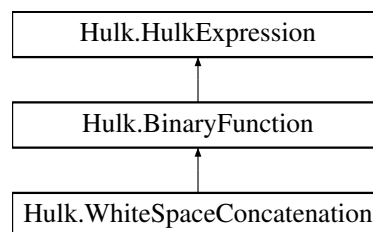
La documentación de esta clase está generada del siguiente archivo:

- `C:/Users/Jossue/Cosas/proyectos y cosas/Proyectos de Programacion/H.U.L.K/Hulk/Declarations.cs`

5.47. Referencia de la clase `Hulk.WhiteSpaceConcatenation`

Clase que representa la concatenacion con espacio en blanco intermedio. Expresion del tipo `[expresion] @ [expresion]`.

Diagrama de herencia de `Hulk.WhiteSpaceConcatenation`



Métodos públicos

- **`WhiteSpaceConcatenation`** ([HulkExpression](#) leftArgument, [HulkExpression](#) rightArgument)

Métodos públicos heredados de [Hulk.BinaryFunction](#)

- [BinaryFunction](#) ([HulkExpression](#) leftArgument, [HulkExpression](#) rightArgument)
Contructor de un objeto que representa a una funcion binaria.
- override object **`GetValue`** (bool execute)
- override [Types](#) **`CheckType`** ()
- object [Evaluate](#) (object left, object right)
Evalua la expresion.
- delegate object **`BinaryOperation`** (object left, object right)

Métodos públicos heredados de [Hulk.HulkExpression](#)

- object [GetValue](#) (bool execute)
Metodo que devuelve el valor de la expresion.
- [Types](#) **`CheckType`** ()
Chequea que no existan errores semanticos en la expresion y devuelve el tipo de retorno de la misma.

Otros miembros heredados

Propiedades heredados de **Hulk.BinaryFunction**

- **HulkExpression LeftArgument** [get, protected set]
Expresion que representa al miembro izquierdo de la expresion.
- **HulkExpression RightArgument** [get, protected set]
Expresion que representa al miembro derecho de la expresion.
- **Types ReturnedType** [get, protected set]
Tipo de retorno aceptado.
- **Types EnteredType** [get, protected set]
Tipo de extrada aceptado.
- **Type AcceptedType** [get, protected set]
Tipo aceptado del objeto que devuelven los argumentos.
- string **OperationToken** [get, protected set]
- BinaryOperation **Operation** [get, protected set]

Propiedades heredados de **Hulk.HulkExpression**

- bool **IsDependent** [get, protected set]
Especificacion de si la expresion depende de un parametro (argumento de funcion) o no.

5.47.1. Descripción detallada

Clase que representa la concatenacion con espacio en blanco intermedio. Expresion del tipo [expresion] @ [expresion].

La documentación de esta clase está generada del siguiente archivo:

- C:/Users/Jossue/Cosas/proyectos y cosas/Proyectos de Programacion/H.U.L.K/Hulk/BinaryOperations.cs