

Class name:

Lambda

Package: langSpec

Documentation:

Lambda symbol that regonises every input.

External Documents:

Export Control: Public

Cardinality: n

Hierarchy:

Superclasses: GrammarSymbol

Public Interface:

Operations:

describe

recognise

State machine: No

Concurrency: Sequential

Persistence: Transient

Operation name:

describe

Public member of: Lambda

Return Class: string

Documentation:

Describes the lambda grammatical symbol.

Concurrency: Sequential

State machine: No

Operation name:

recognise

Public member of: Lambda

Documentation:

The lambda recognition is always true.

Concurrency: Sequential

State machine: No

Class name:

Reify

Package: langSpec

Documentation:

Every language has the Reify statement and this class represent it grammar symbol.

External Documents:
Export Control: Public
Cardinality: n
Hierarchy:
 Superclasses: NonTerminal, GrammarSymbol
Public Interface:
 Operations:
 recognise
 describe

State machine: No
Concurrency: Sequential
Persistence: Transient

Operation name:

recognise

Public member of: Reify
Documentation:
 Finds out if a reify statement is written.

Concurrency: Sequential
State machine: No

Operation name:

describe

Public member of: Reify
Return Class: string
Documentation:
 Describes the grammatical Reify symbol.

Concurrency: Sequential
State machine: No

Class name:

ReifySemanticAction

Package: langSpec
Documentation:
 A reify-statement semantic action to be executed once the rule option has been recognised.

External Documents:
Export Control: Public
Cardinality: n
Hierarchy:
 Superclasses: SemanticAction
Public Interface:

Operations:

execute

State machine:

No

Concurrency:

Sequential

Persistence:

Transient

Operation name:

execute

Public member of:

ReifySemanticAction

Documentation:

Behaviour execution. Takes nitro and application objects.

Concurrency: Sequential

State machine:

No

Class name:

SemanticAction

Package:

langSpec

Documentation:

A semantic action to be executed once the rule option has been recognised.

External Documents:

Export Control:

Public

Cardinality:

n

Hierarchy:

Superclasses:

none

Associations:

<no rolename> : Right in association <unnamed>

Public Interface:

Operations:

execute

Private Interface:

Attributes:

action : string

Action to be executed.

State machine:

No

Concurrency:

Sequential

Persistence:

Transient

Operation name:

execute

Public member of: SemanticAction

Documentation:

Behaviour execution. Takes 'nodes' as the tree nodes set.

Concurrency: Sequential

State machine: No

Class name:

GrammarSymbol

Package: langSpec

Documentation:

By using inheritance, this is the generic representation of every grammar symbol. In Python this class is useless because it has no static type checking.

External Documents:

Export Control: Public

Cardinality: n

Hierarchy:

Superclasses: none

Associations:

<no rolename> : Right in association <unnamed>

<no rolename> : TreeNode in association <unnamed>

State machine: No

Concurrency: Sequential

Persistence: Transient

Class name:

Right

Package: langSpec

Documentation:

An option of a free-context grammar rule right-part.

External Documents:

Export Control: Public

Cardinality: n

Hierarchy:

Superclasses: none

Associations:

<no rolename> : NonTerminal in association <unnamed>

<no rolename> : NonTerminal in association <unnamed>

actions : SemanticAction in association <unnamed>

nodes : GrammarSymbol in association <unnamed>

<no rolename> : TreeNode in association <unnamed>

Public Interface:

Operations:
recognise

Private Interface:

Attributes:
nodes
List of grammatical symbols on the right part of the rule.
actions
List of semantic rules (actions).

State machine: No
Concurrency: Sequential
Persistence: Transient

Operation name:

recognise

Public member of: Right

Documentation:
Finds out whether the right part recognises the input.

Concurrency: Sequential
State machine: No

Class name:

NonTerminal

Package: langSpec

Documentation:
NonTerminal symbols, either in the right or in left part of a free-context grammar rule.

External Documents:

Export Control: Public

Cardinality: n

Hierarchy:

Superclasses: Language, GrammarSymbol

Associations:

<no rolename> : Language in association <unnamed>
<no rolename> : Language in association <unnamed>
<no rolename> : Language in association <unnamed>
<no rolename> : Language in association <unnamed>
options : Right in association <unnamed>
<no rolename> : Right in association <unnamed>

Public Interface:

Operations:
describe
recognise

Private Interface:

Attributes:

name : string

Grammar symbol identification.

description : string

Rule's description. If the NT has no description, it's on a right part of a rule.

options

Right part of the rule.

State machine:

No

Concurrency:

Sequential

Persistence:

Transient

Operation name:

describe

Public member of:

NonTerminal

Return Class:

string

Documentation:

Describes the grammatical element and its right part.

*Concurrency:*Sequential

State machine:

No

Operation name:

recognise

Public member of:

NonTerminal

Documentation:

Finds out whether the grammatical symbol recognises the input. Returns its application tree.

*Concurrency:*Sequential

State machine:

No

Class name:

Terminal

Package:

langSpec

Documentation:

Terminal symbols defined as its character sequence (attribute value).

External Documents:

Export Control:

Public

Cardinality:

n

Hierarchy:

Superclasses: GrammarSymbol

Associations:

<no rolename> : Language in association <unnamed>

<no rolename> : Language in association <unnamed>

<no rolename> : Language in association <unnamed>

Public Interface:

Operations:

describe

recognise

Private Interface:

Attributes:

value : string

Text token to be recognised.

State machine:

No

Concurrency:

Sequential

Persistence:

Transient

Operation name:

describe

Public member of:

Terminal

Return Class:

string

Documentation:

Describes the grammatical terminal symbol.

Concurrency: Sequential

State machine:

No

Operation name:

recognise

Public member of:

Terminal

Documentation:

Finds out whether the grammatical symbol recognises the input.

Concurrency: Sequential

State machine:

No

Class name:

Language

Package:

langSpec

Documentation:

Language Definition Class; defines each application's language.

External Documents:

Export Control: Public

Cardinality: n

Hierarchy:

Superclasses: none

Associations:

<no rolename> : ObjectGrammarCreator in association <unnamed>

lexicalSpec : NonTerminal in association <unnamed>

<no rolename> : NonTerminal in association <unnamed>

<no rolename> : NonTerminal in association <unnamed>

syntacticSpec : NonTerminal in association <unnamed>

notSkipSpec : Terminal in association <unnamed>

skipSpec : Terminal in association <unnamed>

skipSpec : Terminal in association <unnamed>

<no rolename> : AppWindow in association <unnamed>

Public Interface:

Operations:

describe

Private Interface:

Attributes:

name : string

Language's unique identification.

start : GrammarSymbol

Language non-terminal start symbol.

lexicalSpec

Dictionary of non-terminal objects that recognises tokens.

syntacticSpec

Dictionary of syntactic non-terminal objects.

skipSpec

List of tokens to skip (not recognise) automatically.

notSkipSpec

List of tokens not to skip automatically.

State machine:

No

Concurrency:

Sequential

Space complexity:

pepe

Persistence:

Transient

Operation name:

describe

Public member of:

Language

Return Class: string
Documentation:
Gives us information related to the Language at runtime.

Concurrency: Sequential
State machine: No

Association:

Derived: No
Direction: <non-directional>
Association Class: none

Role: actions
Class: SemanticAction
Cardinality / Multiplicity: n
Navigable: Yes
Aggregate: No
Static: No
Friend: No
Access: Implementation
Containment: Unspecified

Role:
Class: Right
Cardinality / Multiplicity:
Navigable: No
Aggregate: Yes
Static: No
Friend: No
Access: Public
Containment: Unspecified

Association:

Derived: No
Direction: <non-directional>
Association Class: none

Role: nodes
Class: GrammarSymbol
Cardinality / Multiplicity: n
Navigable: Yes
Aggregate: No
Static: No
Friend: No
Access: Implementation
Containment: Unspecified

Role:

Class: Right
Cardinality / Multiplicity:
Navigable: No
Aggregate: No
Static: No
Friend: No
Access: Public
Containment: Unspecified

Association:

Derived: No
Direction: <non-directional>
Association Class: none

Role: options
Class: Right
Cardinality / Multiplicity: n
Navigable: Yes
Aggregate: No
Static: No
Friend: No
Access: Implementation
Containment: Unspecified

Role:
Class: NonTerminal
Cardinality / Multiplicity:
Navigable: No
Aggregate: Yes
Static: No
Friend: No
Access: Public
Containment: Unspecified

Association:

Derived: No
Direction: <non-directional>
Association Class: none

Role: lexicalSpec
Class: NonTerminal
Cardinality / Multiplicity: n
Navigable: Yes
Aggregate: No
Static: No
Friend: No
Access: Implementation

Containment: Unspecified

Role:

Class: Language

Cardinality / Multiplicity:

Navigable: No

Aggregate: No

Static: No

Friend: No

Access: Public

Containment: Unspecified

Keys / Qualifiers:

ruleName : string

Association:

Derived: No

Direction: <non-directional>

Association Class: none

Role: syntacticSpec

Class: NonTerminal

Cardinality / Multiplicity: n

Navigable: Yes

Aggregate: No

Static: No

Friend: No

Access: Implementation

Containment: Unspecified

Role:

Class: Language

Cardinality / Multiplicity:

Navigable: No

Aggregate: No

Static: No

Friend: No

Access: Public

Containment: Unspecified

Keys / Qualifiers:

ruleName : string

Association:

Derived: No

Direction: <non-directional>

Association Class: none

Role: notSkipSpec

Class: Terminal
Cardinality / Multiplicity: 0..n
Navigable: Yes
Aggregate: No
Static: No
Friend: No
Access: Implementation
Containment: Unspecified

Role:
Class: Language
Cardinality / Multiplicity:
Navigable: No
Aggregate: No
Static: No
Friend: No
Access: Public
Containment: Unspecified

Association:

Derived: No
Direction: <non-directional>
Association Class: none

Role: skipSpec
Class: Terminal
Cardinality / Multiplicity: 0..n
Navigable: Yes
Aggregate: No
Static: No
Friend: No
Access: Implementation
Containment: Unspecified

Role:
Class: Language
Cardinality / Multiplicity:
Navigable: No
Aggregate: No
Static: No
Friend: No
Access: Public
Containment: Unspecified

Generalization:

Access: Public
Virtual: No

Friend: No
Client:
ReifySemanticAction

Supplier:
SemanticAction

Generalization:

Access: Public
Virtual: No
Friend: No
Client:
Lambda

Supplier:
GrammarSymbol

Generalization:

Access: Public
Virtual: No
Friend: No
Client:
Reify

Supplier:
GrammarSymbol

Generalization:

Access: Public
Virtual: No
Friend: No
Client:
NonTerminal

Supplier:
GrammarSymbol

Generalization:

Access: Public
Virtual: No
Friend: No
Client:
Terminal

Supplier:
GrammarSymbol