

Class: <entity>

Field: `type`

Type: (:union <target-object> <boolean>)

Description: Data type of the expression

Field: `type-dispatched?`

Type: <boolean>

Description: #t if the data type of the expression has been dispatched

Field: `exact-type?`

Type: <boolean>

Description: #t if the data type of the expression is exact

Field: `address`

Type: (:maybe <address>)

Description: The address with which the entity can be accessed.

Class: <expression>

Superclass: <entity>

Field: `pure?`

Type: <boolean>

Description: #t if the expression is pure, i.e. it is guaranteed to have no side effects

Field: `static?`

Type: <boolean>

Description: This field is not used.

Field: `need-revision?`

Type: <boolean>

Description: Set #t in order to force revision of the expression during parametrized object instantiation.

Field: `to-value`

Type: (:maybe <target-object>)

Description: The value of the expression. Set this field to `null` if the value is not known.

Class: <target-object>

Superclass: <entity>

Field: `primitive?`

Type: <boolean>

Description: **#t** iff the class of the object is a primitive class and the other fields are set accordingly

Field: **incomplete?**

Type: **<boolean>**

Description: **#t** iff the object is incomplete, i.e. its fields are not known

Field: **al-field-values**

Type: **(:union (:uniform-list (:pair <symbol> <object>))) <boolean>**

Description: List of field names and values. Value **#f** is used to represent an object that is not known compile time.

Field: **x-prim-contents**

Type: **<object>**

Description: The contents of a primitive object. This field is **null** for nonprimitive objects.

Field: **x-opt-contents**

Type: **<object>**

Description: This field is used to optimize list constants.

Target class: **<object>**

inheritable

immutable

equal by reference

No own fields

Target class: **<class>**

Superclass: **<object>**

inheritable

immutable

equal by reference

Field: **cl-superclass**

Type: **(:union <class> <null>)**

Description: The superclass of the class.

Field: **l-fields**

Type: **(:uniform-list <field>)**

Description: New fields defined in this class in addition to the superclass fields.

Field: **l-all-fields**

Type: **(:uniform-list <field>)**

Description: All fields of the class.

Field: `inheritable?`
Type: `<boolean>`
Description: `#t` iff the class is inheritable

Field: `immutable?`
Type: `<boolean>`
Description: `#t` iff the class is immutable

Field: `eq-by-value?`
Type: `<boolean>`
Description: `#t` iff the class is equal by value

Field: `s-ctr-access`
Type: `<symbol>`
Description: Access to the constructor.

Field: `type-constructor`
Type: `(:maybe <type>)`
Description: The constructor type of the class. This field is computed from the fields defined by the class.

Field: `proc-constructor`
Type: `(:maybe <procedure>)`
Description: The constructor of the class.

Field: `goops?`
Type: `<boolean>`
Description: true iff the class is a GOOPS class

Field: `has-zero?`
Type: `<boolean>`
Description: true iff the class defines a zero value

Field: `zero-prim?`
Type: `<boolean>`
Description: true iff the class defines a zero value and it is a primitive value

Field: `x-zero-value`
Type: `<object>`
Description: A procedure computing the zero value, a primitive zero value, or null. This field is to be used internally by Theme-D.

Field: `module`
Type: `(:union (:uniform-list <symbol>) <symbol>)`
Description: The module where the class is defined. Value `'builtins` means a built-in class.

Field: `str-name`
Type: `<string>`

Description: the name of the class as a string

Target class: `<param-class>`

Superclass: `<class>`

inheritable

mutable

equal by reference

Field: `i-params`

Type: `<integer>`

Description: number of type parameters for this parametrized class

Field: `l-tvars`

Type: `(<uniform-list <type-variable>)`

Description: type variables parametrizing this class

Field: `cl-instance-superclass`

Type: `<class>`

Description: the superclass of the instances of this parametrized class

Field: `l-instance-fields`

Type: `(<uniform-list <field>)`

Description: the fields defined by the instances of this parametrized class

Field: `l-instance-all-fields`

Type: `(<uniform-list <field>)`

Description: all the fields defined by the instances of this parametrized class and its superclasses

Field: `instances-inheritable?`

Type: `<boolean>`

Description: `#t` iff the instances of this parametrized class are inheritable

Field: `instances-immutable?`

Type: `<boolean>`

Description: `#t` iff the instances of this parametrized class are immutable

Field: `instances-eq-by-value?`

Type: `<boolean>`

Description: `#t` iff the instances of this parametrized class are equal by value

Field: `instance-has-constructor?`

Type: `<boolean>`

Description: `#t` iff the instances of this parametrized class have a constructor

Field: `s-instance-ctr-access`

Type: `<symbol>`

Description: Access to the constructors of the instances of this parametrized class.

Field: `instance-has-zero?`

Type: `<boolean>`

Description: True iff the instances of this parametrized class define a zero value.

Field: `proc-instance-zero`

Type: `(:maybe <procedure>)`

Description: A procedure to compute the zero values. This field is to be used internally by Theme-D.

Target class: `<param-logical-type>`

Superclass: `<class>`

inheritable

mutable

equal by reference

Field: `i-params`

Type: `<integer>`

Description: number of type parameters for this parametrized class

Field: `l-tvars`

Type: `(:uniform-list <type-variable>)`

Description: type variables parametrizing this class

Field: `x-value-expr`

Type: `(:union <type> <type-variable> <singleton>)`

Description: the expression of this parametrized logical type

Target class: `<param-procedure>`

Superclass: `<class>`

inheritable

mutable

equal by reference

Field: `i-first-number`

Type: `<integer>`

Description: the first location used by the type variables

Field: `i-nr-of-tvars`

Type: `<integer>`

Description: number of type variables in this class

Field: `l-tvars`

Type: (:uniform-list <type-variable>)

Description: type variables parametrizing this class

Field: type-contents

Type: (:union <type> <type-variable> <singleton>)

Description: the type of the instances of this parametrized procedure