Languages-beta: MiniJava-Funcons-Index *

The PLanCompS Project

`MiniJava-Funcons-Index.cbs` | PLAIN | PRETTY

OUTLINE

---

*Suggestions for improvement: `plancomps@gmail.com`.
Reports of issues: `https://github.com/plancomps/CBS-beta/issues`.

Computations

Normal computation

Flowing

[*Funcon* sequential
Alias seq
Funcon if-true-else
Alias if-else
Funcon while-true
Alias while]

Giving

[*Funcon* initialise-giving
Funcon give
Funcon given
Funcon interleave-repeat]

Binding

[*Type* environments
Alias envs
Datatype identifiers
Alias ids
Funcon initialise-binding
Funcon bound-value
Alias bound
Funcon scope
Funcon collateral
Funcon recursive]

Generating

[*Funcon* fresh-atom]

Storing

[*Funcon* initialise-storing
Datatype variables
Alias vars
Funcon allocate-initialised-variable
Alias alloc-init
Funcon assign
Funcon assigned]

Interacting

Output

[*Funcon* print]

Abnormal computation

Failing

[*Funcon* finalise-failing
Funcon checked]

Values

Value Types

[*Type* value-types
Alias types]

Primitive values

Booleans

[*Datatype* booleans
Alias bools
Funcon true
Funcon false
Funcon not]

Integers

[*Type* integers
Alias ints
Funcon integer-add
Alias int-add
Funcon integer-subtract
Alias int-sub
Funcon integer-multiply
Alias int-mul
Funcon integer-is-less
Alias is-less
Funcon decimal-natural
Alias decimal]

The null value

[*Datatype* null-type
Funcon null-value
Alias null]

Composite values

Sequences of values

[*Funcon* length
Funcon index
Funcon first]

Tuples

[*Datatype* tuples
Funcon tuple-elements]

Strings

[*Funcon* to-string]

Vectors

[*Datatype* vectors
Funcon vector
Funcon vector-elements]

Sets

[*Type* sets
Funcon set-unite]

Maps

[*Funcon* map
Funcon map-lookup
Alias lookup]

References and pointers

[*Datatype* references
Funcon reference
Type pointers
Funcon dereference]

Classes

[*Funcon* class
Funcon class-instantiator
Funcon class-name-single-inheritance-feature-map]

Objects

[*Datatype* objects
Funcon object
Funcon object-class-name
Funcon object-single-inheritance-feature-map]

Abstraction values

Generic abstractions

[*Funcon* abstraction
Funcon closure]

Thunks

[*Funcon* thunk
Funcon force]

Functions

[*Datatype* functions
Funcon function
Funcon apply]

Patterns

[*Datatype* patterns
Funcon pattern
Funcon match]