File created: 23-Feb-2024 12:10:03 **{DSK}<home>larry>il>loops>test>medley>LOOPSTEST.;3**

edit by: lmm

changes to: (FNS MAKE-TEDIT-TEST TESTLOOPS)
(VARS LOOPSTESTCOMS)

previous date: 22-Feb-2024 15:52:27 {DSK}<home>larry>il>loops>system>LOOPSTEST.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

```
(RPAQQ LOOPSTESTCOMS ((FNS MAKE-TEDIT-TEST TESTLOOPS)
                      (VARS TESTAV1 TESTAV2)))

(DEFINEQ

(MAKE-TEDIT-TEST
  [LAMBDA (COMMANDS)                                    ; Edited 23-Feb-2024 12:07 by lmm
    (LET ((STR (OPENTEXTSTREAM)))
         (FOR X IN COMMANDS DO (IF (EQ (CAR (LISTP X))
                                       '*)
                                   THEN ;; Interlisp comment
                                        (TEDIT.INSERT STR [CONCAT ";; " (CL:THIRD X)
                                                                  (CONSTANT (MKSTRING (CHARACTER 13]
                                                  NIL NIL T)
                                   ELSE (TEDIT.INSERT STR ">  " NIL NIL T)
                                        (TEDIT.INSERT.OBJECT (BKSYSOBJ (CONCAT (MKSTRING X)
                                                                               " "))
                                                             STR))
                                (TEDIT.INSERT STR (CONSTANT (MKSTRING (CHARACTER 13)))
                                              NIL NIL T))
         (TEDIT STR])

(TESTLOOPS
  [LAMBDA NIL                                           ; Edited 22-Feb-2024 16:33 by lmm
    (FOR PROMPT IN TESTAV1 DO (if (AND (EQ (CAR PROMPT)
                                           '*)
                                       (EQ (CADR PROMPT)
                                           ';;))
                                  THEN (PRIN1 (CL:THIRD PROMPT)
                                              T)
                                       ;; just print, it's a comment)
                                       (TERPRI T)
                                  ELSE (PRIN2 PROMPT T)
                                       (IF (CL:Y-OR-N-P " Proceed? ")
                                           then (PRIN1 " => " T)
                                                (PRINT (EVAL PROMPT)
                                                       T])

)

(RPAQQ TESTAV1
       ('(CNDIR "loops/system")
        (FILESLOAD LOADLOOPS)
        (LOADLOOPS NIL)
        (SETQ ErrorOnNameConflict T)
        ;;
        ;; Define the classes
        (DefineClass 'Tank)
        (SETQ Tank (_ ($ Tank)
                      SetName
                      'Tank))
        ($ Tank)
        ;;
        (DefineClass 'Pipe)
        (_ ($ Pipe)
           SetName
           'Pipe)
        (PP Pipe)
        ;;
        ;; Add outputPressure as IV to Tank
        (_ ($ Tank)
           AddIV
           'outputPressure)
        (PP Tank)
        ;;
```

```
;; Add inputPressure to Pipe
(_ ($ Pipe)
   AddIV
   'inputPressure)
;;
;; Create subclass of Tank and Pipe named Tank1 and Pipe1
(SETQ Tank1 (_ ($ Tank)
               New
               'Tank1))
(PP ($ Tank1))
(SETQ Pipe1 (_ ($ Pipe)
               New
               'Pipe1))
(PP ($ Pipe1))
;;
;; Create an instance of IndirectVariable")
;;  Initialize its contents to point to the Tank's pressure
(SETQ indVar1 (_ ($ IndirectVariable)
                 New
                 'indVar1))
(_ ($ indVar1)
   SetName
   'indVar1)
;;
;; Assign object and varName
(_@ ($ indVar1)
    object
    ($ Tank1))
(_@ ($ indVar1)
    varName
    'outputPressure)
(PP ($ indVar1))
;;
;; Install the active value instance as the pipe's input pressure
(_ ($ indVar1)
   AddActiveValue
   ($ Pipe1)
   'inputPressure)
(PP ($ indVar1))
;;
;; Accesses to either pipe's input pressure or tank's output pressure
(@ Pipe1 inputPressure)
(_@ Pipe1 inputPressure 100)
(@ Tank1 outputPressure)
(_@ Tank1 outputPressure 200)
(@ Pipe1 inputPressure)
(@ Tank1 outputPressure)
;;
;; Show Inspector Window on Pipe1
(_ Tank1 Inspect NIL)
(_ Pipe1 Inspect NIL)))
```

(RPAQQ **TESTAV2**

```
(;; ** NewTestAV **
;; From Section 8.2, Example 2 of the LRM **
;; Create the Bin class for the Conveyor
(DefineClass 'Bin)
(DefineClass 'Conveyor)
;; Add IVs to describe Bin
(_ ($ Bin)
   AddIV
   'height 0)
(_ ($ Conveyor)
   AddIV
   'height 0)
;; Create a Bin instance.
(SETQ Bin1 (_ ($ Bin)
              New
              'Bin1))
(SETQ Bin1 (_ ($ Bin1)
              SetName
              'Bin1))
```

;; Create a Conveyor instance.

```
(SETQ Conveyor1 (_ ($ Conveyor)
                   New
                   'Conveyor1))
(SETQ COnveyor1 (_ ($ Conveyor1)
                   SetName
                   'Conveyor1))
```

;; Define 3FeetAbove as a class.

```
(DefineClass '3FeetAbove '(IndirectVariable))
(SETQ 3FeetAbove (_ ($ 3FeetAbove)
                    SetName
                    '3FeetAbove))
(PP 3FeetAbove)
```

;; Create an instance of 3FeetAbove.

;; Initialize its contents to point to the bin's height.

```
(_ ($ 3FeetAbove)
   New
   '3fa1)
(_@ ($ 3fa1)
    object
    ($ Bin1))
(_@ ($ 3fa1)
    varName
    'height)
(_ ($ 3fa1)
   Inspect NIL)
```

;; Install 3fa1 as the value of the conveyor's height.

```
(_ ($ 3fa1)
   AddActiveValue
   ($ Conveyor1)
   'height)
(_ ($ 3fa1)
   Inspect NIL)
```

;; The height of Bin1 defaults to 0, but what is the height of conveyor?

```
(@ ($ Bin1)
   height)
(@ ($ Conveyor1)
   height)
```

;; Now, set Bin1's height or Conveyor1's height.

;; See how the track each other.

```
(_@ ($ Bin1)
    height 15)
(@ ($ Conveyor1)
   height)
(_@ ($ Conveyor1)
    height 21)
(@ ($ Bin1)
   height)
```

;; Define subclass of LocalStateActiveValue.

;; Provide two IVs relative to height.

```
(DefineClass 'WarningAV '(LocalStateActiveValue))
(_ ($ WarningAV)
   AddIV
   'lowTrigger 0)
(_ ($ WarningAV)
   AddIV
   'highTrigger 100)))
```

FUNCTION INDEX

VARIABLE INDEX