This file is called SMSEMANTICS.DC and contains a semantic description of
SMALLTALK written in itself.
This version was last changed on June 10, 1973.
Use font SMDELEG.FD

SMALLTALK and its Semantics
by
Alan Kay

W A R N I N G ! ! This is an unchecked version done simply to try it out
for basic taste and compactness.

```
♦To ← class¦♦Do ← ♦¦◇name ◇actions.
               ¦↑ Find (name) in CALLER ← class¦♦Do ← activity


To ♦ ¦◇name ¦ᵒ← ⇒ ¦:exp.
     ¦      ¦     ¦Find (name) in CALLER ← exp.
     ¦      ¦!name


To Find ¦:name ¦ᵒin ⇒ ¦:context
        ¦      ¦♦context ← CALLER
        ¦
        ¦Repeat
        ¦  ¦context♦table name  OR  context♦table♦global♦empty?
        ¦  ¦        ⇒¦ᵒ← ⇒ ¦:exp. context♦table name ← exp. Done
        ¦  ¦♦context ← context♦table♦global
        ¦
        ¦↑ context♦table name


To List ¦ᵒ← ⇒ ¦:first :rest. ↑self
        ¦
        ¦ᵒfirst ⇒ ¦ᵒ← ⇒ ¦:first. ↑self
        ¦         ¦!first
        ¦
        ¦ᵒrest ⇒ ¦ᵒ← ⇒ ¦:rest. ↑self
        ¦        ¦!rest
        ¦
        ¦ᵒlength ⇒ ¦! ¦first=NIL ⇒ ¦0
        ¦          ¦  ¦1+rest♦length
        ¦
        ¦ᵒprint ⇒ ¦!"(" ⌽ first♦print ⌽" "⌽ rest♦print ⌽")"
        ¦
        ¦ᵒlist? ⇒ ¦!self
        ¦
        ¦ᵒeval ⇒ ¦Repeat
        ¦        ¦   ¦first = ")" ⇒ ¦Done
        ¦        ¦   ¦first = "."  ⇒ ¦♦value ← restᵒeval
        ¦        ¦   ¦♦value ← first♦eval
        ¦        ¦!value


To Repeat ¦◇program.
          ¦CODEFOR ¦Repeat ¦ clause ♦eval global message self
```

```
To Again ¦⇑ EMPTY CALLER.CALLER

To Done ¦:value. ⇑ value CALLER.CALLER.CALLER

To If ¦:exp ⇒ ¦◦then :exp ¦◦else ⇒ ¦◊. ↑exp
      ¦                    ¦↑exp
      ¦          ¦error "I can't find a "then""
      ¦
      ¦◦then ◊. ¦◦else ⇒ ¦:exp. ↑exp
      ¦         ¦↑EMPTY

To User ¦Repeat
        ¦   ¦Display Read.eval.print

To ◊¦self.table.name ← message.table.pc.first.
    ¦message.table.pc ← message.table.PC.rest.
    ¦message.table name ← message.table.message.table.PC.first.
    ¦message.table.message.table.PC ← message.table.message.table.PC.rest.
    ¦!name.

To : ¦◊name.
     ¦!message.table.name ←
     ¦        message.table.message.table.PC.first .eval message.message

To ◦ ¦◊token ≠ message.table.PC.first ⇒ ¦!EMPTY
     ¦message.table.PC ← message.table.PC.rest

To ⇒ ¦:clause. ⇑ clause CALLER.CALLER.CALLER

To EMPTY ¦◦⇒ ⇒ ¦◊. ⇑ self CALLER.CALLER
         ¦◦empty? ⇒ ¦! .TRUE
         ¦! self

To Apply ¦ :t :g :c :m.
         ¦ (t ← .global g  .caller c  .message m ).eval

To ⇑ ¦:value :destination.
     ¦Apply value destination destination destination.

To Remember ¦◦← ⇒ ¦Repeat
            ¦      ¦   ¦◦EMPTY ⇒ ¦↑self
            ¦      ¦   ¦self :name ← :value
            ¦
            ¦◦copy ⇒ ¦! CODEFOR "somehow copy the table"
            ¦
            ¦◦eval ⇒ ¦"Do something or other"
            ¦
            ¦:name ¦◦← ⇒ ¦:value.
            ¦      ¦     CODEFOR "associate name and value somehow"
            ¦      ¦     ¦↑value
            ¦      ¦
            ¦      ¦!CODEFOR "Get the value associated with the name"
```

```
To class ¦◦bindings.
         ¦↑ instantiate ¦ Remember ← •class •class
         ¦              ¦              •global global
         ¦              ¦              •caller self          ¦               ¦
           •message message
         ¦              ¦
         ¦              ¦              •PC  bindings
         ¦             •eval

To instantiate ¦:classdef.
               ¦Repeat
               ¦   ¦Pause.
               ¦   ¦¦classdef•copy ← •class classdef
               ¦   ¦¦                •global global
               ¦   ¦¦                •caller caller
               ¦   ¦¦                •message message
               ¦   ¦¦                •PC  classdef•DO
               ¦   ¦•eval


To Word
   ¦◦← ⇒ ¦:first :rest.
   ¦     ¦•rest ← Word ← first•butfirst rest.
   ¦     ¦•first ← first•first.
   ¦     ¦↑ self
   ¦
   ¦◦first ⇒ ¦◦← ⇒ ¦:first•character ⇒ ¦↑first
   ¦         ¦     ¦error"input is not a character"
   ¦         ¦! first
   ¦
   ¦◦rest ⇒ ¦◦← ⇒ ¦:rest•character ⇒ ¦↑rest
   ¦        ¦     ¦Error"Input is not a word"
   ¦        ¦! rest
   ¦
   ¦◦length ⇒ ¦! rest = NULL ⇒ 1
   ¦          ¦ 1 + rest•length
   ¦
   ¦◦print ⇒ ¦↑ first•print. rest•print
   ¦
   ¦◦word? ⇒ ¦! self
   ¦
   ¦◦= ⇒ ¦:value. ↑ (first = value•first) AND next = value•next
   ¦
   ¦◦eval ⇒ ¦•env ← global.
   ¦        ¦Repeat
   ¦        ¦   ¦env•empty? ⇒ ¦↑EMPTY
   ¦        ¦   ¦•temp ← env•table self ⇒ ¦↑ apply temp global caller
   ¦        ¦   ¦                         ¦         message
   ¦        ¦   ¦•env ← global•table•global
```