OO programming with Language E8
OO prog features

- Encapsulation
- Inheritance
- Dynamic Method Binding
- Name binding
- Exceptions
- Typing
Encapsulation

(fun pair ( x y)
  (object (def fst x)
    (def snd y)
    (method asString ( )
      (++ "(" (self.fst.asString) ","
        (self.snd.asString) ")")
    ))
)

An object is described by what requests it responds to (fst,snd)
Inheritance

(val bool
  (object
    (def true
      (object
        inherits True
        (method and (x) x)
        (method or (x) self)
        (method not () (bool.false))
        (method ifTrueFalse (x y) (x.apply))
    ))

  (def false
    (object
      inherits False
      (method and (x) self)
      (method or (x) x)
      (method not () (bool.true))
      (method ifTrueFalse (x y) (y.apply))
    ))
))
Dynamic method binding

(val i3
  (object inherits done
    (def x 5)
    (def y 3)
    (method asString ()
      (++ (super.asString)
        " + "
        (self.x.asString)
        " + "
        (self.y.asString)))))

textual representation: [Enter Exp>
  i3
  done[x 5[+ <prim>,
           - <prim>,
           * <prim>,
           / <prim>,
           asString <prim>,
           = <prim>,
           <= <prim>],
  y 3[+ <prim>,
       - <prim>,
       * <prim>,
       / <prim>,
       asString <prim>,
       = <prim>,
       <= <prim>],
  asString <asString>,
  asString <prim>]
Name Binding vs record field declaration

(fun temp (x y)
  (local (val hidden x)
    (object (def age y)
      (method foo (w)
        (+ hidden (self.age) w))))
)

(val tt (@temp 4 7))
Exceptions

(fun elem (x xs)
   (local (ref ptr xs)
      (block (while (@not (ptr.null))
         (if (ptr.head.= x)
            (return elem 99)
            (return elem 99)
            (:= ptr (ptr.tail))))
   0))))