

```

#lang scheme/base
(require racket/class)
(define 2tree-interface<%> (interface () isEmpty? printTree))
(define Empty2tree%
  (class* object% (2tree-interface<%>)
    (super-new)
    (define/public (isEmpty?) #t)
    (define/public (printTree) '())
  )
)

(define Nonempty2tree%
  (class* object% (2tree-interface<%>)
    (super-new)
    (init-field tag data)
    (field (left null) (right null))
    (define/public (isEmpty?) #f)
    (define/public (get-tag) tag)
    (define/public (get-data) data)
    (define/public (set-tag! tg) (set! tag tg))
    (define/public (set-data! dt) (set! data dt))
    (define/public (get-left) left)
    (define/public (get-right) right)
    (define/public (set-left! lft) (set! left lft))
    (define/public (set-right! rgt) (set! right rgt))
    (define/public (printTree)
      (
        begin
          (cond ((not (null? left))
                 (send left printTree)))
          (print tag)
          (newline)
          (cond ((not (null? right))
                 (send right printTree)))
        )
      )
    )
  )
)

```