

```

#| Team Steele
   Software Engineering I
   Txmminidom

   Software that creates a document object model from XML input.
|#

(require "../interfaces/iBoard.lisp")
(require "../modules/mBoard.lisp")
(require "../modules/Mbasiclex.lisp")

(module tBoard
  (import iBoard)

  (include-book "testing" :dir :teachpacks)
  ;=====SANITY CHECKS=====;
  ; serializedresults (xmlnodes prefix postfix)
  (check-expect (serializedresults nil *tablehead* *tabletail*)
    (string-append *tablehead* *tabletail*))

  (check-expect
    (serializedresults
      (list
        (mv
          "Result"
          (list
            (mv "SectionLabel" "A")
            (mv "PairID-NS" "3")
            (mv "PairID-EW" "5")
          )
          (list
            (mv "TotalScore"
              (list (mv "Direction" "N-S"))
              (list (mv 'text nil "5.0")))
            (mv "MatchpointsNS" nil (list (mv 'text nil "120.0")))
            (mv "MatchpointsEW" nil (list (mv 'text nil "120.0")))
          )
        )
      )
    *tablehead*
    *tabletail*)
    (stringlist-append
      (list
        *tablehead*
        "<tr>"
        "<td>A3</td>"
        "<td>A5</td>"
        "<td>5.0</td><td>&nbsp;</td>"
        "<td>120.0</td>"
        "<td>120.0</td>"
        "</tr>"
        *tabletail*)))

  (check-expect
    (serializedresults
      (list
        (mv
          "Result"
          (list
            (mv "SectionLabel" "A")
            (mv "PairID-NS" "3")
            (mv "PairID-EW" "5")
          )
          (list
            (mv "TotalScore"
              (list (mv "Direction" "E-W"))
              (list (mv 'text nil "7.0")))
          )
        )
      )
    )

```

```

    )
    (mv "MatchpointsNS" nil (list (mv 'text nil "120.0")))
    (mv "MatchpointsEW" nil (list (mv 'text nil "120.0")))
  )
))
*tablehead*
*tabletail*)
(stringlist-append
(list
  *tablehead*
  "<tr>"
  "<td>A3</td>"
  "<td>A5</td>"
  "<td>&nbsp;</td><td>7.0</td>"
  "<td>120.0</td>"
  "<td>120.0</td>"
  "</tr>"
  *tabletail*)))

; serializedboards (xmlnode)
(check-expect (serializedboards nil) "")
(check-expect
(serializedboards
(list
  (mv
    "Board"
    nil
    (list
      (mv "Vulnerable" nil (list (mv 'text nil "NS")))
      (mv "Dealer" nil (list (mv 'text nil "N")))
      (mv "BoardNo" nil (list (mv 'text nil "33")))
      (mv "Deal" nil
        (list
          (mv
            "Hand"
            (list (mv "direction" "N"))
            (list
              (mv "Suit"
                (list (mv "symbol" "S"))
                (list (mv 'text nil "234")))))
          ))
        )))))
))
)
(stringlist-append (list
  "<div class=\"board\">"
  "<div class=\"boardnum\">Board: 33</div>\n"
  "<div class=\"N\">"
  "<div class=\"dealer\">Dealer</div>\n"
  "<div class=\"vulnerable\">Vulnerable</div>\n"
  "&spades;234<br />\n"
  "</div>\n"
  "</div>\n"
  *tablehead*
  *tabletail*)))

(check-expect
(serializedboards
(list
  (mv
    "Board"
    nil
    (list
      (mv "Vulnerable" nil (list (mv 'text nil "NS")))
      (mv "Dealer" nil (list (mv 'text nil "N")))
      (mv "BoardNo" nil (list (mv 'text nil "33")))
      (mv "Deal" nil

```

```

    (list
      (mv
        "Hand"
        (list (mv "direction" "N")))
      (list
        (mv "Suit"
          (list (mv "symbol" "S"))
          (list (mv 'text nil "234"))))
      ))
    ))
  (mv
    "Result"
    (list
      (mv "SectionLabel" "A")
      (mv "PairID-NS" "3")
      (mv "PairID-EW" "5")
      )
      (list
        (mv "TotalScore"
          (list (mv "Direction" "N-S"))
          (list (mv 'text nil "5.0")))
        )
      (mv "MatchpointsNS" nil (list (mv 'text nil "120.0")))
      (mv "MatchpointsEW" nil (list (mv 'text nil "120.0")))
      )
    )
  )))
)
(stringlist-append (list
  "<div class=\"board\">"
  "<div class=\"boardnum\">Board: 33</div>\n"
  "<div class=\"N\">"
  "<div class=\"dealer\">Dealer</div>\n"
  "<div class=\"vulnerable\">Vulnerable</div>\n"
  "&spades;234<br />\n"
  "</div>\n"
  "</div>\n"
  *tablehead*
  "<tr>"
  "<td>A3</td>"
  "<td>A5</td>"
  "<td>5.0</td><td>&nbsp;</td>"
  "<td>120.0</td>"
  "<td>120.0</td>"
  "</tr>"
  *tabletail*))
;=====

;=====PREDICATE TESTS=====
(include-book "doublecheck" :dir :teachpacks)

(defproperty serializedresults-nil=string-append-prefix-postfix-tst
  :repeat 100
  (prefix :value (random-string)
    postfix :value (random-string))
  (implies (and (stringp prefix) (stringp postfix))
    (string-equal (serializedresults nil prefix postfix)
      (string-append prefix postfix)))
  )

(defproperty serializedresults-results-proportional-to-input-tst
  :repeat 100
  (n :value (random-between 1 200)
    nodes :value (random-list-of
      (mv
        "Result"

```

```

      (list
        (mv "SectionLabel" "A")
        (mv "PairID-NS" "3")
        (mv "PairID-EW" "5")
      )
      (list
        (mv "TotalScore"
          (list (mv "Direction" "N-S"))
          (list (mv 'text nil "5.0")))
        )
        (mv "MatchpointsNS" nil (list (mv 'text nil "120.0")))
        (mv "MatchpointsEW" nil (list (mv 'text nil "120.0")))
      )
    )
    :size n))
  (implies (> n 0)
    (< (length (serializedresults (cdr nodes) "" ""))
      (length (serializedresults nodes "" ""))))
  )
;=====;

)

(link Test
  (Mbasiclex mBoard tBoard))

(involve Test)

```