```
;; The first four lines of this file were added by Dracula.
;; They tell DrScheme that this is a Dracula Modular ACL2 program.
;; Leave these lines unchanged so that DrScheme can properly load this file.
#reader(planet "reader.ss" ("cce" "dracula.plt") "modular" "lang")
#| Team Steele
   Software Engineering I
   Mpsc

   Personal Score Card Module
|#
(require "../interfaces/Ipsc.lisp")
(require "../interfaces/Iboard.lisp")
(require "../interfaces/Irankings.lisp")
(require "../interfaces/Ixmlminidom.lisp")
(require "../interfaces/Ibasiclex.lisp")

(module Mpsc
  (import Iboard)
  (import Ixmlminidom)
  (import Irankings)
  (import Ibasiclex)
  (include-book "io-utilities" :dir :teachpacks)
  (include-book "list-utilities" :dir :teachpacks)

  ;Pulls the Name Strings for a given Pair ID
  ;PairID format: (String Direction, String SectionNumber)
  ;Data format: Nodes format
  ;Output format: (String String), Names of the two players
  (defun getNameForID (pairid data) nil)

  ;;;
  ;;;
  (defun getBoardForPair (rbrds sections sectionlabel dir)
    (if (null rbrds)
        ""
        (let* ((sbrd (car rbrds))
               (rest (cdr rbrds))
               (id (second sbrd))
               (contestants (getcontestants sectionlabel dir id sections))
               (players (getcontestantsnames contestants)))
          (concatenate 'string
                       "<tr>"
                       "<td><a href=\"boards-trav.htm#" (first sbrd) "\">"
                            (first sbrd)    "</a></td>"    ; boardnum
                       "<td><a href=\"psc.htm#"
                       dir id sectionlabel "\">"
                       sectionlabel id "</a></td>"    ; vs. info
                       "<td>" players         "</td>"    ; names
                       "<td>" (third sbrd)    "</td>"    ; score
                       "<td>" (fourth sbrd)   "</td>"    ; matchpoints
                       "</tr>"
                       (getBoardForPair rest sections sectionlabel dir)))))

  ;Pulls the match results for a given Pair ID
  ;PairID format: (String Direction, String SectionNumber)
  ;Results format: ?
  ;Output format: String, HTML formatted text comprising all the boards
  ;    for one player pair
  (defun getBoardsForPair (pairid sectionlabel results sections dir)
    (let* ((bforp (assoc-equal (mv pairid sectionlabel) results)))
      (getBoardForPair (cdr bforp) sections sectionlabel dir)))

  ;;;
  ;;;
  (defun getAllPairs (results sections gamestring average top gamenode)
    (let* ((ns (car results))
           (ew (cadr results))
```

```
         (nextns (car ns))
         (keyns (car nextns))
         (nextew (car ew))
         (keyew (car nextew))
         (restns (cdr ns))
         (restew (cdr ew)))
    (if (null nextew)
        ""
        (if (null nextns)
            (let* ((pairid (car keyew))
                   (sectionlabel (cadr keyew)))
              (concatenate 'string
                           *psctableheadpre*
                           "E-W" (car keyew) (cadr keyew)
                           *psctableheadpost*
                           (pscheader sectionlabel "E-W" pairid sections
                                      average top)
                           *psctableheadcontents*
                           ;get info from gamenode
                           (getBoardsForPair pairid
                                             sectionlabel
                                             ew           ; results
                                             sections
                                             "N-S") ; The *opponents'* dir.
                           (pscfooter gamestring)
                           *psctabletail*
                           (getAllPairs (mv ns restew)
                                        sections
                                        gamestring
                                        average
                                        top
                                        gamenode)))
            (let* ((pairid (car keyns))
                   (sectionlabel (cadr keyns)))
              (concatenate 'string
                           *psctableheadpre*
                           "N-S" (car keyns) (cadr keyns)
                           *psctableheadpost*
                           (pscheader sectionlabel "N-S" pairid sections
                                      average top)
                           *psctableheadcontents*
                           (getBoardsForPair pairid
                                             sectionlabel
                                             ns           ; results
                                             sections
                                             "E-W")
                           (pscfooter gamestring)
                           *psctabletail*
                           (getAllPairs (mv restns ew)
                                        sections
                                        gamestring
                                        average
                                        top
                                        gamenode)))))))

(defun pscheader (sectionlabel dir id sections average top)
  (let* ((contestants (getcontestants sectionlabel dir id sections)))
    (concatenate 'string
                 "<tr><td colspan=\"5\">Scorecard for <b>Pair "
                 sectionlabel
                 id
                 " "
                 dir
                 " (strat "
                 (xml-getattribute contestants "Strat")
                 ")</b><br>"
                    (getcontestantsnames contestants)
                 "<br>"
                 " Score: "
                    (xml-gettext (xml-getnode contestants "MatchpointTotal"))
                    " ("
                    (xml-gettext (xml-getnode contestants "Percentage"))
                    "%) "
                 (let* ((award (xml-getnode contestants "Award")))
                   (if award
                       (concatenate 'string
                                    "MP: "
                                    (xml-getattribute award "TypeOfAward")
                                    " "
                                    (xml-gettext award)
                                    " ("
                                    (xml-getattribute award "AwardCategory")
                                    (xml-getattribute award "AwardStrat")
                                    ")")
                       ""))
                 " Ave: "
                 average
                 " Top: "
                 top
                 "<br>"
                 (getrankstring "Section" contestants)
                 " "
                 (getrankstring "Overall" contestants)
                 "</td></tr>")))

(defun pscfooter (gamestring)
  (concatenate 'string "<tr><td colspan=\"5\">" gamestring "</td></tr>"))

;Pulls the Personal Score Card data for all players, and put's them all
;into html table format
;XMLnodes format: Nodes format
;Output format: String, HTML formatted text comprising the score card
;    for one player pair
(defun serializedPSC (gamenode boardnodes)
  (let* ((sections (xml-bfsfindnodes (list gamenode) "Section"))
         (results (getAllSeparateResults boardnodes))
         (gamestring (getgamestring gamenode))
         (movement (xml-getnode gamenode "Movement"))
         (average (xml-gettext (xml-getnode movement "Average")))
         (top (xml-gettext (xml-getnode movement "Top"))))
    (getAllPairs results sections gamestring average top gamenode)))

(export Ipsc))
```