Tom Roberts

LISP MP3 Database Report

Approach

The main approach I took for each requirement was to first make predicate functions that would prevent bad inputs from the user being given to the recursive functions. This increased the cohesion of the code as the recursion only had to carry out its task instead of having to handle bad inputs. In my predicate functions I used asserts to give more meaningful error messages to the user and allow continuable errors.

Problems Faced

The main difficulty I faced was figuring out the recursive functions as most of the other code in the program was simple input checking and error handling. In particular, I was unfamiliar with the recursive functions that had to incrementally build a list to return when the recursive stack completed.

Solutions

To create the predicate functions I referenced the lecture slides and online reference documentation to find built-in functions such as listp, integer etc.

I also used the lecture slides to create some of my functions. I used the 'remove nth element of a list' function for my removeSong function. I also modified this into an 'update nth element of a list' for the updateRating function. This was also very useful in the extension functions I attempted as similar code could be used to gradually build up a list of all an artist's songs, all songs between 2 dates, and all songs with the maximum rating.