Christian Hansen

OoL 3

TLDDR

For my section of this project, I was in charge of implementing comparable and comparator, as well as writing the Javadoc for each file.

Starting with comparable, I updated the compareTo methods in Event, Venue, and Athlete in order to sort the arrays of Events, Venues, and Athletes that were contained in the Driver. I added the lists of these objects into temporary lists so the main lists would maintain their order. I wasn’t 100% sure if the order mattered here, so I played it safe just in case the order did matter. After this, I updated the compareTo method in Fan to compare the Fans by their happiness ratings. After the events had been run, I again created temporary lists for fans and sorted them to print the 10 happiest, then reversed the sort to print the 10 unhappiest fans.

When implementing comparators for athlete, I made one for gold medals, silver medals, bronze medals, and total medals. I then used the tiebreakers in the order from the spec as else statements, continuing until a comparison was found. Using else statements was what made the most sense to me, because if the current comparison would result in a tie (if the if and else if statements failed), it would move to the next step in the comparison (the else was left, allowing for another form of comparison).

I think everything in my part of this was fairly straightforward. I know there is probably another way to implement the tiebreakers in the comparators, but I did what made the most sense to me.