Programming in Java
Preparatory Term 2019
Lab 3

Tournaments are a common way of determining relative strengths of teams in various sports. All tournaments have some things in common: they have a set of competing teams, and they can all determine a “winner”. The exact way a tournament is organized can be different. For example::

- knockout - where a team is eliminated in each match
- league: where teams are assigned to one or more groups, and teams in a group play each other one or more times, accumulating points as they go.
- combination of the above, where the first part is a league format, and the winners of the leagues play in a knockout format
- etc

Each specific tournament, e.g. Wimbledon, IPL, may further fine-tune these mechanisms.

The participants in Tournaments are Teams (here we can assume that a team can comprise one or more players). The structure and composition of teams are sport specific. Every team has a name and a numerical ranking, and smaller the number, the better the team.

The general mechanisms for tournaments is quite independent of the sport being played - cricket, tennis, football, etc. Tournaments run Matches in a particular way, where each Match is a contest between two identified teams. Thus, the same mechanism for running a tournament can be used to handle a cricket world cup or a football league or a tennis grand slam. What changes are the way teams are organized, how matches are played, and how winners are determined.

Any match can have many kinds of outcomes: a clear winner, a draw or abandoned. The tournament rules will determine which of these outcomes are acceptable for a given format/match. Thus we need a way to simulate a match, and return a result that fits the requirements of the match. As an example, a league match could end in a win or draw result, but a knockout would require a clear winner.

Since this is a simulation, you can assume there is a method/module that will “guess” the winner for any pair of teams. This module may use the ranking of the teams or other data, including random numbers, to determine the winner. Note that a draw is also a legitimate outcome at this level.

How can you model this as a Java program? Broadly, the “main” would set up the elements of a tournament:

- Create teams of the appropriate sport
- Instantiate a Tournament of the appropriate format
- Assign the set of participating teams to the tournament

- Ask the tournament to run and produce a result.

The design should allow the following flexibility:

- Ability to change the way matches are organized for a particular tournament, without impacting other classes
- Ability to choose the way a winner is determined. This should be set up at the initial stage (i.e. in main)

Once the simulation is run, we should be able to query the tournament for details of the matches:

- The format and sport being run
- The winner
- The rank of a team in this tournament. May not be well-defined for certain formats
- The result of a particular match (specifying the pair of teams). This should not only return the result (win/draw), but also the details of the scores in that match. Since this is mainly for printing the results, this could be returned as a String or a more convenient form such as JSON. Obviously, the contents would change depending on the sport and the format of the tournament.

Design the classes, their public interfaces, and any helper classes needed. Implement a main that simulates two different kinds of tournaments - different sports and different formats.