Tennis Scoreboard

1. App Specifications:

Tennis has a rather quirky scoring system, and to newcomers it can be a little difficult to keep track of. The tennis society has contracted you to build a scoreboard to display the current score during tennis games.

“TennisGame” class contains the logic which outputs the correct score as a string for display on the scoreboard. When a player scores a point, it triggers a method to be called on your class letting you know who scored the point. Later, you will get a call “score()” from the scoreboard asking what it should display. This method should return a string with the current score.

You can read more about Tennis scores here which is summarized below:

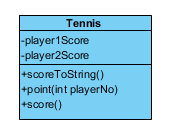
A game is won by the first player to have won at least four points in total and at least two points more than the opponent.

The running score of each game is described in a manner peculiar to tennis: scores from zero to three points are described as "Love", "Fifteen", "Thirty", and "Forty" respectively.

If at least three points have been scored by each player, and the scores are equal, the score is "Deuce".

If at least three points have been scored by each side and a player has one more point than his opponent, the score of the game is "Advantage" for the player in the lead.

The app reports the score for the current game. Sets and Matches are out of scope.



1. Creational patterns:
   1. Singleton

Given the fact that the app reports the score for the current game and looking at the app implementation, we can see that it is a singleton implementation.

* 1. Builder

Each tennis match is made up of two to three sets. To win a set, you must win at least six games. To determine who serves first, you flip a coin or (more likely) spin a racket. Whoever wins the toss gets to decide one of four things: that she wants to serve first, that she wants to receive first, which side of the court she wants to start on (in which case, the opponent chooses who serves first), or that she wants to leave the choices up to her opponent.

Whoever starts serving continues to serve until that game is over. Then the serve moves to the other player.

You serve from behind the baseline, starting on the right-hand side of the court, anywhere between the singles sideline and the center mark on the court.

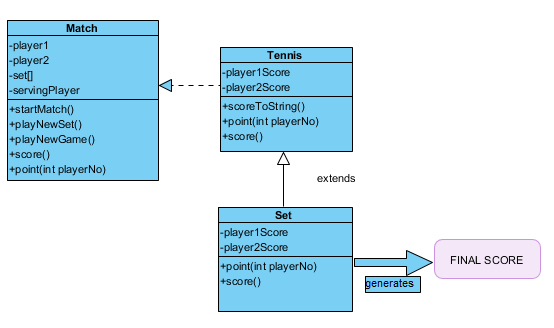
Before the first serve in each new game, whoever is serving announces the score in sets. Say your score first, then your opponent's. So, if you won the first set, you would say "1-0."

To complete a set, someone must win six games; the first person to win six games wins the set.

However, as with "deuce," you must win a set by at least two games. So, if the score is 6-5, the person with 5 must win by two games. If the score ties at 6-6, you play a tiebreaker.

The whole shebang is called a match. The match is determined by the best two out of three sets. So, if you win two sets, you win. If you each win a set, then you play a third set to determine the winner.

Given this use case, we can use a builder in order to keep track of a match and get the return a score.

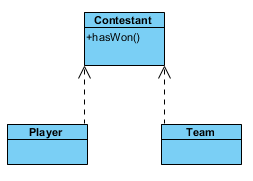


* 1. Factory

The client wants to keep track of both single and double matches.

Same rules apply for the scoring mechanism.

We can use a factory design pattern in order to satisfy the order.



Abstract class Contestant defines some common methods which are applicable to both Player and Team.

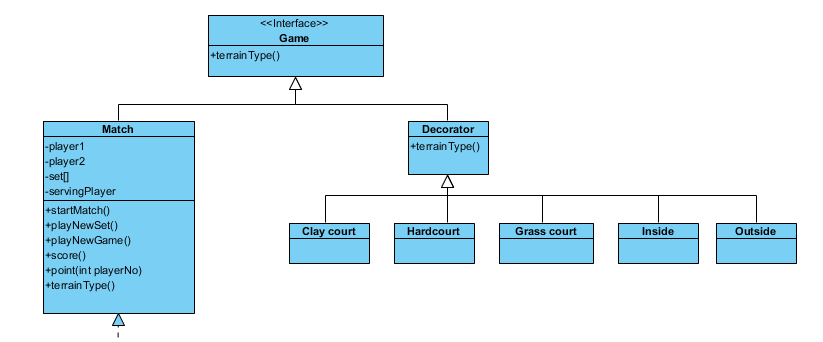
Player class extends Contestant.

Team class extends Contestant.

3. Structural patterns:

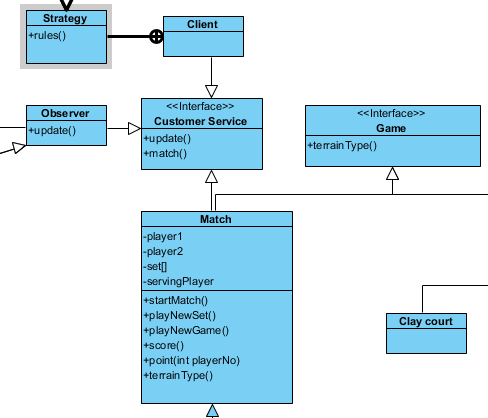
3.1. Decorator

The client made an investment and he acquired multiple court yard types.

He would like to keep track of the court yard where the matches take place. 

3.2. Façade

The client thinks that the application provided is way too complex to be handled. He would like a way to manage all the features that he was provided.

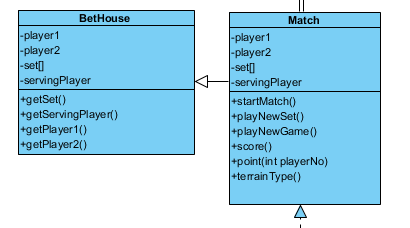


3.3. Private class data

The client was offered a contract with a bet company.

The betting company should be provided reading rights for the matches.

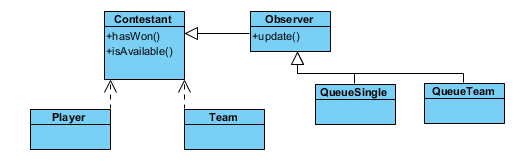
No extra information is necessary.



4. Behavioral patterns:

4.1. Observer

The client just opened a tennis club and wants to keep track of his clients’ status in order to help them find a matchup in case they do not have one.



4.2. Strategy

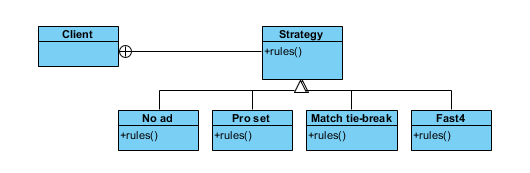
Customers asked the client for different form of scoring.

No ad: The first player or doubles team to win four points wins the game, regardless of whether the player or team is ahead by two points. When the game score reaches three points each, the receiver chooses which side of the court (advantage court or deuce court) the service is to be delivered on the seventh and game-deciding point.

Pro set: Instead of playing multiple sets, players may play one "pro set". A pro set is first to 8 (or 10) games by a margin of two games, instead of first to 6 games. A 12-point tie-break is usually played when the score is 8–8 (or 10–10). These are often played with no-ad scoring.

Match tie-break: This is sometimes played instead of a third set. A match tie-break (also called super tie-break) is played like a regular tie-break, but the winner must win ten points instead of seven.

Fast4: Fast4 is a shortened format that offers a "fast" alternative, with four points, four games and four rules: there are no advantage scores, lets are played, tie-breakers apply at three games all and the first to four games wins the set.



4.3. State

The client started to upgrade his courtyards and now they can be used for other events.

He would like to keep his courtyards occupied as much as possible. In order to change a courtyard from a game to an event location, the courtyard’s administration team requires some time in the meantime. When a customer asks for a yard, he will either be delivered a yard right away, a yard in a time frame or no yard due to no availability.

