Data Structure & Classes

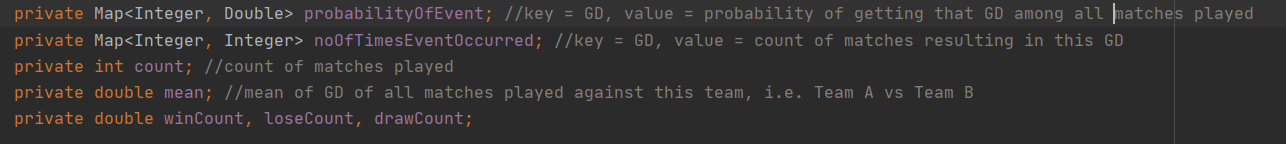
**RankingSystem.java**

This is the main class of our project. The main functionality of this class is to read the EPL datasets from season 2011-2012 till season 2019-2020 and to calculate the final standings of season 2019-2020 using our Ranking algorithm.

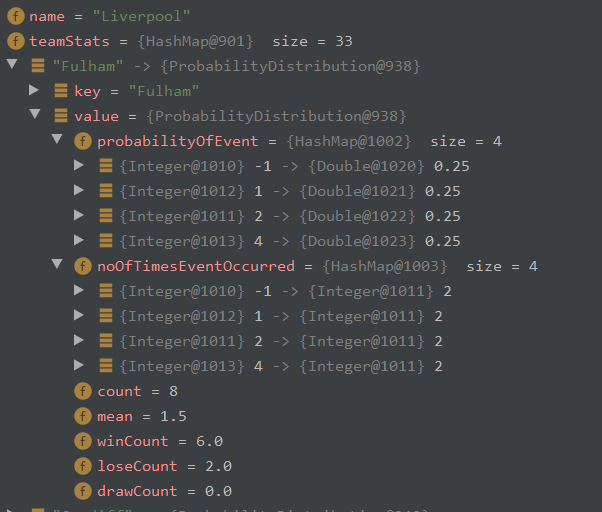
**ProbabilityDistribution.java**

The single instance of this class represents how many times Team A and Team B have played(count), the number of time TeamA win against Team B (winCount), the number of times team A loses against Team B(loseCount), the number of time Team A and Team B end up with a draw(drawCount), the Average number of Goal Difference of Team A against Team B(mean) and two maps that gives us how many times Team A have the same goal difference(key-> Goal Difference, Value-> the number of times GD occurred).

Below are the properties of this class:



E.g. Let us see the probability distribution instance for Liverpool against Fulham.

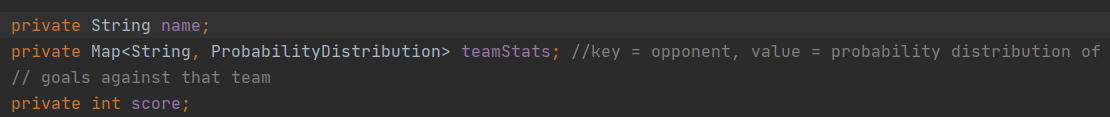


Two more important methods belong to this class and those methods are getMean and getSD, these two methods will calculate the average goal difference of Team A against Team B and average standard deviation in goal difference.

**Team.java**

This class represents the data associated with a particular team. The single instance of this class represents the name of the team(name), the score of the team(score), and a map that has Key as opponent team name and values as Probability Distribution of Team against the opponent(teamstats).

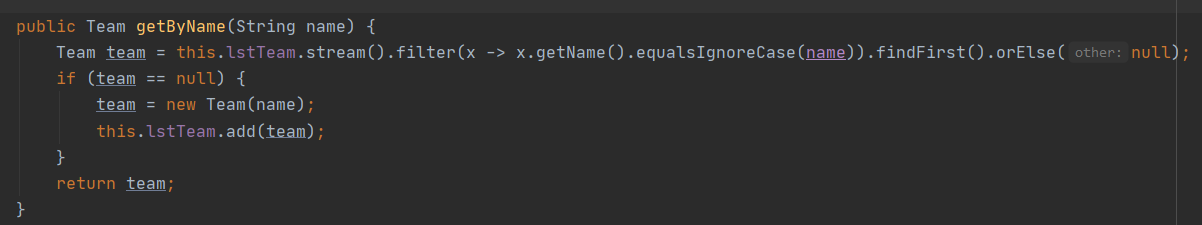
Below are the properties of this class:



Two more important methods belong to this class and those methods are getAvgMean and getAvgSD, these two methods will calculate the average goal difference of this team and average standard deviation in goal difference.

**Teams.java**

This class represents the collection of Team objects. The main functionality of this class is to check for the record of a particular team is available or not. If records are present then pass the same instance to the caller else create a new instance and return the new instance to the caller(getByName).



This class contains one more method i.e. compareTeams that calls the TeamComparator class methods(static).

**TeamComparator.java**

This class represents our algorithm to rank or compare two teams based on various parameters. We will discuss this class in more detail in the below section.

Algorithm and Formulae:

**Team Ranking algorithm:**

Step1: Get the Team instances of two teams those need to be compared.

Step2: Get the Probability Distribution instances of both teams.

Step3: Using Probability Distribution check if two teams have previous records(More than 4 matches) against each other else Go to Step 5

Step4: Return Goal Difference using getMean method from the probability distribution instance.

Step5: Check if both the team shares common opponents else Go to step 7

Step6: Calculate the Average Mean and Average Standard Deviation of goal difference of both teams against common opponents and then Return the Difference between both means as a result.

Step7: Get the Average Mean and Average Standard Deviation of both the teams and return the Difference between both means as a result.

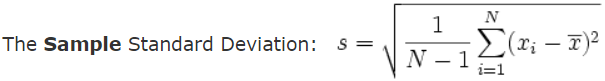
*Note: Whenever the Means are very close to each other(normal distribution graphs almost overlap each other) then to differentiate between them we can consider the standard deviation for result calculation. The team with less standard deviation is considered to be more consistent so we will be giving results by comparing standard deviation values in such cases.*

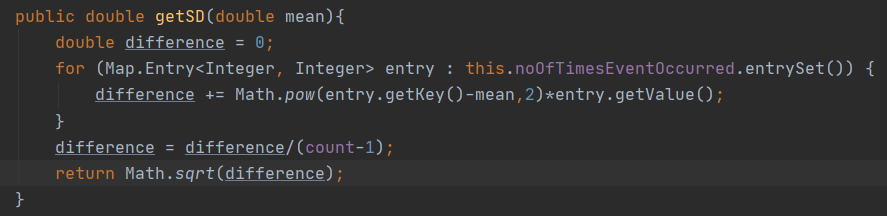
**Formulae:**

To calculate the mean we are using a general formula that first calculates the sum and then adds the new value to the sum and divides the new sum by the new count.

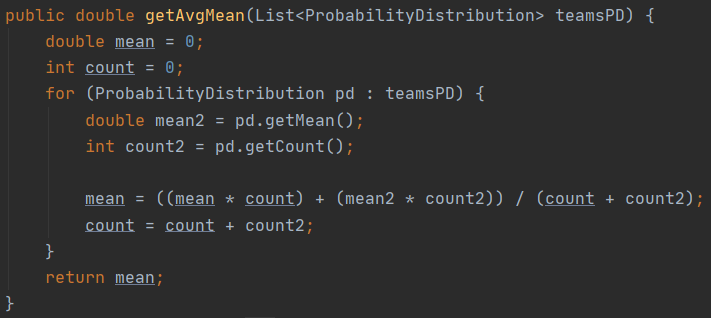


To calculate the standard deviation of the sample:



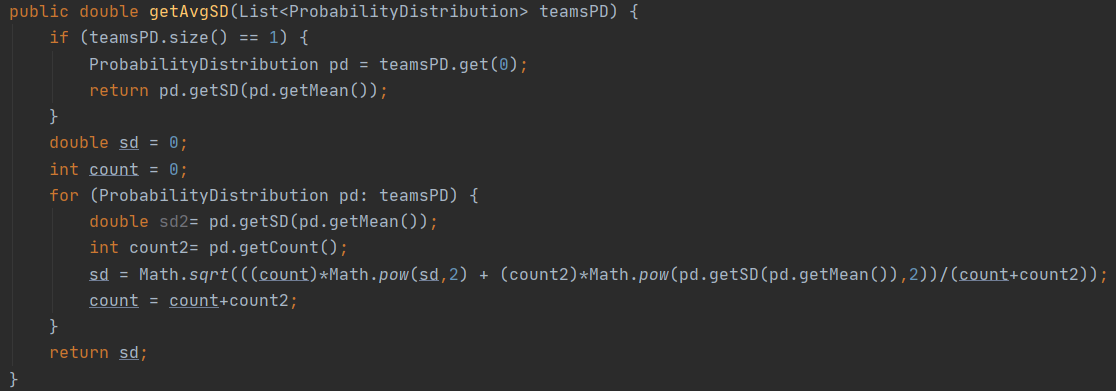


To merge two means: Calculating the sum of both the means and then calculating the new mean.



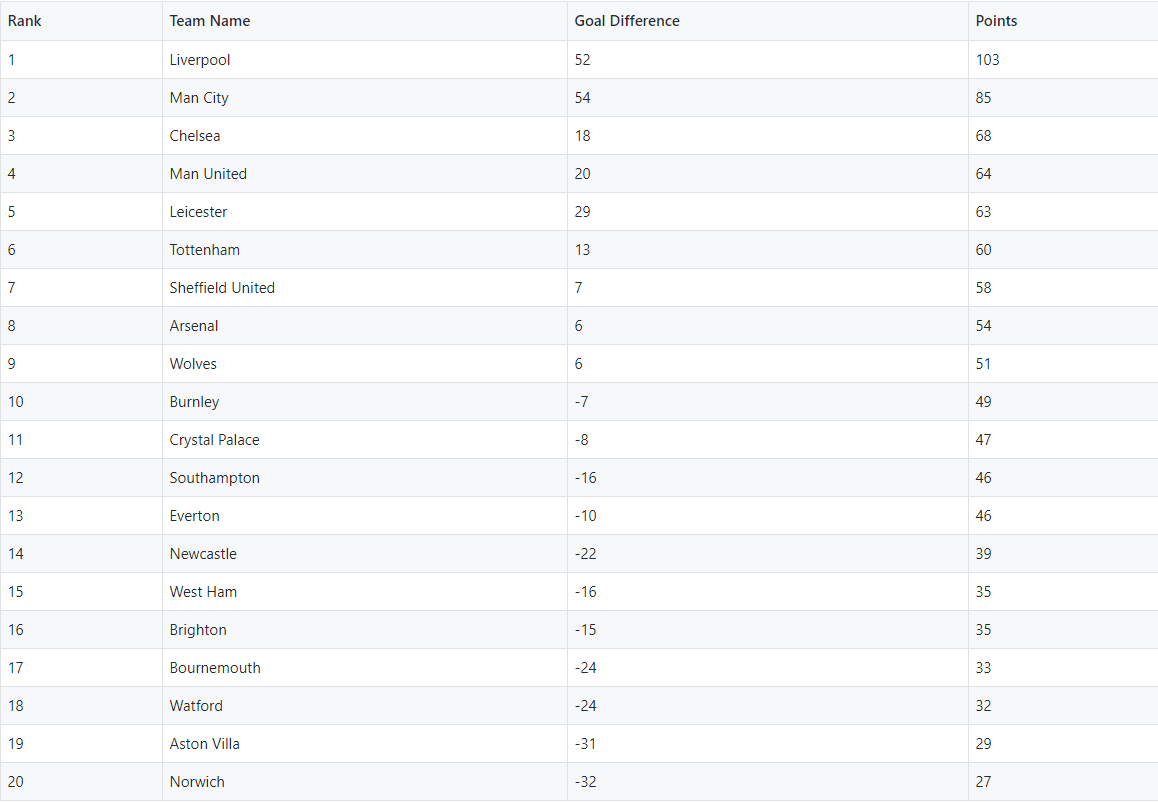
To merge two standard deviations:





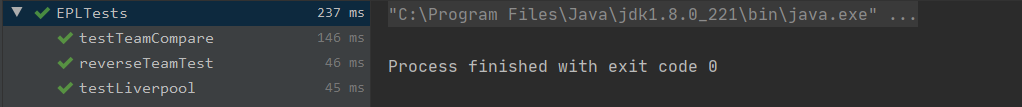
**Result and Analysis:**

Standings at the end of the season 2019-2020

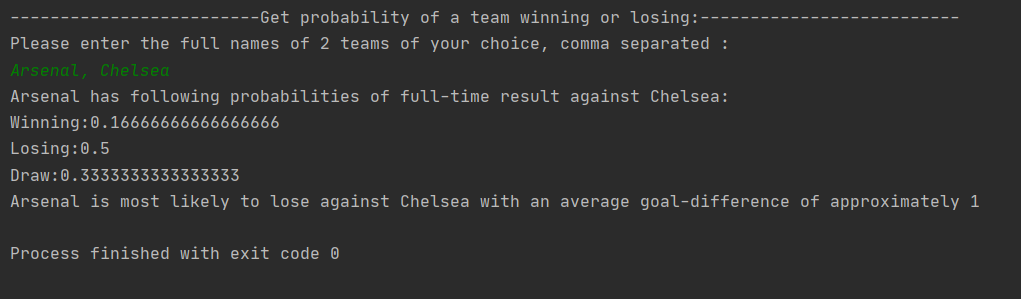


*(output of result.csv from GitHub)*

The output of test cases:



Command-line user input result:

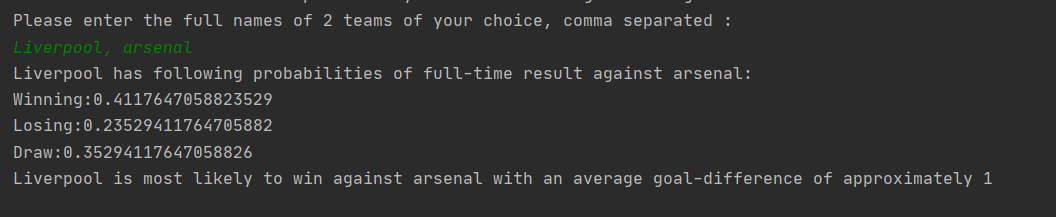


**Observations:**

Team Liverpool vs Team Arsenal goal difference graph

From the above graph, we can see that Liverpool has a high chance of winning against Arsenal.

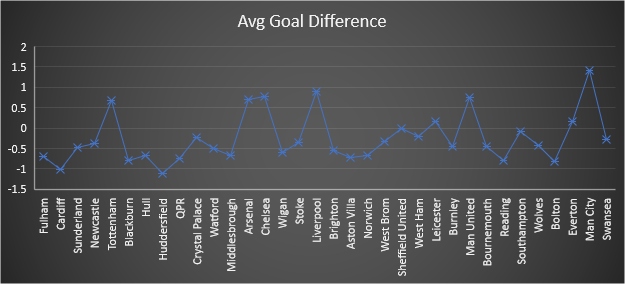
Let’s test this against command line input.

****

Here mean of Liverpool against Arsenal is approx 0.64

****

Let’s see the graphical representation of the average goal difference of all the teams.



From the graph it is clear that Man City on an average score more goal that is perfectly matches with the current circumstances of Manchester City since 2011.

**References:**

Calculating the Standard Deviation of a sample

<https://www.mathsisfun.com/data/standard-deviation-formulas.html>

Combining Standard Deviation of two groups

<https://math.stackexchange.com/questions/2971315/how-do-i-combine-standard-deviations-of-two-groups/2971522#2971522>