Challenge 2: Aggregate `Em All!

In this exercise, you have to perform aggregation between 3 classes.

WE'LL COVER THE FOLLOWING

- Problem Statement
- Coding Exercise

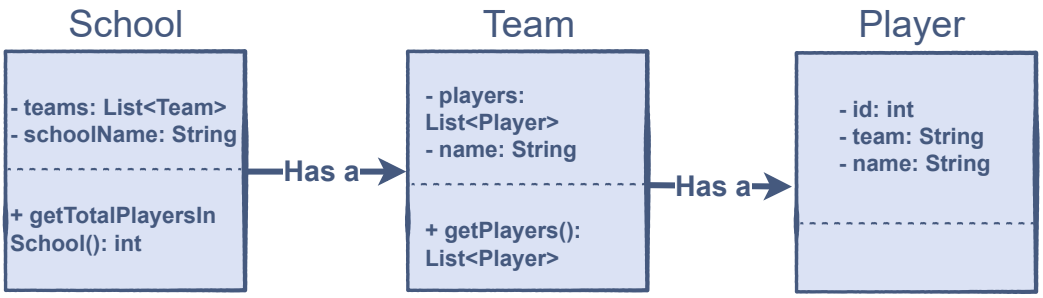
Problem Statement

You have to implement **3** classes:

```
class School {
}

class Team {
}

class Player {
}
```

in such a way that an instance of a `School` should contain instances of `Team` objects. Similarly, a `Team` object can contain instances of `Player` class.

Consider this diagram for clarification:



School, Team, Player: Class Representation

You have to implement a `School` class containing a list of `Team` objects, a `Team` class comprising of a list of `Player` objects and a `Player` class having an *id*, *name*, and *team*.

The `School` class will also have an implemented `getTotalPlayersInSchool()` function which eventually counts the total players in all of the **teams** in the **school** and returns the count!

So, your school should have these players in their respective teams:

Player ID's	Player Names	Teams
1	Harris	Red
2	Carol	Red
1	Johnny	Blue
2	Sarah	Blue

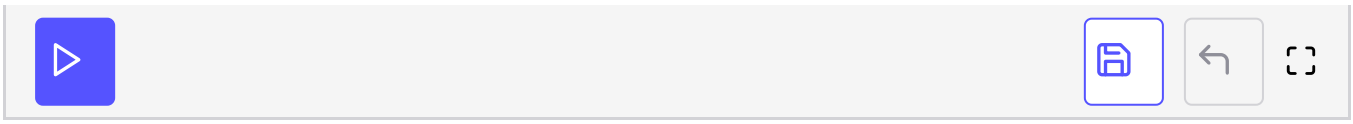
Coding Exercise

First, take a close look and design a step-by-step algorithm before jumping to the implementation. This problem is designed for your practice, so initially try to solve it on your own. If you get stuck, you can always refer to the solution provided in the solution review.

Good Luck!

```
// Player class
class Player {

    // Complete the implementation

}

/* Team class contains a list of Player
Objects.*/
class Team {
```

```
    // Complete the implementation


}

/* School class contains a list of Team
Objects.*/
class School {

    // Complete the implementation

}

// Main class
class Main {

    public static void main (String[] args) {
        // Write your code here
    }

}
```



The solution is explained in the next lesson!