**Sports Club exercise**

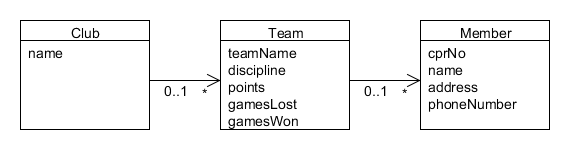
A sports club wants a system to keep track of their different teams, the teams’ scores in the tournaments and the members in each team.

To implement this system, you need three classes: Member, Team, and Club.

**Testing**

After each exercise you must test the program written so far. Make sure that previously implemented functionality has not been broken by your recent changes.

**Class Diagram**Here is the class diagram over the system showing the attributes described in the text.

****

**The Member class:**

1. The purpose of this class is to hold information about the members. This class must have the instance variable (fields) cprNo, name, address, phoneNumber.

The cprNo is 10 digits, containing the members birthday (first 6 digits), and a 4 digit number.

You decide the data types of the fields.

You must make a constructor for the class as well as set- and get- methods.

2. Add a method to the Member class that returns the birth year of a member. Use the following header: **public int getBirthYear()**

To convert a **String** to an **int** use: **int number = Integer.parseInt(XXXX);** where **XXXX** is the string you want to convert to an integer value.

(Hint: if you define the cprNo as a String, you can use the method substring to get the year, se in the BlueJ book page 46 in the method getLoginName).

**The Team class**

3. The purpose of this class is to hold information about a team. This class shall have the following instance variables: teamName, discipline, points, gamesLost, gamesWon and an ArrayList of members playing on the team.

Write a constructor that initializes the instance variables.

Write the set- and get- methods for the instance variables and a method to add members to the team.

4. Make a method in the Team class that returns the oldest team member (the lowest birth Year – Hint: use the method from exercise 2) Use the following header:

**public Member getOldestMember()**

5. Add a method to the Team class that returns the number of members on the team.

6. Add a method to the Team class which returns a member whose name matches the parameter of the method. (Given a name, it finds the first matching member object and returns it.) Use the following header:

**public Member getMember(String memberName)**

7. Add **a method** to the Team class that returns **true** if the number of games won is greater than the number of games lost and **false** if the team has lost most of its games. You make up the signature of this method.

The Club class

8. Make a new class called Club, the purpose of the class is to aggregate Teams. This aggregation is implemented by using an ArrayList. The Club class should have an attribute that stores the club name.

Add a constructor and a method to add a Team to the collection.

9. Add a method that given a team name can delete a Team from the collection. Use the following header:

**public void deleteTeam(String teamName)**

10. Add a method to the Club class that finds (return in an ArrayList) all the teams that have won more games than it they have lost. (Hint: maybe you can use the method form exercise 7). Use the following header: **public ArrayList wonMostGames()**

11 Make a method, which given a name of a member, returns the team he is playing on.

12. Make a method that can return the most senior member (the oldest of all the members).