

LAB 2 - PROGRAMACIÓN ORIENTADA A OBJETOS

Implementación de las relaciones entre clases

El objetivo de este laboratorio es seguir desarrollando la implementación del diseño previamente hecho en los seminarios 1 y 2 , rescatando el código hecho en la práctica 1 y realizando una serie de cambios. Este consta de implementar las clases Match y League.

La información de relación entre clases facilitada por el profesorado define que: La clase League tiene una relación de composición con la clase Match, una relación de agregación con la clase Team, y una relación de asociación con la clase Country. Por otro lado, Match tiene una relación de composición con la clase League, una relación de asociación con la clase Team y una relación de asociación con la clase Player.

Sabiendo esto, la clase Match tiene que contener los siguientes atributos: *homeTeam* (variable del tipo Team, clase creada durante la práctica 1), *awayTeam* (variable del tipo Team, clase creada durante la práctica 1), *homeGoals*, *awayGoals*, *homeScorers* (lista de jugadores del tipo Player de la clase creada durante la práctica 1), *awayScorers* (lista de jugadores del tipo Player de la clase creada durante la práctica 1). En este caso, los métodos a definir son el constructor: Match que cogerá los atributos *homeTeam*, *awayTeam*, los siguientes getters: *getHomeTeam()*, *getAwayTeam()*, *getGoalsHome()*, *getGoalsAway()*, *getHomeScorers()*, *getAwayScorers()*, *getListHomeScorers()*, *getListAwayScorers()*, *simulateMatch()* y para acabar el método *printMatch()*. También ha sido preciso la implementación de los getters *getSizeListHomeScorers()* y *getSizeListAwayScorers()*.

Para hacer posible la implementación de *simulateMatch()* era preciso la creación de un número aleatorio de goles para ambos equipos.

```
public void simulateMatch() {  
    //Generamos los goles de cada equipo  
    int n = 6;  
    int m = 6;  
    homeGoals = random.nextInt(n) ;  
    awayGoals = random.nextInt(m) ;  
}
```

Esta función también crea la lista de goleadores de forma aleatoria para cada equipo.

La función *printMatch()* simplemente imprime los resultados generados aleatoriamente por la función *simulateMatch*.

La clase League tiene que contener los siguientes atributos: *name*, *country* (variable del tipo Country de la clase ya proporcionada en el código), *gender*, *teams* (lista de jugadores del tipo Team de la clase creada previamente) y *matches* (lista de partidos del tipo Match de la clase creada recientemente). El constructor de esta clase se trata de la función *League()* en la que entran como parámetros el nombre de la liga, el país de la liga y el género de la liga. Por otro lado, esta clase tiene por métodos las funciones: *getName()*, *getCountry()*, *getGender()*, *addTeam()*, *removeTeam()*, *generateMatches()*, *simulateMatches()*, *printMatches()*, *printTable()* y por último la función *printGoalScorers()*, estas dos últimas no era necesaria su programación.

Para implementar los métodos *addTeam()* y *removeTeam()*, seguimos la misma estrategia que en *addPlayer()* y *removePlayer()* , es decir, nos aseguramos que el gender del team coincide con el gender de la liga en la que queremos añadir.

```
//methods
public void addTeam(Team t){
    if (t.getGender() == Gender.HOMBRE && gender == Gender.HOMBRE) {
        teams.add(counterTeams, t);
        counterTeams++;
        System.out.println(t.getName() + " added sucessfully");
    } else if(t.getGender() == Gender.MUJER && gender == Gender.MUJER){
        teams.add(counterTeams, t);
        counterTeams++;
        System.out.println(t.getName() + " added sucessfully");
    }
    else if((t.getGender() == Gender.HOMBRE || t.getGender() == Gender.MUJER) && gender == Gender.MIXTO){
        teams.add(counterTeams, t);
        counterTeams++;
        System.out.println(t.getName() + " added sucessfully");
    } else {
        System.out.println(t.getName() + " can not be added due to their gender type.");
    }
}
```

Para la implementación de *generateMatches()* nos costó encontrar la forma de implementar la función, pero finalmente entendimos la relación con un doble loop. Teniendo una lista de equipos apuntados en la liga el primer loop recorre de forma normal la lista, mientras que el segundo loop para cada elemento de la lista sobre el que se itera, genera los partidos con el resto de elementos de la lista ya que se comprueba que los equipos sobre los que se está generando un partido no sean iguales.

```
public void generateMatches () {
    for (int i = 0; i < counterTeams; i++){
        for(int j = 0; j < counterTeams; j++){
            if (i != j) {
                Match m = new Match(getTeam(i), getTeam(j));
                matches.add(i, m);
                counterMatches++;
            }
        }
    }
}
```

La función *simulateMatches()* simplemente sabiendo los partidos generados llama a la función creada en la clase Match *simulateMatch* para que simule cada partido generado por *generateMatches()*. Esta función acaba por actualizar la estadísticas de ambos equipos de los que *simulateMatch* ha generado resultados y printea los resultados actualizados de los equipos y de los jugadores de los equipos.

Esto último se tuvo que implementar por la simple razón de que sino no se podía verificar correctamente que las estadísticas de los jugadores y los equipos se fueran actualizando en cada partido en el main method, además también sirve para poder mirar cómo evolucionan los equipos y comprobar que efectivamente las estadísticas finales concuerdan con el transcurso de los partidos.

```
public void simulateMatches(){
    for(int i = 0; i < counterMatches; i++){

        //simulamos partido
        Match m = getMatch(i);
        m.simulateMatch();

        //actualizamos stats de los equipos
        m.getHomeTeam().updateStats(m);
        m.getAwayTeam().updateStats(m);

        //actualizamos stats jugadores
        for(int j = 0; j < m.getHomeTeam().getNumPlayers() ; j++){
            Player p = m.getHomeTeam().getPlayer(j);
            p.updateStatsPlayer(m);
        }
        for(int j = 0; j < m.getAwayTeam().getNumPlayers(); j++){
            Player p = m.getAwayTeam().getPlayer(j);
            p.updateStatsPlayer(m);
        }

        //imprimimos las stats del partido jugado
        System.out.println(x:"\n
        System.out.println("\nMatch " + (i + 1));
        m.printMatch();
        m.getHomeTeam().printStats();
        m.getAwayTeam().printStats();

        //imprimimos stats de los jugadores
        System.out.println("\nStats match " + (i + 1));
        System.out.println(m.getHomeTeam().getName() + " players:");
        for (int j = 0; j < m.getHomeTeam().getSizeListPlayers(); j++) {
            Player player = m.getHomeTeam().getPlayer(j);
            player.printStatsPlayer();
        }

        System.out.println("\n" + m.getAwayTeam().getName() + " players:");
        for (int j = 0; j < m.getAwayTeam().getSizeListPlayers(); j++) {
            Player player = m.getAwayTeam().getPlayer(j);
            player.printStatsPlayer();
        }
    }
}
```

Por tanto con la nueva implementación de la clase Match la función *updateStats()* de las clases Team y Player se vieron modificadas ya que toda la información necesaria se encuentra en Match.

Función inicial de *updateStats()*:

```
public void playMatch(int scored, int against){
    goalsScored += scored;
    goalsAgainst += against;
    if(scored > against){
        noWins +=1;
    }else if (scored < against){
        noLosses +=1;
    }else {
        noTies +=1;
    }
    noMatches +=1;
}
```

Función modificada de *updateStats()*:

```
public void updateStats (Match m) {
    if(m.getHomeTeam().name == getName()){
        //actualizamos goles marcados
        goalsScored += m.getGoalsHome();
        goalsAgainst += m.getGoalsAway();
        if(m.getGoalsHome() > m.getGoalsAway()){
            noWins += 1;
        }else if (m.getGoalsHome() < m.getGoalsAway()){
            noLosses += 1;
        }else {
            noTies += 1;
        }
    }else if(m.getAwayTeam().name == getName()){
        //actualizamos goles marcados
        goalsScored += m.getGoalsAway();
        goalsAgainst += m.getGoalsHome();
        if(m.getGoalsHome() > m.getGoalsAway()){
            noLosses += 1;
        }else if (m.getGoalsHome() < m.getGoalsAway()){
            noWins += 1;
        }else {
            noTies += 1;
        }
    }
    noMatches += 1;
}
```

Al finalizar la implementación, hemos definido una clase de prueba *TestApplication* con un método principal, hemos creado instancias de las diferentes clases, aplicando los diversos métodos y compilando frecuentemente para asegurarnos de que no hay errores.

Para hacerlo posible, primero empezamos creando una nueva liga, tres nuevos Team y seguimos con la creación de seis Jugadores.

Después de añadir los diferentes jugadores a cada equipo, y los equipos a la liga, llamamos los métodos correspondientes para generar todos los partidos y simularlos. Al final de todo imprimimos los stats finales para cada equipo.

Primero hicimos pruebas con solo dos equipos (Foto1) y para verificar que funcionaba todo correctamente hicimos más pruebas (Foto2) pero con tres equipos y poder comprobar que se jugaron seis partidos.

Con todo esto pudimos observar que todo funcionaba correctamente.

(Foto1)

```
Lionel Messi added sucessfully
Luis Suárez added sucessfully
Sergio Ramos added sucessfully
Karim Benzema added sucessfully

Match 1
FC Barcelona --> Home team: 2
Real Madrid --> Away team: 5

Home scorers: [Lionel Messi, Luis Suárez]
Away scorers: [Sergio Ramos, Sergio Ramos, Sergio Ramos, Sergio Ramos, Karim Benzema]

FC Barcelona: Matches: 1, Wins: 0, Ties: 0, Losses: 1, Goals scored: 2, Goals against: 5
Real Madrid: Matches: 1, Wins: 1, Ties: 0, Losses: 0, Goals scored: 5, Goals against: 2

Stats match 1
FC Barcelona players:
Lionel Messi--> Matches: 1, Tackles: 4, Passes: 0, Shots: 2, Assists: 2, Goals: 1
Luis Suárez--> Matches: 1, Tackles: 2, Passes: 3, Shots: 3, Assists: 1, Goals: 1

Real Madrid players:
Sergio Ramos--> Matches: 1, Tackles: 5, Passes: 6, Shots: 2, Assists: 0, Goals: 4
Karim Benzema--> Matches: 1, Tackles: 3, Passes: 1, Shots: 3, Assists: 1, Goals: 1

Match 2
Real Madrid --> Home team: 0
FC Barcelona --> Away team: 0

Home scorers: []
Away scorers: []

Real Madrid: Matches: 2, Wins: 1, Ties: 1, Losses: 0, Goals scored: 5, Goals against: 2
FC Barcelona: Matches: 2, Wins: 0, Ties: 1, Losses: 1, Goals scored: 2, Goals against: 5

Stats match 2
Real Madrid players:
Sergio Ramos--> Matches: 2, Tackles: 5, Passes: 11, Shots: 3, Assists: 0, Goals: 4
Karim Benzema--> Matches: 2, Tackles: 4, Passes: 6, Shots: 3, Assists: 1, Goals: 1

FC Barcelona players:
Lionel Messi--> Matches: 2, Tackles: 4, Passes: 7, Shots: 3, Assists: 4, Goals: 1
Luis Suárez--> Matches: 2, Tackles: 8, Passes: 9, Shots: 4, Assists: 3, Goals: 1
```

(Foto2)

Lionel Messi added successfully
Luis Suárez added successfully
Sergio Ramos added successfully
Karim Benzema added successfully
Ingrid Engen added successfully
Irene Paredes added successfully
FC Barcelona added successfully
Real Madrid added successfully
FC Barcelona Femenino added successfully

Match 1

FC Barcelona --> Home team: 3
FC Barcelona Femenino --> Away team: 0

Home scorers: [Lionel Messi, Luis Suárez, Luis Suárez]
Away scorers: []

FC Barcelona: Matches: 1, Wins: 1, Ties: 0, Losses: 0, Goals scored: 3, Goals against: 0

FC Barcelona Femenino: Matches: 1, Wins: 0, Ties: 0, Losses: 1, Goals scored: 0, Goals against: 3

Stats match 1

FC Barcelona players:

Lionel Messi--> Matches: 1, Tackles: 6, Passes: 0, Shots: 1, Assists: 0, Goals: 1
Luis Suárez--> Matches: 1, Tackles: 5, Passes: 6, Shots: 0, Assists: 0, Goals: 2

FC Barcelona Femenino players:

Ingrid Engen--> Matches: 1, Tackles: 6, Passes: 4, Shots: 0, Assists: 1, Goals: 0
Irene Paredes--> Matches: 1, Tackles: 4, Passes: 2, Shots: 2, Assists: 1, Goals: 0

Match 2

Real Madrid --> Home team: 0
FC Barcelona Femenino --> Away team: 3

Home scorers: []
Away scorers: [Irene Paredes, Ingrid Engen, Irene Paredes]

Real Madrid: Matches: 1, Wins: 0, Ties: 0, Losses: 1, Goals scored: 0, Goals against: 3

FC Barcelona Femenino: Matches: 2, Wins: 1, Ties: 0, Losses: 1, Goals scored: 3, Goals against: 3

Stats match 2

Real Madrid players:

Sergio Ramos--> Matches: 1, Tackles: 1, Passes: 4, Shots: 3, Assists: 0, Goals: 0
Karim Benzema--> Matches: 1, Tackles: 4, Passes: 9, Shots: 3, Assists: 1, Goals: 0

FC Barcelona Femenino players:

Ingrid Engen--> Matches: 2, Tackles: 8, Passes: 12, Shots: 1, Assists: 3, Goals: 1
Irene Paredes--> Matches: 2, Tackles: 7, Passes: 4, Shots: 2, Assists: 2, Goals: 2

Match 3

FC Barcelona Femenino --> Home team: 2
Real Madrid --> Away team: 5

Home scorers: [Irene Paredes, Irene Paredes]
Away scorers: [Sergio Ramos, Sergio Ramos, Karim Benzema, Sergio Ramos, Sergio Ramos]

FC Barcelona Femenino: Matches: 3, Wins: 1, Ties: 0, Losses: 2, Goals scored: 5, Goals against: 8

Real Madrid: Matches: 2, Wins: 1, Ties: 0, Losses: 1, Goals scored: 5, Goals against: 5

Stats match 3

FC Barcelona Femenino players:

Ingrid Engen--> Matches: 3, Tackles: 13, Passes: 19, Shots: 2, Assists: 4, Goals: 1
Irene Paredes--> Matches: 3, Tackles: 8, Passes: 13, Shots: 3, Assists: 2, Goals: 4

Real Madrid players:

Sergio Ramos--> Matches: 2, Tackles: 1, Passes: 9, Shots: 4, Assists: 2, Goals: 4
Karim Benzema--> Matches: 2, Tackles: 9, Passes: 16, Shots: 3, Assists: 3, Goals: 1

Match 4

FC Barcelona Femenino --> Home team: 1

FC Barcelona --> Away team: 0

Home scorers: [Ingrid Engen]

Away scorers: []

FC Barcelona Femenino: Matches: 4, Wins: 2, Ties: 0, Losses: 2, Goals scored: 6, Goals against: 8

FC Barcelona: Matches: 2, Wins: 1, Ties: 0, Losses: 1, Goals scored: 3, Goals against: 1

Stats match 4

FC Barcelona Femenino players:

Ingrid Engen--> Matches: 4, Tackles: 15, Passes: 19, Shots: 5, Assists: 5, Goals: 2

Irene Paredes--> Matches: 4, Tackles: 11, Passes: 19, Shots: 5, Assists: 4, Goals: 4

FC Barcelona players:

Lionel Messi--> Matches: 2, Tackles: 7, Passes: 3, Shots: 3, Assists: 2, Goals: 1

Luis Suárez--> Matches: 2, Tackles: 10, Passes: 6, Shots: 1, Assists: 1, Goals: 2

Match 5

Real Madrid --> Home team: 0

FC Barcelona --> Away team: 1

Home scorers: []

Away scorers: [Lionel Messi]

Real Madrid: Matches: 3, Wins: 1, Ties: 0, Losses: 2, Goals scored: 5, Goals against: 6

FC Barcelona: Matches: 3, Wins: 2, Ties: 0, Losses: 1, Goals scored: 4, Goals against: 1

Stats match 5

Real Madrid players:

Sergio Ramos--> Matches: 3, Tackles: 1, Passes: 15, Shots: 6, Assists: 3, Goals: 4

Karim Benzema--> Matches: 3, Tackles: 15, Passes: 23, Shots: 6, Assists: 3, Goals: 1

FC Barcelona players:

Lionel Messi--> Matches: 3, Tackles: 12, Passes: 5, Shots: 4, Assists: 3, Goals: 2

Luis Suárez--> Matches: 3, Tackles: 16, Passes: 6, Shots: 1, Assists: 2, Goals: 2

Match 6

FC Barcelona --> Home team: 3

Real Madrid --> Away team: 3

Home scorers: [Luis Suárez, Luis Suárez, Luis Suárez]

Away scorers: [Karim Benzema, Sergio Ramos, Karim Benzema]

FC Barcelona: Matches: 4, Wins: 2, Ties: 1, Losses: 1, Goals scored: 7, Goals against: 4

Real Madrid: Matches: 4, Wins: 1, Ties: 1, Losses: 2, Goals scored: 8, Goals against: 9

Stats match 6

FC Barcelona players:

Lionel Messi--> Matches: 4, Tackles: 16, Passes: 14, Shots: 5, Assists: 5, Goals: 2

Luis Suárez--> Matches: 4, Tackles: 16, Passes: 13, Shots: 4, Assists: 4, Goals: 5

Real Madrid players:

Sergio Ramos--> Matches: 4, Tackles: 6, Passes: 21, Shots: 9, Assists: 5, Goals: 5

Karim Benzema--> Matches: 4, Tackles: 16, Passes: 25, Shots: 9, Assists: 3, Goals: 3

FINAL STATS OF TEAMS:

FC Barcelona: Matches: 4, Wins: 2, Ties: 1, Losses: 1, Goals scored: 7, Goals against: 4

Lionel Messi--> Matches: 4, Tackles: 16, Passes: 14, Shots: 5, Assists: 5, Goals: 2

Luis Suárez--> Matches: 4, Tackles: 16, Passes: 13, Shots: 4, Assists: 4, Goals: 5

Real Madrid: Matches: 4, Wins: 1, Ties: 1, Losses: 2, Goals scored: 8, Goals against: 9

Sergio Ramos--> Matches: 4, Tackles: 6, Passes: 21, Shots: 9, Assists: 5, Goals: 5

Karim Benzema--> Matches: 4, Tackles: 16, Passes: 25, Shots: 9, Assists: 3, Goals: 3

FC Barcelona Femenino: Matches: 4, Wins: 2, Ties: 0, Losses: 2, Goals scored: 6, Goals against: 8

Ingrid Engen--> Matches: 4, Tackles: 15, Passes: 19, Shots: 5, Assists: 5, Goals: 2

Irene Paredes--> Matches: 4, Tackles: 11, Passes: 19, Shots: 5, Assists: 4, Goals: 4