Université de Carthage

Institut national des sciences appliquées et de technologie

Année Universitaire 2018-2019

Rapport
Mini-Projet 1

Réalisé par :

SNOUSSI Anis

BARGHOUDA Mohamed Lamine

CONTENU

Description du projet

Le projet consiste à créer une application permettant de gérer une compétition d'un match de football.
La compétition doit être, tout d'abord, définie dans le programme, cela nécessite seulement son nom, et puis on peut ajouter les éléments suivants :

- La liste des équipes participantes
- La liste des entraineurs
- La liste des arbitres actifs
- La listes des matchs qui se sont déroulés

La compétition se déroule sur deux tours : Aller et Retour, qui doivent être joués dans l'ordre Aller puis Retour (mais cela n'interdit pas les équipes de jouer des matchs avec d'autres équipes avant de jouer le Retour avec une équipe donnée).

La compétition est de type championnat, de tel façon que chaque équipe admet un nombre de points qui détermine son classement.
Le calcul des points suit les règles suivantes :

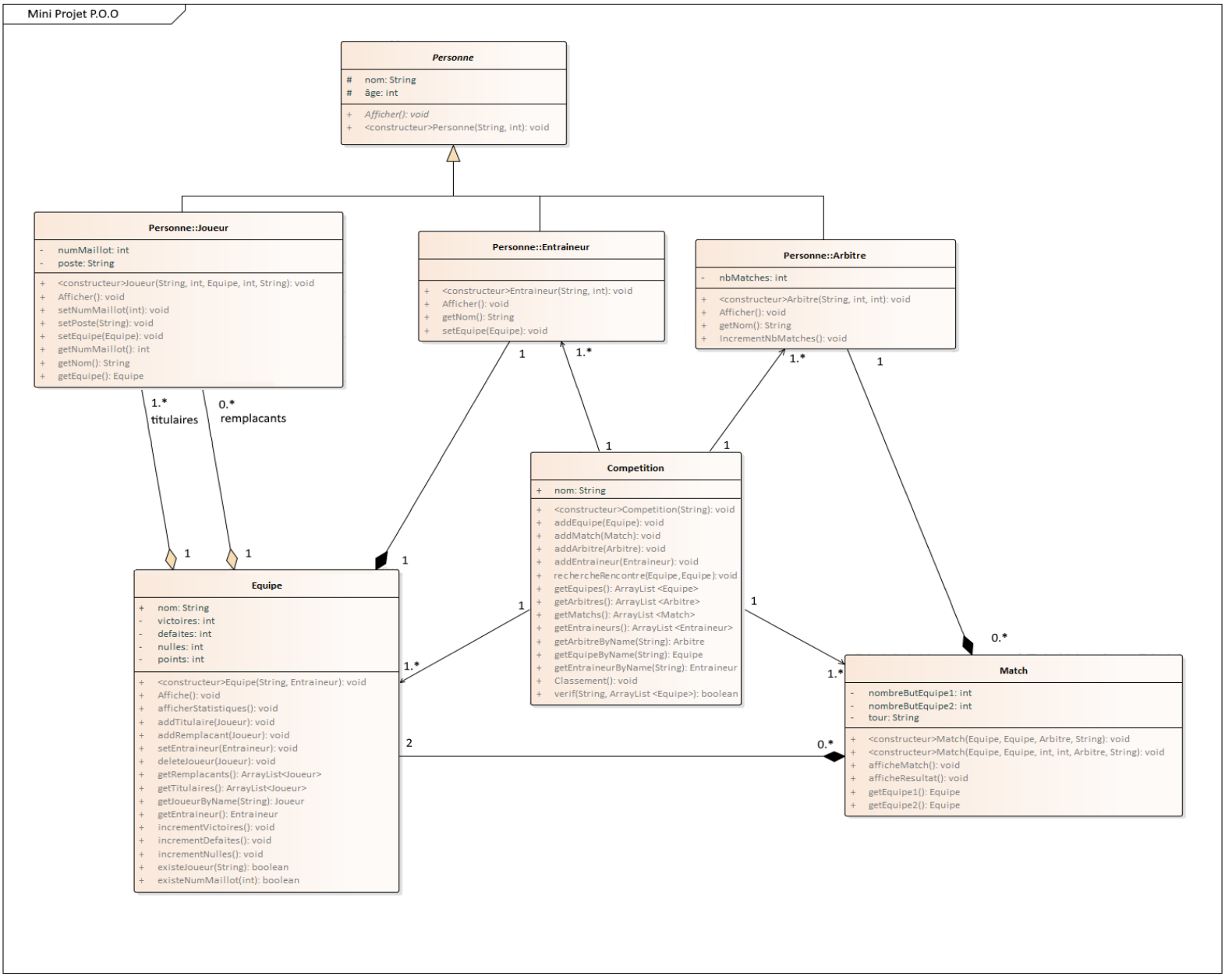
- Une victoire vaut 3 points.
- Une défaite vaut 0 points.
- Une nulle vaut 1 point.

Comme ce projet est visé à gérer cette compétition, l'utilisateur a le pouvoir d'exécuter plusieurs tâches :

- L'ajout et la Mise à jour des joueurs **(tous changer sauf le nom et l'âge)**, des équipes **(gestion des joueurs titulaires et remplaçants, seulement l'incrémentation de nombre des victoires/nulles/défaites)**, des arbitres **(seulement l'incrémentation de nombre de matchs)** et des entraineurs **(seulement changement d'équipe)**.
- La saisie des matchs : en utilisant deux connecteurs différents **(nombre de buts connue ou aléatoire).**
- L'affichage des résultats des matchs ou la liste des joueurs d'une équipe donnée.
- La recherche de l'historiques de rencontres entre 2 équipes données.
- Les statistiques d'une équipe donnée (victoires, défaites, nulles).
- L'affichage de classement actuel de la compétition.

Présentation du projet

Diagramme de classe



Implémentation

L'exécution du programme vous ramène à un menu dont on va décrire la signification :

1. **Effectuer le Test Complet de la Compétition** : le programme ajoute automatiquement 3 équipes contenant leurs entraineurs et un nombre différent de joueurs et 3 arbitres puis effectue tous les matchs entre ces équipes (le résultat est aléatoire) laissant à l'utilisateur la saisie d'une 4ème équipe pour tester les fonctionnalités du programme.
2. **Ajout ou mise à jour joueur** : comme son nom l'indique, permet l'ajout d'un joueur ou de modifier, si c'est possible, les attributs d'un joueur donné.
3. **Ajout d'une équipe :** ajouter une équipe
4. **Ajout ou mise à jour entraineur :** ajout ou mise à jour de l'équipe entrainée.
5. **Ajout Arbitre :** ajouter un arbitre
6. **Saisie d'un match :** entrer le résultat d'une rencontre entre deux équipes
7. **Affichage des résultats des matchs** : afficher les résultats de l'ensemble des matchs joués dans la compétition.
8. **Afficher une équipe** : afficher la liste des joueurs titulaires et remplaçants d'une équipe
9. **Afficher l'historique d'une rencontre entre deux équipes** : afficher toutes les rencontres entre deux équipes (dans notre cas : Aller et Retour)
10. **Statistiques d'une équipe** : afficher les victoires, nulles et défaites d'une équipe
11. **Classement de la compétition** : afficher le classement actuel des équipes
12. **Importation des données** : importation des données à partir des fichiers CSV
13. **Exit**

Dans tous les cas, le programme fait tous les tests nécessaires pour assurer la validité des informations entrées et éviter les redondances.

Difficultés rencontrées (problème : solution)

- Nombre Imprévisible des équipes, joueurs, arbitres et entraineurs : utilisation de ArrayList aux lieux des tableaux à tailles fixes.
- Grand nombre de données : lecture d'après les fichiers CSV.

Code source

➢ Package : Personne

• Personne.java

```
package Personne;
import java.util.*;

public abstract class Personne {
protected String nom;
protected int age;

static public Scanner Sc = new Scanner(System.in);

public Personne(String nom,int age) {
        this.nom= nom;
        this.age= age;
}

public abstract void Afficher();

}
```

• Arbitre.java

```
package Personne;

public class Arbitre extends Personne{
private int nbMatches;

public Arbitre(String nom,int age,int NbMathches){
     super(nom,age);
     this.nbMatches= NbMathches; // Contient le nombre de matchs déjà déroulés par un
arbitre avant qu'il est ajouté à la base de donnée
}

        public void Afficher() {
                System.out.println("Nom Arbitre : " + nom);
                System.out.println("Age Arbitre : " + age);
                System.out.println("Nombre de Matches :"+ nbMatches);
                }

    // Le nombre de matches s'incrémente automatiquement si un arbitre vient de se charge
de la déroulement d'un match
   public void IncrementNbMatches() {
           nbMatches++;
   }

    // on en besoin à l'affichage d'un match
    public String getNom() {
        return nom;
    }
```

- **Entraineur.java**

```
package Personne;
import Sport.Equipe;

public class Entraineur extends Personne {
private Equipe equipe;

public Entraineur(String nom,int age){
    super(nom,age);
}

        public void Afficher() {
                System.out.println("Nom Entraineur : " + nom);
                System.out.println("Age Entraineur : " + age);
                System.out.println("Equipe entrainé :"+ equipe.nom);
        }

        // Un entraineur peut changer l'equipe qu'il l'entraine
        public void setEquipe(Equipe e) {
                equipe = e;
        }

        public String getNom() {
                return nom;
        }
```

- **Joueur.java**

```
package Personne;
import Sport.Equipe;

public class Joueur extends Personne{
private int numMaillot;
private String poste;
private Equipe equipe;

public Joueur(String nom,int age,Equipe equipe,int numMaillot,String poste) {
        super(nom,age);
        this.equipe=equipe;
        this.numMaillot = numMaillot;
    this.poste = poste;
}

public void Afficher() {
        System.out.println("Nom Joueur : " + nom);
        System.out.println("Equipe actuel : "+ equipe.nom);
        System.out.println("Numero Maillot : " + numMaillot);
        System.out.println("Poste : " + poste);
        }
```

```
// Un joueur peut changer son numero de maillot
public void setNumMaillot(int x ) {
        numMaillot = x;
}

// un joueur peut changer sa poste de jeu
public void setPoste(String p) {
        poste = p;
}

// Un joueur peut changer d'equipe
public void setEquipe(Equipe E) {
        equipe = E;
}

public int getNumMaillot() {
        return numMaillot;
}

public Equipe getEquipe() {
        return equipe;
}

public String getNom() {
        return nom;
}

}
```

- **Package : Sport**
  - **Competition.java**

```
package Sport;
import java.util.*;
import Personne.*;

public class Competition{

        public String nom;
        private ArrayList <Equipe> equipes;
        private ArrayList <Match> matchs;
        private ArrayList <Arbitre> arbitres;
        private ArrayList <Entraineur> entraineurs;
```

```
// Constructeur
public Competition(String nom) {
        this.nom = nom;
        this.equipes = new ArrayList<>();
        this.matchs = new ArrayList<>();
        this.arbitres = new ArrayList<>();
        this.entraineurs = new ArrayList<>();
}


public void addEquipe(Equipe E) {
        equipes.add(E);
}


public void addMatch(Match M) {
        matchs.add(M);
}


public void addArbitre(Arbitre A) {
        arbitres.add(A);
}


public void addEntraineur(Entraineur E) {
        entraineurs.add(E);
}


public ArrayList <Equipe> getEquipes() {
        return equipes;
}


public ArrayList <Arbitre> getArbitres(){
        return arbitres;
}


public ArrayList <Match> getMatchs(){
        return matchs;
}


public ArrayList <Entraineur> getEntraineurs(){
        return entraineurs;
}
```

```
        public void rechercheRencontre(Equipe e1,Equipe e2){
                System.out.println("** Historique de rencontres entre "+ e1.nom + " Vs " +
e2.nom);
                for(int i=0;i<matchs.size();i++) {
                        if(e1.equals(matchs.get(i).getEquipe1()) &&
e2.equals(matchs.get(i).getEquipe2()))
                                matchs.get(i).afficheMatch();;
                }

        }

        // Necessaire pour le MAJ d'un Arbitre et pour éviter la redondance lors de saisie
        public Arbitre getArbitreByName(String name) {
                for(int i=0;i<arbitres.size();i++) {
                        if(arbitres.get(i).getNom().equals(name))
                                return arbitres.get(i);
                }
                return null;

        }

        // Necessaire pour le MAJ d'un équipe, l'affichage et pour éviter la redondance lors
de saisie
        public Equipe getEquipeByName(String name) {
                for(int i=0;i<equipes.size();i++) {
                        if (equipes.get(i).nom.equals(name))
                                return equipes.get(i);
                }
                return null;
        }

        // Necessaire pour le MAJ d'un Entraineur et pour éviter la redondance lors de saisie
        public Entraineur getEntraineurByName(String name) {
                for (int i=0;i<entraineurs.size();i++) {
                        if(entraineurs.get(i).getNom().equals(name))
                                return entraineurs.get(i);
                }
                return null;
        }
```

```
        // On avait un problème lors du classement des équipes selon le nombre de points en
utilisant l'ArrayList
        // La problème est résolu en utilisant une ArrayList auxiliare pour le classement
dernier
        // Cette méthode est dedié pour vérifier si un équipe est déjà classé (existe) dans
l'ArrayList auxiliare contenant les équipes classés

        private boolean verif(String nom, ArrayList <Equipe> aux) {
                for(int i=0;i<aux.size();i++) {
                        if (aux.get(i).nom.equals(nom))
                                return true;
                }
                return false ;
        }

        public void Classement() {
                ArrayList <Equipe> aux = new ArrayList<>();
                int min=0;
                Equipe temp = null;
                for(int j=0;j<equipes.size();j++) {
                        min = 0;
                        temp = null;
                for(int i=0;i<equipes.size();i++){
                if((equipes.get(i).getPoints() >= min) && (! verif(equipes.get(i).nom,aux))) {
                temp = equipes.get(i);
                min = equipes.get(i).getPoints();
                }
        }
                aux.add(temp);
                }
                for(int i=0;i<aux.size();i++) {
                        System.out.println((i+1) + "    " + aux.get(i).nom + "    " +
aux.get(i).getPoints());
                }
                aux = null;
        }

}
```

- **Equipe.java**

```
package Sport;
import Personne.*;
import java.util.ArrayList;

public class Equipe {

        // Chaque Equipe est composee par un ensemble de joueurs titulaires, d'autres
remplaçants et Entraineur.

        public String nom;
            private ArrayList <Joueur> titulaires;
            private ArrayList <Joueur> remplacants;
            private Entraineur entraineur;
            private int victoires;
            private int defaites;
            private int nulles;
            private int points;


           public Equipe(String nom,Entraineur entraineur){
                    this.nom= nom;
                    this.entraineur=entraineur;
                    titulaires = new ArrayList <>();
                 remplacants = new ArrayList<>();
                victoires=0;
                defaites=0;
                nulles=0;
                points=0;

             }

             public void addTitulaire(Joueur J) {
                titulaires.add(J);
             }

             public void addRemplacant(Joueur J) {
              remplacants.add(J);
             }

             // Une équipe peut changer son entraineur
             public void setEntraineur(Entraineur E) {
                entraineur=E;
             }

             // Calculer le nombre de points de l'équipe
             public void setPoints() {
                points = 3*victoires + nulles;
             }
```

```
    //Un joueur peut changer d'équipe sera donc supprimer de son équipe actuel
    public void deleteJoueur(Joueur j) {
        for(int i=0;i<titulaires.size();i++) {
                if(titulaires.get(i).equals(j))
         titulaires.remove(j);
        }

    for(int i=0;i<remplacants.size();i++) {
        if(remplacants.get(i).equals(j))
 remplacants.remove(j);
        }
}

    // Afficher la liste de joueurs d'une équipe donnée
    public void Affiche() {
        System.out.println("********* " + nom + "*********");
        System.out.println("** Liste des Joueurs Titulaires **");
        System.out.println();
        for (int i=0;i<titulaires.size();i++) {
                System.out.println();
                titulaires.get(i).Afficher();
                System.out.println();
    }
        System.out.println("** Liste des Joueurs Remplacants **");
        System.out.println();
        for(int i=0;i<remplacants.size();i++) {
                System.out.println();
                remplacants.get(i).Afficher();
                System.out.println();
        }
    }


    public void afficherStatistiques() {
        System.out.println("    **** " + nom + " ****");
        System.out.println("Nombre de matchs gagnés = " + victoires);
        System.out.println("Nombre de matchs nulles = " + nulles);
        System.out.println("Nombre de matchs perdus  = " + defaites);
        System.out.println("Nombre de points = " + points);

    }


    public void incrementVictoires() {
        victoires++;
        points+=3;
    }
```

```
            public void incrementDefaites() {
                defaites++;
            }


            public void incrementNulles() {
                nulles++;
                points+=1;
            }

            // Vérifie l'existence d'une joueur à une équipe graçe à son nom pour eviter la
redondance de donnée
                public boolean existeJoueur(String name) {
                        for(int i=0;i<titulaires.size();i++) {
                       if(titulaires.get(i).getNom().equals(name))
                                return true;
                        }
                    for(int i=0;i<remplacants.size();i++) {
                        if(remplacants.get(i).getNom().equals(name))
                                return true;
                    }
                     return false;
                  }


                // Vérifie si un num de maillot appratient déjà à un joueur d'une équipe
donnée car deux joueurs ne peuvent pas porter le meme numéro
                 public boolean existeNumMaillot(int num) {
                   for(int i=0;i<titulaires.size();i++) {
                        if(titulaires.get(i).getNumMaillot() == num)
                                return true;
                        }
                   for(int i=0;i<remplacants.size();i++) {
                        if(remplacants.get(i).getNumMaillot() == num)
                                return true;
                        }
                    return false;
                   }

                // Necessaire pour le MAJ d'un joueur donnée et pour éviter la redondance lors
de saisie
         public Joueur getJoueurByName(String name) {
                   for(int i=0;i<titulaires.size();i++) {
                     if (titulaires.get(i).getNom().equals(name))
                            return titulaires.get(i);
                  }
                   for(int i=0;i<remplacants.size();i++) {
                    if (remplacants.get(i).getNom().equals(name))
                            return remplacants.get(i);
            }
                            return null;
    }
```

```
    public ArrayList<Joueur> getTitulaires() {
        return titulaires;
    }

    public ArrayList<Joueur> getRemplacants() {
        return remplacants;
    }


public  Entraineur getEntraineur() {
        return entraineur;
 }


public int getPoints() {
       return points;
}
```

- **Match.java**

```
package Sport;
import Personne.Arbitre;
import java.util.concurrent.ThreadLocalRandom;

public class Match {

    private Equipe equipe1 ;
    private Equipe equipe2 ;
    private int nombreButEquipe1;
    private int nombreButEquipe2;
    private Arbitre arbitre ;
    private String tour ; // "Aller" ou "Retour"

    // Constructeur
    public Match(Equipe equipe1,Equipe equipe2,Arbitre arbitre,String tour) {
    this.equipe1 = equipe1;
    this.equipe2 = equipe2;
    // Le nombre de buts de chaque équipes est calculé arbitrerement mais sans depasser 10
    nombreButEquipe1 = ThreadLocalRandom.current().nextInt(0, 10 + 1);
    nombreButEquipe2 = ThreadLocalRandom.current().nextInt(0, 10 + 1);
    this.arbitre = arbitre;
    this.tour = tour;
```

```
            if(nombreButEquipe1>nombreButEquipe2) {
                        equipe1.incrementVictoires();
                        equipe2.incrementDefaites();
                }
                else if (nombreButEquipe1<nombreButEquipe2) {
                    equipe1.incrementDefaites();
                        equipe2.incrementVictoires();
                        }
                else {
                        equipe1.incrementNulles();
                        equipe2.incrementNulles();
                }
            equipe1.setPoints();
            equipe2.setPoints();
    }


    // Constructeur surchargé si on veut saisir le nombre de buts par le clavier
    public Match(Equipe equipe1,Equipe equipe2,int e1b,int e2b, Arbitre arbitre,String tour)
{
        this.equipe1 = equipe1;
        this.equipe2 = equipe2;
        nombreButEquipe1 = e1b;
        nombreButEquipe2 = e2b;
        this.arbitre = arbitre;
        this.tour = tour;
          // mettre à jour les statistique ( victoire, defaite, match nulle et nombre total
des points) de chaque équipe
            if(nombreButEquipe1>nombreButEquipe2) {
                        equipe1.incrementVictoires();
                        equipe2.incrementDefaites();
                }
                else if (nombreButEquipe1<nombreButEquipe2) {
                    equipe1.incrementDefaites();
                        equipe2.incrementVictoires();
                        }
                else {
                        equipe1.incrementNulles();
                        equipe2.incrementNulles();
                }
            equipe1.setPoints();
            equipe2.setPoints();
        }

    // Necessaire pour afficher l'historique des matches ( cad tous les details du match)
    public void afficheMatch() {
        System.out.println();
        System.out.println("Tour :" + tour);
        System.out.println(equipe1.nom +" " + nombreButEquipe1 + "-"+ nombreButEquipe2 + " "+
equipe2.nom);
        System.out.println("Arbitre :" + arbitre.getNom());
        System.out.println();
        }
```

```
  public void afficheResultat() {
        System.out.println("Tour :" + tour + " : " + equipe1.nom +" " + nombreButEquipe1 +
"-"+ nombreButEquipe2 + " "+ equipe2.nom);
    }


   public Equipe getEquipe1(){
           return(equipe1);
   }


   public Equipe getEquipe2(){
           return(equipe2);
   }


   public String getTour() {
           return tour;
   }


}
```