

M3103 - ALGORITHMIQUE AVANCEE

JAVA - TP N°7

Nous souhaitons élaborer et manipuler un document XML permettant de suivre un tournoi sportif, et d'obtenir le palmarès de ce tournoi.

Pour cela, nous nous appuierons sur les classes Joueur et Equipe que nous avons déjà conçues dans un exercice précédent.

Notre travail consistera à développer une nouvelle classe Tournoi ; puis de compléter la trame d'une autre classe permettant de créer le fichier XML voulu.

1 Rappels

- Une équipe est désignée par un nom, et est composée d'un nombre défini de joueurs.
- Les joueurs des différentes équipes sont caractérisés par un nom, un prénom, et un numéro qu'ils portent sur leurs maillots. Un joueur joue à un poste (pour simplifier, nous supposons que les joueurs ne jouent qu'à un seul poste).
Un joueur ne joue que pour une seule équipe.
Enfin, un joueur est soit titulaire soit remplaçant.

Pour gagner du temps, vous pouvez utiliser les classes que vous avez déjà codées, ou utiliser le code fourni en annexe.

2 Gestion de tournoi

Chaque année a lieu un tournoi pour lequel des équipes concourent.

Nous souhaitons obtenir un fichier XML contenant :

- Les équipes de l'édition courante du tournoi
- Les vainqueurs des éditions passées du tournoi

Ci-dessous, un exemple de fichier voulu :

```
<?xml version="1.0" encoding="UTF-8"?>
<tournoi>
  <nom>Superbowl</nom>
  <equipes>
    <equipe>
      <nom>Cardinals</nom>
      <joueurs>
        <joueur titulaire="true">
          <nom>Lee</nom>
          <prenom>Andy</prenom>
          <poste>P</poste>
          <numero>2</numero>
        </joueur>
        ...
      </joueurs>
    </equipe>
    ...
  </equipes>
  <palmares>
    <vainqueur annee="2018">Eagles</vainqueur>
    <vainqueur annee="2017">Patriots</vainqueur>
  </palmares>
</tournoi>
```

2.1 Classe Tournoi

Vous devez concevoir une classe Tournoi qui contiendra les données à placer dans le document XML. A l'aide d'un diagramme de classe UML que vous concevrez, et de l'extrait de document XML ci-dessus, vous déduirez les attributs de cette classe.

Indice : idéalement, vous serez amenés à gérer, entre autres, un attribut de type List et un autre attribut de type Map

2.2 Classe GestionDocumentXml

Maintenant que nous avons les 3 classes Joueur, Equipe et Tournoi qui permettent de gérer les données du tournoi ; nous allons créer une classe GestionDocumentXml qui pourra placer ces données dans un document au format XML.

Pour ce faire, vous allez :

- Compléter votre diagramme UML pour y faire figurer la classe GestionDocumentXml ; et visualiser ainsi les quatre classes qui seront utilisées depuis la classe Main et sa propre méthode main()
- Puis, compléter la trame de la classe GestionDocumentXml donnée ci-dessous :

```
import java.io.FileOutputStream;
import java.io.IOException;
import java.util.Iterator;

import org.jdom2.Attribute;
import org.jdom2.Document;
import org.jdom2.Element;
import org.jdom2.output.Format;
import org.jdom2.output.XMLOutputter;

public class GestionDocumentXml {
    //creation de constantes pour gérer les noms des balises
    private final String NOM_BALISE_RACINE = "tournoi";
    private final String NOM_BALISE_JOUEUR = "joueur";
    private final String NOM_ATTRIBUT_TITULAIRE = "titulaire";
    private final String NOM_BALISE_NOM = "nom";
    private final String NOM_BALISE_PRENOM = "prenom";
    private final String NOM_BALISE_NUMERO = "numero";
    private final String NOM_BALISE_POSTE = "poste";
    //...

    //attributs : le document et la racine
    private Document document;
    private Element racine;

    public GestionDocumentXml() {
        //Le constructeur permet d'initialiser les attributs
        racine = new Element(NOM_BALISE_RACINE);
        document = new Document(racine);
    }

    private void ajouteJoueur(Joueur joueur, Element parentJoueur){
        //Permet d'ajouter un objet Joueur au document XML dans la liste
        //des joueurs d'une equipe
        Element elementJoueur = new Element(NOM_BALISE_JOUEUR);
        parentJoueur.addContent(elementJoueur);

        Attribute attributeClasse = new
        Attribute(NOM_ATTRIBUT_TITULAIRE, Boolean.toString(joueur.isTitulaire()));
        elementJoueur.setAttribute(attributeClasse);

        Element elementNom = new Element(NOM_BALISE_NOM);
        elementNom.setText(joueur.getNom());
        elementJoueur.addContent(elementNom);

        Element elementPrenom = new Element(NOM_BALISE_PRENOM);
        elementPrenom.setText(joueur.getPrenom());
```

```

        elementJoueur.addContent(elementPrenom);

        Element elementPoste = new Element(NOM_BALISE_POSTE);
        elementPoste.setText(joueur.getPoste());
        elementJoueur.addContent(elementPoste);

        Element elementNumero = new Element(NOM_BALISE_NUMERO);
        elementNumero.setText(String.valueOf(joueur.getNumero()));
        elementJoueur.addContent(elementNumero);
    }

    private void ajouteEquipe(Equipe equipe, Element parentEquipe){
        //Permet d'ajouter un objet Equipe au document XML dans la liste
        des equipes
        //TODO: A compléter...
    }

    public void ajouteTournoi(Tournoi tournoi){
        //Permet d'ajouter les attributs d'un tournoi au document XML
        //TODO: ajouter les Equipes
        //TODO: ajouter le Palmares
    }

    public void affiche(){
        try{
            XMLOutputter sortie = new
XMLOutputter(Format.getPrettyFormat());
            sortie.output(document, System.out);
        }
        catch (IOException ioe){
            ioe.printStackTrace();
        }
    }

    public void enregistre(String nomFichier){
        try{
            XMLOutputter sortie = new
XMLOutputter(Format.getPrettyFormat());
            sortie.output(document, new
FileOutputStream(nomFichier));
        }
        catch (IOException ioe){
            ioe.printStackTrace();
        }
    }

    public Document getDocument() {
        return document;
    }

    public void setDocument(Document document) {
        this.document = document;
    }

    public Element getRacine() {
        return racine;
    }

    public void setRacine(Element racine) {
        this.racine = racine;
    }
}

```

3 Programme de tests

Le programme ci-dessous va créer des joueurs, des équipes et les inscrire au tournoi. Le palmarès du tournoi est complété au fur et à mesure du code de création du tournoi.

```
public class Main {

    public static void main(String[] args) {
        Tournoi tournoi = new Tournoi("Superbowl");

        Equipe equipe = new Equipe("Cardinals");
        tournoi.ajouteEquipe(equipe);
        Joueur joueur1 = new Joueur("Lee", "Andy", 2, "P", true);
        Joueur joueur2 = new Joueur("Fitzgerald", "Larry", 11, "WR",
true);

        Joueur joueur3 = new Joueur("Ford", "Rudy", 30, "DB", false);
        equipe.ajouterJoueur(joueur1);
        equipe.ajouterJoueur(joueur2);
        equipe.ajouterJoueur(joueur3);

        equipe = new Equipe("Eagles");
        tournoi.ajouteEquipe(equipe);
        joueur1 = new Joueur("Foles", "Nick", 9, "QB", false);
        joueur2 = new Joueur("Agholor", "Nelson", 13, "WR", true);
        joueur3 = new Joueur("Jenkins", "Malcolm", 27, "SS", true);
        equipe.ajouterJoueur(joueur1);
        equipe.ajouterJoueur(joueur2);
        equipe.ajouterJoueur(joueur3);

        tournoi.inscrirePalmares("2018", equipe);

        equipe = new Equipe("Patriots");
        tournoi.ajouteEquipe(equipe);
        joueur1 = new Joueur("Brady", "Tom", 12, "QB", true);
        joueur2 = new Joueur("Hogan", "Chris", 15, "WR", true);
        joueur3 = new Joueur("Gostkowski", "Stephen", 3, "K", true);
        equipe.ajouterJoueur(joueur1);
        equipe.ajouterJoueur(joueur2);
        equipe.ajouterJoueur(joueur3);

        tournoi.inscrirePalmares("2017", equipe);

        GestionDocumentXml documentXml = new GestionDocumentXml();
        documentXml.ajouteTournoi(tournoi);
        documentXml.affiche();

    }

}
```

Saisissez ce programme qui vous permettra de vérifier le bon fonctionnement de vos classes. En outre, ce programme vous permettra d'obtenir du contenu XML pour la question suivante.

4 Recherche d'informations

Modifiez la classe de gestion du document XML pour proposer :

- Une méthode qui va parcourir le document XML, afin de retourner le nom du tournoi
- Une méthode qui va parcourir le document XML, afin de retourner la liste des noms des équipes participant au tournoi
- Une méthode qui va parcourir le document XML, afin de retourner le nom du vainqueur du tournoi d'une année passée en paramètre
- Une méthode qui va parcourir le document XML, afin de retourner le poste du joueur dont le numéro et le nom de l'équipe sont passés en paramètres

Vérifiez la validité de vos méthodes en les appelant depuis votre programme de tests.

5 Annexes

5.1 Joueur

```
public class Joueur {
    private String nom;
    private String prenom;
    private int numero;
    private String poste;
    private boolean titulaire;

    public Joueur() {
        super();
        // TODO Auto-generated constructor stub
    }

    public Joueur(String nom, String prenom, int numero, String poste,
boolean titulaire) {
        super();
        this.nom = nom;
        this.prenom = prenom;
        this.numero = numero;
        this.poste = poste;
        this.titulaire = titulaire;
    }

    public String getNom() {
        return nom;
    }

    public void setNom(String nom) {
        this.nom = nom;
    }

    public String getPrenom() {
        return prenom;
    }

    public void setPrenom(String prenom) {
        this.prenom = prenom;
    }

    public int getNumero() {
        return numero;
    }

    public void setNumero(int numero) {
        this.numero = numero;
    }

    public String getPoste() {
        return poste;
    }

    public void setPoste(String poste) {
        this.poste = poste;
    }

    public boolean isTitulaire() {
        return titulaire;
    }

    public void setTitulaire(boolean titulaire) {
        this.titulaire = titulaire;
    }

    public void afficher() {
        System.out.println(" " + numero + " " + nom + " " + prenom + "
" + poste +
                                (isTitulaire() ? " Titulaire" : " Remplaçant")
                                );
    }
}
```

5.2 Equipe

```
import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;

public class Equipe {
    private static final int NB_MAX_JOUEURS = 15;

    private String nom;
    private List<Joueur> joueurs;

    public Equipe(String nom) {
        this.nom = nom;
        joueurs = new ArrayList<Joueur>();
    }

    public String getNom() {
        return nom;
    }

    public void setNom(String nom) {
        this.nom = nom;
    }

    public List<Joueur> getJoueurs() {
        return joueurs;
    }

    public void setJoueurs(List<Joueur> joueurs) {
        this.joueurs = joueurs;
    }

    public void ajouterJoueur(Joueur joueur) {
        if (joueurs.size() < NB_MAX_JOUEURS && joueur != null)
            joueurs.add(joueur);
    }

    public List<Joueur> getTitulaires() {
        List<Joueur> titulaires = new ArrayList<Joueur>();

        for (Joueur joueur : joueurs)
            if (joueur.isTitulaire())
                titulaires.add(joueur);

        return titulaires;
    }

    public List<Joueur> getRemplacants() {
        List<Joueur> remplacants = new ArrayList<Joueur>();

        for (Joueur joueur : joueurs)
            if ( ! joueur.isTitulaire() )
                remplacants.add(joueur);

        return remplacants;
    }

    public List<Joueur> getTypeJoueur(boolean titulaire) {
        if (titulaire)
            return getTitulaires();
        else
            return getRemplacants();
    }

    public void enleverJoueur(int numero) {
        Iterator<Joueur> iterateurJoueur = joueurs.iterator();

        while (iterateurJoueur.hasNext()) {
            Joueur joueur = iterateurJoueur.next();
            if (joueur.getNumero() == numero) {
                iterateurJoueur.remove();
                break;
            }
        }
    }
}
```

```
public void afficher() {  
    System.out.println(nom);  
    for (Joueur joueur : joueurs)  
        joueur.afficher();  
}
```