

Table des Matières

[Implémentation des classes et la description du programme](#)

[Classification](#)

[Implémentation](#)

[Diagramme](#)

Implémentation des classes et la description du programme

Classification

Lors de la classification de notre programme, nous avons opté pour différente stratégie d'implémentation pour chaque classe. Dans ce document, vous y retrouverez la description des classes suivante: personne et ses héritages, les classes filles de personne, le club, le calendrier, les rencontres, les matchs, les résultats et certaines structures de données.

Notre premier grand défi à été d'implémenter la classe personne. Nous avons décidés d'utiliser "Person" comme super-classe. Cette classe décrit toute les sortes de personne disponible dans notre application. Cette classe comprend les paramètres suivant: un nom, un prénom, un âge et un rôle. Comme par exemple, la variable "staffTechnique" est un ensemble de personne sous forme de vecteur. De plus, cette classe est représentée par 2 types de classe, "Joueur" ou "Entraîneur". Ces deux classes héritent de tout les variables et fonctions de "Personne". Par la suite, la classe "Joueur" donne naissance à deux autres sous-classes, "Joueur-Autonomie" et "Joueur non-autonomie". La seule différence entre les deux classes est que, "Joueur-autonomie" peut rompre un contrat avec sont équipe actuel. Contrairement à un "Joueur non-autonomie", celui-ci doit attendre que sont contrat devienne à terme ou obtenir un avis favorable pour rompre sont contrat avec son équipe actuel.

Nous avons un deuxième cas particulier d'héritage. Nous avons la super-classe "Palmares" qui donne tout ses fonctions et méthodes à la classe "TitreGagné".

Nous avons décidé d'utiliser différente structure de donnée, par exemple des *“Struct”* et des *“Enum”*. Dans le cas des *“enum”*, nous avons déterminé les différents rôles possible qu'un *“staffTechnique”* (Personne) peut prendre. Il y a deux autres *“enum”* que nous avons déclaré, *“Titre”* et *“Raison”* (raison de départ). Une structure de donnée de type *“Date”* a été implémenté pour nous faciliter la tâche. Nous lui avons ajouté des caractéristiques et certaines méthodes que nous avons utilisé tout au long du travail.

Le reste des classes, nous les avons traitées comme des entités individuel. C'est-à-dire, que chaque entités représentent qu'une seule classe. Il y a certaine relation entre ses classes là. Vous aller obtenir un diagramme visuelle pour mieux voir ces relations.

Implémentation

Cette section du travail à été assez simple à gérer, puisque l'implémentation des classes nous venaient naturellement. Chaque classe possède des *get* et *set* de leur propre attributs. En plus des *set* et *get*, certaines classes ont des méthodes pour afficher des éléments important de leur classe. La logistique principal de l'application ce retrouve dans la classe *“Ligue”*. Cette classe gère les calendriers, les rencontres et les match entre 2 équipes. Pour l'intégration des effectifs, staff, ruptures et contrat la logique ce retrouve dans la classe *“Club”*.

Diagramme

