

Sommaire

- | | | |
|-------------|---|--------------|
| <i>I.</i> | <u><i>Introduction.</i></u> | p.2 |
| <i>II.</i> | <u><i>Analyse fonctionnelle générale.</i></u> | p.2-3 |
| <i>III.</i> | <u><i>Analyse fonctionnelle détaillée</i></u> | p.3-4 |
| <i>IV.</i> | <u><i>Conclusion</i></u> | p.4 |

I.Introduction

Notre projet a pour fonction d'assurer la gestion d'un club de tennis, en analysant les éléments qui constituent le club que ce soit les membres, les compétitions, les évènements sportifs... Par exemple, si le club veut réaliser une compétition, notre programme sera capable d'ajouter une compétition en y incluant les membres....

II.Analyse fonctionnelle générale

Tout d'abord, pour pouvoir mener à mieux notre projet, nous avons analysé le cahier des charges en utilisant un code de couleurs pour identifier dans un premier temps les classes, les attributs pour chaque classe ainsi que les éléments qui nous conduisaient à la création de méthodes.

Ensuite, nous avons réalisé au brouillon un diagramme de classes provisoire sur lequel un premier défi est apparu concernant le nombre de classes est surtout les associations entre elles.

Plusieurs classes ont été prévu telles que club, personne, membre, personnel, entraineur, salarié, évènements, equipe simple, equipe double....

Puis, on avait envisagé de créer une classe joueur, mais suite à une longue hésitation et après avoir analysé d'avantage le cahier des charges, ce dernier stipule que les

membres jouent soit au titre de loisir, soit au titre de la compétition. Par conséquent, nous avons choisi de considérer tous les membres comme des joueurs.

En outre, une grande difficulté était de créer des équipes (simples ou doubles) à partir des membres et qu'ensuite pour chaque compétition, on affiche l'ensemble des équipes qui y participent.

On avait hésité à créer une classe équipe mais pour pouvoir affiner notre analyse, nous avons distingué deux classes: EquipeSimple et EquipeDouble.

III. Analyse fonctionnelle détaillée

Concernant la classe membre, nous avons eu l'idée de créer une méthode qui calcule les cotisations d'un membre non seulement en fonction de son âge mais aussi par rapport à sa provenance vis-à-vis du club. Nous avons créé une méthode qui calcule en plus la gestion et le coût de chaque membre en fonction des compétitions choisies par le membre et leurs prix respectifs.

De surcroît, pour pouvoir ajouter un nouveau membre, nous avons créé un programme où l'administration entre dans la console tous les attributs et le programme retourne un membre qui est désormais inscrit.

De plus, dans AjoutCompétition, on a créé deux listes; une équipe simple et une équipe double; où le club choisit le nombre d'équipe et grâce à une boucle itérative, on insère

les équipes qui participent à la compétition au fur et à mesure. Les deux derniers attributs *ListeEquipeSimple* et *ListeEquipeDouble* nous ont permis de mener une réflexion dans la manière d'insérer une équipe (une liste de membres), dans une compétition (une liste d'équipes). Le problème a été résolu avec succès grâce aux classes *EquipeSimple* et *EquipeDouble* où on a mis en attributs les membres ainsi que les nombres de victoire.

IV. Conclusion

En définitive, grâce à une analyse structurée du cahier des charges, nous avons pu créer différentes classes qui décrivent pertinemment les problématiques du club que ce soit au niveau de la gestion des membres, des entraîneurs, des équipes, des compétitions...

Le diagramme de classes a joué un rôle indispensable dans notre organisation des informations stockées, certaines méthodes ont été pour nous assez complexe que ce soit la création des équipes; qu'elles soient simples ou doubles; ou bien dans l'insertion d'une compétition.