

## Projects 2 :

### Finalité du sujet :

On désire créer un programme permettant de gérer l'organisation du Tournoi de Tennis du “Grand Chelem”.

Le championnat du “Grand Chelem” est constitué de 4 tournois majeurs d'une durée de deux semaines :

- **l'Open d'Australie** : le premier des quatre, se déroule à Melbourne pendant la période de Janvier – Février. Surface : Plexicushion
- **Roland Garros** : Se déroule à Paris lors de la dernière semaine de Mai et la première semaine de Juin. Surface : Terre Battue.
- **Wimbledon** : Commence la dernière semaine de Juin à Londres. Surface : Gazon.
- **l'US Open** : Se déroule au mois d'août à New York. Surface : Decoturf.

Chacun de ces tournois se déroule en 7 tours (1<sup>er</sup> tour, 2<sup>nd</sup>, 3<sup>ème</sup>, huitième de finale, quart, demi et Finale), il y a donc 128 joueurs qualifiés dans chaque catégorie.

Pour Simplifier nous ne considérerons que les catégories des matchs Simples (Hommes et Femmes). La seule différence se situe dans le fait qu'il est nécessaire de remporter 3 sets pour les hommes pour remporter le match (au lieu de 2 pour les femmes).

Également, puisque nous nous trouvons dans une durée limitée, il est demandé de pouvoir gérer un unique match entre 2 joueurs. Cependant les règles de marquage et d'annonce des fautes doivent être respectées.

Le sujet peut paraître long, il peut être préférable de bien le lire et d'avoir un aperçu de la manière dont vous allez aborder le(s) problèmes sous-jacents

Tout d'abord nous allons gérer les différents protagonistes, lors d'un match de tennis nous pouvons rencontrer des Joueurs, des Arbitres ainsi que des Spectateurs. Tous sont des Personnes et ont des caractéristiques communes que nous détaillons brièvement ci-après :

### Indications :

Ce projet va se poursuivre jusqu'à la fin des séances de TP, il s'agit d'un TP en escalier c'est à dire que chaque élément de chaque séance peut être et sera surement nécessaire au bon déroulement du TP de la semaine suivante. Différents consignes quant au contenu du code vous seront données au fur et à mesure. Veuillez donc à bien penser avant de coder, de faire les vérifications nécessaires, de faire en sorte que le programme ne « plante » pas des qu'un élément imprévu survient.

Veuillez à avoir votre méthode main dans une classe à part et distincte afin que la modification de celui-ci soit simple notamment pour la correction.

Le contenu des TP à venir est, il me semble, assez conséquent. Essayez par conséquent de ne pas prendre trop de retard en vous disant que vous le ferez la semaine prochaine. Si toutefois vous avez du temps libre à la fin d'une séance vous pouvez me le signaler afin que je vous dise quelle préparation vous pouvez faire pour la semaine suivante.

## 1) Création d'une personne

Afin de pouvoir organiser tout cela, commençons par une Personne. Celle-ci possède de nombreuses caractéristiques :

Un nom de naissance :	nomNaissance
Un nom courant :	nomCourant
Un prénom :	prenom
Un surnom :	surnom
Une date de naissance :	dateNaissance
Un lieu de naissance :	lieuNaissance
Une date de décès :	dateDeces
Une nationalité :	nationalite
Une taille (en cm) :	taille
Un poids (en Kg) :	poids

Une fois les données nomNaissance, genre, dateNaissance, lieuNaissance, dateDeces définis ils ne doivent pas pouvoir être modifiés.

Le nomCourant n'est défini que si la personne est mariée et qu'il s'agit d'une femme (dans ce cas la le nom courant est identique à celui du mari).

- a) Créer cette classe ainsi que les attributs associés, vous déterminerez par conséquent les types de données ainsi que les visibilitées les plus adaptés.
- b) Déterminer le(s) constructeur(s) nécessaire.
- c) Lorsque possible et/ou utile, créer les getters et setters de cette classe.
- d) On doit pouvoir connaître l'âge de la Personne en vie ou décédée.

## 2) Création d'un joueur

Un joueur est une personne dont on connaît quelques caractéristiques supplémentaires :

Une main de jeu (Droitier ou Gaucher) :	main
Un sponsor :	sponsor
Un classement :	classement
Un entraîneur :	entraîneur

- a) Créer cette classe ainsi que les attributs associés, vous déterminerez par conséquent les types de données ainsi que les visibilitées les plus adaptés.
- b) Déterminer le(s) constructeur(s) nécessaire.
- c) Lorsque possible et/ou utile, créer les getters et setters de cette classe.

On distinguera les Joueurs hommes des joueurs femmes par le fait que les femmes portent une jupe et les hommes un short, ces deux éléments sont caractérisés par la couleur et lorsque celle-ci change les joueurs s'expriment pour notifier à l'assemblée ce changement.

**note :** Lors de l'initialisation le classement du Joueur est fixé par l'ordre de création, ensuite il sera modifié en fonction des victoires/défaites de chacun.

### **3) Les arbitres**

Lors d'une rencontre hormis les joueurs il y a également plusieurs arbitres sur le terrain. Il y a des arbitres de fond de cours et un arbitre central qui dirige le déroulement du match. Nous considérerons uniquement l'arbitre central.

Un arbitre est une personne qui peut effectuer certaines actions, il dispose d'un micro lui permettant d'être entendu par les joueurs lorsqu'il annonce le score du Jeu en cours, les fautes, ...

Occasionnellement il est appelé par un joueur en cas de litige sur l'arbitrage, la décision de l'arbitre central est celle retenue.

a) Créer cette classe ainsi que les attributs associés, vous déterminerez par conséquent les types de données ainsi que les visibilité les plus adaptés.

b) Déterminer le(s) constructeur(s) nécessaire.

c) Lorsque possible et/ou utile, créer les méthodes de cette classe.

### **4) Création d'un spectateur**

Un spectateur est une personne qui a acheté un billet pour être dans les gradins d'un match (le prix du billet varie selon l'avancement du tournoi).

Il a une place numérotée dans les gradins d'un court représenté par la tribune (soit un nom soit une lettre ou double lettre), ainsi qu'un numéro de place.


Un spectateur peut applaudir, crier, huer ou dormir (mais c'est rare) en fonction du déroulement du match.

Les spectateurs hommes se distinguent par la couleur de leur chemise qu'ils mettent en évidence lorsqu'ils en changent. Les femmes quant à elles sont reconnues par les lunettes portées. (non non le chapeau n'est pas un élément distinctif).

a) Créer cette classe ainsi que les attributs associés, vous déterminerez par conséquent les types de données ainsi que les visibilité les plus adaptés.

b) Déterminer le(s) constructeur(s) nécessaire.

c) Lorsque possible et/ou utile, créer les méthodes de cette classe.

<b>Level</b>	M1	
<b>Course</b>	Java Fundamentals	
<b>Session</b>	5	
<b>Subject</b>	Defining and using classes in Java	
<b>Duration</b>	3 hours 30 minutes	

## **Projects Tennis part 2 :**

### **1) Création d'un match**

Un match de Tennis peut avoir lieu à partir du moment où il y a 2 joueurs et 1 arbitre au minimum. Il y a obligatoirement l'un des joueurs qui sera vainqueur tandis que l'autre sera perdant.

L'organisation doit également permettre de connaître facilement la catégorie du match (simple, double, homme, femme ou mixte), ainsi que son niveau ( $x^{\text{ème}}$  tour ou  $1/x^{\text{ème}}$  de finale, voire amical).

Un match alloue un certain nombre de points à chaque joueur (classement ATP ou WTA).

La mise en jeu lors du premier jeu de service sera déterminée aléatoirement.

Comme dit précédemment, un match se termine en fonction du nombre de sets gagnés par un joueur (2 ou 3 en fonction de différents critères).

- a) Créer cette classe ainsi que les attributs associés, vous déterminerez par conséquent les attributs, les types de données ainsi que les visibilité les plus adaptés.
- b) Déterminer le(s) constructeur(s) nécessaire.
- c) Lorsque possible et/ou utile, créer les getters et setters de cette classe.
- d) Créez une méthode permettant de démarrer le match et de gérer le déroulement de celui-ci.

### **2) Création d'un Set**

Un set est remporté lorsqu'un certain nombre de Jeu ont été remporté par l'un des joueurs et sous conditions d'un écart d'au moins 2 jeu (sauf le cas particulier du Set décisif voir plus loin).

- a) Créer cette classe ainsi que les attributs associés, vous déterminerez par conséquent les attributs, les types de données ainsi que les visibilité les plus adaptés.
- b) Déterminer le(s) constructeur(s) nécessaire.
- c) Lorsque possible et/ou utile, créer les getters et setters de cette classe.
- d) Créez une méthode permettant de jouer (par conséquent le(s) Jeu(x)).

### **3) Création d'un Jeu**

Un jeu est servi intégralement par l'un des 2 Joueurs (sauf pour le jeu décisif).

Pour remporter un Jeu, l'un des joueurs doit remporter suffisamment de points, la notation des points se fait dans l'ordre suivant :

0 pt, 15 pts, 30 pts, 40 pts, AV.

Un joueur ayant 40 points qui gagne le service suivant gagne le jeu. En cas d'égalité (40 – 40) on parle alors d'Avantage.

Si un joueur à l'avantage et qu'il remporte le service, il gagne le jeu. Dans le cas contraire les

joueurs reviennent à égalité 40 – 40.

- a) Créer cette classe ainsi que les attributs associés, vous déterminerez par conséquent les attributs, les types de données ainsi que les visibilitées les plus adaptés.
- b) Déterminer le(s) constructeur(s) nécessaire.
- c) Lorsque possible et/ou utile, créer les getters et setters de cette classe.
- d) Créez une méthode permettant de gérer les différentes balles de services (les différents Echanges) ainsi que les scores.

#### **4) Création d'un Echange**

Au cours d'un match, les balles de Services sont le plus bas niveaux que nous allons prendre en compte. Un échange se fait entre 2 Joueurs (dans le cas d'un match simple), lors de la mise en service différents cas peuvent se présenter :

- Le service est faute (dans le filet ou en dehors de la zone où elle est supposée rebondir). Dans ce cas le Joueur qui sert a le droit à un second service. Si la faute se représente alors le point est gagné par l'adversaire.
- La balle touche le filet mais passe de l'autre côté du terrain, dans ce cas il n'y a pas faute et le Joueur recommence sa première balle.
- La balle arrive correctement dans la zone désirée.

- a) Créer cette classe ainsi que les attributs associés, vous déterminerez par conséquent les attributs, les types de données ainsi que les visibilitées les plus adaptés.
- b) Déterminer le(s) constructeur(s) nécessaire.
- c) Lorsque possible et/ou utile, créer les getters et setters de cette classe.
- d) Créez une méthode permettant pour chaque échange de saisir au clavier les différents événements possible (faute, filet, correct, ...) afin de pouvoir avancer dans l'échange (et par conséquent dans le Jeu puis dans le Set, puis dans le Match.

#### **5) L'arbitre**

Lors du TP précédant vous avez créé une classe Arbitre, maintenant que nous pouvons mettre en place un match entre deux Joueurs, l'arbitre prends tout son sens :

- a) A chaque fin d'un échange l'arbitre annonce le score du Jeu en cours en commençant par énoncer le score du Joueur qui sert.
- b) A chaque fin de jeu, l'arbitre annonce le nom du Joueur qui a remporté le Jeu ainsi que le score du Set.
- c) A chaque fin de Set, l'arbitre annonce le nom du joueur qui a remporté le Set ainsi que le score actuel du match.
- d) En cas de litige sur un points (de temps en temps) Un joueur appelle l'arbitre en évoquant le motif du litige (faute, balle hors du terrain de jeu, ...), ce dernier prends une décision et sera respectée. La décision se prendra par différent paramètre : un premier paramètre aléatoire en fonction de l'humeur du juge, un second paramètre en fonction de la réputation du Joueur appelant.

Modifiez votre classe Arbitre afin de prendre les 4 points énoncés ci-dessus en compte.

Modifiez votre classe Joueur afin que celui-ci puisse servir, retourner un service, renvoyer la balle de l'adversaire, faire des fautes, appeler l'arbitre, s'encourager, boire, crier sa victoire/sa défaite, ...

## **6) Règles (Source : wikipédia)**

Dans une rencontre de simple comme de double, chaque joueur sert à tour de rôle, la durée d'un tour de service est d'un jeu. Le joueur qui sert doit impérativement être placé derrière la ligne de service. Le serveur possède en outre deux services : s'il manque le premier, souvent frappé avec un certain risque, il bénéficie d'une seconde balle de service, en général frappée avec plus de prudence. Le receveur, en revanche peut se placer où il le souhaite sur le court pour retourner le service. Il lui est néanmoins interdit d'intercepter à la volée (sans rebond) le service de celui qui sert. Après que les deux personnes ont touché la balle, le point peut commencer.

La plupart du temps, il est nécessaire de remporter deux Sets pour gagner la partie. La seule exception est celle des matchs du tableau masculin des tournois du Grand Chelem, qui se jouent en trois sets gagnant. Pour gagner une manche (un Set), il faut être le premier à marquer six jeux avec au moins deux jeux d'écart, dans le cas contraire la manche se poursuit. Les scores possibles pour remporter un Set sont ainsi : 6/0, 6/1, 6/2, 6/3, 6/4, 7/5 (Si les joueurs n'ont pas pu se départager au bout de dix jeux). Si les deux joueurs n'ont pas été en mesure de se départager au bout des douze premiers jeux (donc à égalité à 6/6), ils disputent un jeu décisif, qui vaut un jeu, et permet donc de remporter la manche 7/6. En revanche, dans les tournois du Grand Chelem (sauf l'US Open), chez les hommes comme chez les femmes, il n'y a pas de jeu décisif dans la manche décisive (le 5ème chez les hommes et la 3ème chez les femmes) et, le match n'est remporté que lorsqu'on parvient à avoir deux jeux d'avance sur l'adversaire ; par exemple 8/6, 9/7, 10/8, etc.

L'invention du Jeu décisif date de 1970, afin d'empêcher des match interminables (29/27 par exemple). Le déroulement de ce jeu est assez simple, les joueurs servent à tour de rôle. Celui qui débute ne sert qu'une fois de droite à gauche, puis son adversaire sert deux fois de suite (de gauche à droite puis de droite à gauche, et ainsi de suite. Le gagnant de la manche est le premier joueur à atteindre sept points avec au moins deux points de différence (7/2, 7/5, 9/7, ...) La manche est alors gagnée sur le score de 7-6.

Une manche se remporte donc en marquant un certain nombre de jeux. Comme mentionné ci-dessus, chaque joueur sert à tour de rôle, pendant la durée d'un jeu. Afin de remporter le jeu, il est nécessaire de marquer au moins quatre points, soit sur son service lorsque que l'on sert, soit sur le service adverse lorsque l'on reçoit. Il est donc possible, soit pour le serveur, soit pour le receveur de remporter un jeu, même si théoriquement, le serveur est avantagé par rapport au receveur. Si les deux adversaires marquent trois points, on a une situation d'égalité, expliquée ci-après.

Lors d'un jeu, voici la manière dont les points sont décomptés :

zéro : pour aucun point marqué  
quinze : pour un point marqué  
trente : pour deux points marqué  
quarante : pour trois points marqué

Lorsque les deux joueurs ont marqué trois points (donc à 40/40) il y a égalité. Celui qui marque le point suivant obtient un « avantage ». Pour remporter le jeu, un joueur qui a l'avantage doit marquer un autre point. Si ce n'est pas le joueur qui a l'avantage qui marque le point suivant, on revient à égalité et ainsi de suite jusqu'à ce que l'un des deux joueurs remporte le jeu. Le joueur qui reçoit choisit sa zone de retour de service pour le point décisif (à 40/40).


Concernant l'arbitrage, on donne toujours le score du serveur en premier

### Les Fautes :

On distingue plusieurs types de fautes au tennis. Une balle sera par exemple annoncée faute lorsqu'elle ne retombe pas dans les limites du terrain. Le point est alors accordé à l'adversaire. Lorsque la balle tombe dans le filet, du côté du joueur qui a frappé la balle, le point est également accordé à l'adversaire, mais il n'est pas nécessaire d'annoncer faute. L'arbitre ne doit d'ailleurs pas signaler une balle qui tombe dans le filet comme une faute.

Le service doit être frappé en diagonale de telle sorte que la balle tombe dans le carré de service. Si la balle ne tombe pas dans le carré de service lors de la mise en jeu, l'arbitre annonce faute, et le serveur doit, soit servir une seconde balle si la faute survient sur le premier service, soit accorder le point à l'adversaire si la faute survient sur la seconde balle de service. Lorsque les deux services sont en faute (hors du carré de service ou dans le filet), on parle alors de « double faute ». Le point est alors accordé au receveur. Lorsqu'au service, la balle touche la bande du filet et retombe dans le carré de service où le joueur était sensé servir, l'arbitre annonce « let », la balle n'est pas faute, et le serveur peut rejouer le service. En revanche, si la balle du serveur touche la bande du filet et tombe en dehors du carré de service, la balle est annoncée comme faute, et le joueur doit soit passer à sa seconde balle, soit accorder le point à l'adversaire selon qu'il a frappé une première ou une seconde balle.

Une faute plus complexe concerne le serveur. En effet, pour effectuer un service valable, il est nécessaire que la balle soit frappée avant que le joueur ne franchisse la ligne de fond de court. Ainsi, lorsque le joueur frappe son service, et à déjà une partie de son corps qui touche le sol à l'intérieur du court ou bien lorsque le serveur « mord » la ligne de fond de court au moment de son lancer de balle, le service est refusé. Le joueur peut alors soit frapper une seconde balle de service si sa faute a été commise sur son premier service, soit accorder le point à l'adversaire si cette faute intervient sur sa deuxième balle. Cette faute est assez rarement signalée car difficile à juger, est nommée « faute de pied ». La « faute de pied » est également valable sur un plan latéral : le serveur doit se trouver du bon côté du terrain de telle sorte à servir dans une diagonale. Si le serveur sert à droite, il doit se tenir à gauche (et inversement) sans être dans le prolongement du couloir.

<b>Level</b>	M1	
<b>Course</b>	Java Fundamentals	
<b>Session</b>	5	
<b>Subject</b>	Defining and using classes in Java	
<b>Duration</b>	3 hours 30 minutes	

## **Projects Tennis part 3 :**

Avant d'entamer ce sujet, je vous rappelle que les exercices précédents doivent être fonctionnels, pour l'évaluation de ce projet chaque partie sera notée sur 10 (il y en aura 4 en tout) ce qui vous fera 5 notes et un total de 60 points, divisé par 3 vous aurez ainsi votre note de TP (50% de la note finale).

### **1) Petites Modifications**

Jusqu'à présent nous avons considéré chaque type de personnages distincts, seulement que fait un Joueur lorsqu'il n'est pas sur le terrain ?

Nous voudrions que celui-ci puisse aller voir d'autres Matches qui sont en train de se jouer, par conséquent un Joueur peut également être Spectateur.

Effectuez les modifications nécessaire pour rendre cela possible (Attention je vous rappelle qu'en Java l'héritage multiple est interdit).

### **2) Création d'un Tournoi**

L'objectif principal du programme étant de pouvoir gérer le déroulement d'un Tournoi de tennis du "Grand Chelem", il est désormais temps de passer à celui-ci.

Pour qu'un Tournoi puisse être mis en place il faut bien évidemment utiliser tous les éléments que vous avez créé précédemment (Joueurs, Arbitres, Spectateurs, Matches, ...).

Un Tournoi donné est principalement caractérisé par la ville dans laquelle celui-ci se déroule (Melbourne, Paris, Londres, New York) ainsi que par l'année. Les surfaces de terrains sont différentes pour chaque ville (Plexicushion, Terre Battue, Gazon, Decoturf).

Dans un tournoi, il y a différentes catégories (Simple Homme, Simple Femme, Double Homme, Double Femme, Junior, ...). Afin de ne pas trop vous surcharger nous ne considérerons que les catégories Simple. Dans chaque catégorie il y a 128 Joueurs qualifiés, il vous faut donc au moins 256 Joueurs. On considèrera qu'une dizaine d'Arbitres suffisent pour gérer tous les matchs et, pour que l'ambiance soit bonne il faut qu'il y ait un minimum de 100 spectateurs présent à chaque Match.

a) Créez une méthode permettant de créer automatiquement tous ces Joueurs, Arbitres, Spectateurs. Pour cela vous déterminerez la solution la plus adaptée pour vous afin que chaque protagoniste soit différent.

Nous considérerons que les 64 premières rencontres de chaque catégorie (1er tour) soient totalement aléatoire quant à l'ordre des Joueurs. Les 32 suivantes (2<sup>nd</sup> tour) seront déterminés par les vainqueurs du tour précédent et ainsi de suite (3<sup>ème</sup> tour, 1/8<sup>ème</sup> de finale, Quart de finale, Demi



finale puis finale).

- b) Créer cette classe ainsi que les attributs associés, vous déterminerez par conséquent les attributs, les types de données ainsi que les visibilitées les plus adaptés.
- c) Déterminer le(s) constructeur(s) nécessaire.
- d) Lorsque possible et/ou utile, créer les getters et setters de cette classe.

### 3) Déroulement des Matches

Afin de tester le bon déroulement du tournoi, je ne pourrais pas faire tourner manuellement les 127 Matches de chaque catégorie. Il serait alors intéressant de mettre en place un système permettant de choisir pour chaque Match, le mode de déroulement (manuel ou automatique).

Dans le cas où le Match se joue "tout seul" on doit pouvoir choisir d'afficher ou non le détail.

Dans tous les cas lors de saisie clavier de la part de l'utilisateur, votre programme ne doit jamais planter, l'exécution ne doit pas s'arrêter sauf sur demande de l'utilisateur. Les cas d'erreurs doivent être gérés et pris en compte par le biais d'exceptions, si par exemple des demandes de modifications de tel ou tel attribut ne peut pas être effectuée car il s'agirait d'une donnée non valide, une exception doit être levée signalant l'erreur et indiquant que les modifications **ne sont pas faites**.

### 4) Statistiques


Pour chaque Match et pour chaque Joueur, généralement de nombreuses statistiques sont conservées :

- Nombre de Set, de Jeu, d'échanges joués, nombre total de points remportés.
- Nombre d'aces effectués, nombre de premier services, de second services, de double fautes.
- Nombre de balles de break, nombre de balles de break remportées.
- Vitesse moyenne des balles de services (1er et 2<sup>nd</sup>).

Chaque Joueur possède également des Statistiques personnelles (sur l'année et sur sa carrière) :

- Le Nombre de Matches remportés
- Le nombre de Match joués
- Le nombre de Tournoi auxquels il a participé
- Le total de ses gains

- a) Créer cette classe ainsi que les attributs associés, vous déterminerez par conséquent les attributs, les types de données ainsi que les visibilitées les plus adaptés.
- b) Déterminer le(s) constructeur(s) nécessaire.
- c) Lorsque possible et/ou utile, créer les getters et setters de cette classe.
- d) Effectuez les ajouts de code nécessaire à la mise à jours des statistiques

<b>Level</b>	M1	
<b>Course</b>	Java Fundamentals	
<b>Session</b>	6	
<b>Subject</b>	Fin du projet	
<b>Duration</b>	3 hours 30 minutes	

## **Projects Tennis part 4 :**

Il s'agit de votre dernière séance de TP de Java 1, je sais bien que certains d'entre vous penseront comme moi : « quel dommage, on commence à peine à pouvoir faire des choses intéressantes avec cette application » alors que d'autres sont déjà certainement en train de penser « Oh non encore un truc à la ??? sur le tennis ».

Comme c'est le dernier TP il s'agit désormais de rendre votre programme fonctionnel pour un utilisateur qui aura acheté votre logiciel.

### **1) Vérifications**

a) Comme vous le savez, tout bon programme doit être documenté, j'ose espérer que vous avez bien consciencieusement pensé à faire une JavaDoc complète et détaillée sur chacune de vos classes et méthodes au fur et à mesure. Si tel n'est pas le cas cela va vous prendre beaucoup de temps maintenant.

b) Tout programme Java doit prendre en compte la gestion des cas d'erreurs au travers des exceptions, soit celles déjà intégrées nativement dans le langage, soit vos propres exceptions personnelles. Lorsqu'une exception est interceptée l'utilisateur doit en être averti et disposer de l'information nécessaire pour pouvoir corriger l'erreur. En aucun cas des données erronées doivent passer au travers des mailles de vos tests, par exemple si la modification d'un attribut n'est pas correcte celle-ci ne doit pas avoir lieu et l'utilisateur doit en être averti.

### **2) Interface utilisateur**

Idéalement votre programme devrait disposer d'une interface graphique, seulement ceci n'est pas au programme de Java 1 et, de plus vous ne disposez pas du temps requis pour cela. Par conséquent on se contentera d'une interface en mode console (ce que vous avez fait jusqu'à présent).

Cette interface devra proposer à l'utilisateur plusieurs menus, chacun redirigera certainement vers d'autres sous-menus.

- On doit être capable de créer des personnages (Joueurs, Arbitres, Public) soit personnalisé, soit de manière automatique.
- On doit également pouvoir créer un Tournoi et y inscrire certains Joueurs manuellement (n'oubliez pas qu'il y a plus de 128 Joueurs existant alors qu'une catégorie d'un tournoi limite à 128), et compléter automatiquement (128 c'est long à inscrire).
- On doit pouvoir obtenir les informations d'un Joueur donné (infos basiques, statistiques générale), à la fin d'un tournoi on doit pouvoir connaître les stats d'un Joueur pour chacun de ses matches, ...
- On peut quitter le programme à tout moment.

Lors d'un tournoi :

- On doit pouvoir connaître l'ensemble des matchs passés ou à venir
  - Pour les matchs passés on doit pouvoir obtenir le détail du match
  - Pour les matchs à venir on doit pouvoir lancer un match manuellement et saisir tout ce qu'il s'y passe (bien évidemment on ne pourra pas jouer la finale si la demi finale n'a pas eu lieu, etc)
  - On peut connaître le nombre de matchs restant, passés, et ainsi de suite sur tout ce qu'il vous semble bon de pouvoir connaître.
  - On doit être capable de faire tout ce qu'il était possible au départ (création de joueurs, récupérations des infos, ...) sans pour autant quitter le tournoi. Bien souvent de nombreux amateurs devant la télé se mettent au tennis si un français effectue de bonnes performances par exemples.

Lorsque le tournoi est terminé, on doit avoir la possibilité d'obtenir une synthèse de celui-ci (nombre de spectateurs moyen, nombre de balles utilisées, nombre de lunettes/chapeaux vendus, ...)

Soyez imaginatif, si vous pensez à d'autres éléments utiles à ajouter à votre code n'hésitez surtout pas.

### **3) Conclusion**

Comme à la fin de chaque projet, vous rédigerez un rapport. Celui-ci contiendra un descriptif global de votre programme (diagramme de classe, explications associées), ainsi que pour chaque élément (classe, méthode) une explication de votre code, si vous avez du effectuer tel ou tel choix pour le déroulement.

Si vous avez eu des idées supplémentaires par rapport à ce qu'il a été demandé dans les sujets indiquez-le.

Même s'il s'agit d'une matière informatique, je vous rappelle que tout rapport contient une introduction, une conclusion, des transitions entre les parties, une suite logique dans le déroulement de la lecture ainsi que des références si certaines informations ne sont pas de vous.