Programmation Objet en Java - Feuille de TD 3

Le but de ce TD est de simuler un **match de tennis** entre deux joueurs. Pour cela, nous aurons besoin de pouvoir compter les points. Un match se déroule en plusieurs *sets* et un set, se déroule en plusieurs *jeux*. Un match se joue en 2 sets gagnants : le premier joueur qui remporte deux sets gagne le match.

Compter le score d'un jeu :

- Les points d'un jeu se comptent ainsi : 0, 15, 30, 40.
- Pour remporter un jeu, il faut dépasser 40 et avoir au moins deux coups d'avance.
- Si les deux joueurs sont à 40, et que l'un d'eux a un coup d'avance, on indique A, à côté de son score pour signifier qu'il a l'avantage. Si celui qui a l'avantage marque, il remporte le jeu.

Compter le score d'un set :

- Les points d'un set se comptent normalement : 0,1,2,3,4,5,6,7,...
- Pour remporter un set, il faut dépasser 6 et avoir au moins deux jeux d'avance.

Exercice 1. Jeu

- On souhaite créer un objet **GameCounter** pour compter les points d'un jeu. Vaut-il mieux faire un record ou une classe ?
- Quels sont les champs nécessaires pour modéliser le score d'un jeu pour deux joueurs ? Y a-t-il une seule façon de faire ? Quel(s) mot(s)-clé doit-on utiliser lors de leur déclaration ?
- Écrire le code de **GameCounter** afin de pouvoir créer un nouveau compteur de score qui maintient le score de chaque joueur et l'information de qui a l'avantage.
- Si ce n'est pas déjà fait, faire en sorte de pouvoir afficher le score courant du jeu.
- Ajouter à **GameCounter** une méthode **score**, qui prend en paramètre le numéro du joueur (1 ou 2) et met à jour le score lorsque ce joueur marque un point. De plus, la méthode devra renvoyer **true** si le joueur remporte le jeu et **false** sinon.

Remarque : vous pouvez ajouter d'autres méthodes si vous le souhaitez. Quelle devra être leur visibilité ?

Exercice 2. Set

- On souhaite créer un objet **SetCounter** pour compter les points d'un set. Vaut-il mieux faire un record ou une classe ?
- Quels sont les champs nécessaires pour modéliser le score d'un set pour deux joueurs ? Y a-t-il une seule façon de faire ? Quel(s) mot(s)-clé doit-on utiliser lors de leur déclaration ?
- Écrire le code de **SetCounter** afin de pouvoir créer un nouveau compteur de score qui maintient le score de chaque joueur au cours d'un set (c'est à dire la score du jeu courant et le nombre de jeux déjà remportés par chaque joueur).
- Si ce n'est pas déjà fait, faire en sorte de pouvoir afficher le score courant du jeu (score du set et score du jeu courant. Ajouter également une méthode **setScore** permettant d'afficher uniquement le score du set.

- e) Ajouter à `SetCounter` une méthode `score`, qui prend en paramètre le numéro du joueur (1 ou 2) et met à jour le score lorsque ce joueur marque un point. De plus, la méthode devra renvoyer 1 si le joueur remporte un jeu lorsqu'il marque ce point, 2 si le joueur remporte un set lorsqu'il marque ce point et 0 sinon.

Remarque : est-il possible de déclarer le champs correspondant au score du jeu courant final ?

Exercice 3. Et Match !

- a) On souhaite enfin créer un objet `TennisMatch` pour simuler le déroulement d'un match avec deux joueur qui marque des points de façon aléatoire (vous avez un exemple complet d'affichage à la fin du TD). Vaut-il mieux faire un record ou une classe ?
- b) Quels sont les champs nécessaires pour modéliser un match (et donc suivre le score tout au long du match) ? Y a-t-il une seule façon de faire ?
- c) Écrire le code de `TennisMatch` afin de pouvoir créer un nouveau match pour deux joueurs dont on peut choisir les noms.
- d) Ajouter à `TennisMatch` une méthode `score` qui prend en paramètre le numéro du joueur (1 ou 2) et met à jour le score lorsque ce joueur marque un point. De plus, la méthode devra renvoyer afficher le nom du joueur si celui-ci remporte un jeu en marquant ce point.
- e) Ajouter une méthode `set` qui simule le déroulement complet d'un set en affichant le score à chaque fois qu'un joueur marque un point. Cette méthode devra également renvoyer le numéro du joueur qui a gagné le set.

Remarque : dans quelle classe doit-on écrire cette méthode ?

- f) Ajouter une méthode `match` qui simule le déroulement complet d'un match en jouant chaque set tour à tour et en affichant le nom du joueur qui remporte le set à chaque fois puis finalement le nom du vainqueur et le score final.

Exercice 4. Re-match

En fait, il serait possible de conserver le score de chaque jeu avec uniquement deux entiers. Ré-écrire tout votre code pour que l'on puisse jouer un match de cette façon.

```

[0-0] 0 -- 0 > Bob
[0-0] 0 -- 15 > Alice
[0-0] 15 -- 15 > Bob
[0-0] 15 -- 30 > Alice
[0-0] 30 -- 30 > Bob
[0-0] 30 -- 40 > Bob
-----> jeu Bob
[0-1] 0 -- 0 > Alice
[0-1] 15 -- 0 > Alice
[0-1] 30 -- 0 > Alice
[0-1] 40 -- 0 > Bob
[0-1] 40 -- 15 > Alice
-----> jeu Alice
[1-1] 0 -- 0 > Alice
[1-1] 15 -- 0 > Alice
[1-1] 30 -- 0 > Alice
[1-1] 40 -- 0 > Bob
[1-1] 40 -- 15 > Bob
[1-1] 40 -- 30 > Bob
[1-1] 40 -- 40 > Alice
[1-1] 40 A -- 40 > Bob
[1-1] 40 -- 40 > Bob
[1-1] 40 -- 40 A > Bob
-----> jeu Bob
...
[4-5] 0 -- 0 > Bob
[4-5] 0 -- 15 > Alice
[4-5] 15 -- 15 > Alice
[4-5] 30 -- 15 > Bob
[4-5] 30 -- 30 > Alice
[4-5] 40 -- 30 > Bob
[4-5] 40 -- 40 > Bob
[4-5] 40 -- 40 A > Alice
[4-5] 40 -- 40 > Bob
[4-5] 40 -- 40 A > Bob
-----> jeu et set Bob
...
-----> jeu et set Alice
...
[6-5] 0 -- 0 > Bob
[6-5] 0 -- 15 > Bob
[6-5] 0 -- 30 > Alice
[6-5] 15 -- 30 > Bob
[6-5] 15 -- 40 > Alice
[6-5] 30 -- 40 > Alice
[6-5] 40 -- 40 > Alice
[6-5] 40 A -- 40 > Bob
[6-5] 40 -- 40 > Alice
[6-5] 40 A -- 40 > Alice
-----> jeu et set Alice

Alice remporte le match
[4-6]
[6-1]
[7-5]

```