

**Travaux Dirigés n° 2 : Classes**

*Objectifs : Savoir construire une classe avec ses attributs et ses méthodes.  
Savoir manipuler des objets instances de classes. Être capable de décider la  
visibilité et l'aspect statique ou dynamique d'un attribut ou d'une méthode.*



THE KILLER

## 1 Introduction

Peut-être connaissez-vous déjà le jeu appelé « Killer », en voici le principe. Dans ce jeu, chaque joueur est un tueur et possède un contrat : il doit « tuer » (en général avec un pistolet à eau) un autre joueur dont le nom lui a été donné en début de partie. Quand il exécute son contrat, il récupère alors le contrat de sa victime et doit maintenant tuer cette nouvelle personne, et ainsi de suite... Le jeu s'arrête quand il n'y a plus qu'un seul joueur : le « KILLER ».

Le but de ce TP est de pouvoir programmer ce jeu en langage JAVA.

## 2 Premiers objets

Dans un premier temps nous allons modéliser et créer des joueurs sans leur affecter de contrat.

1. Écrire une classe Tueur modélisant un joueur avec ses nom et âge.
2. Créer un constructeur pour la classe Tueur qui prendra en argument les informations sur le joueur à créer.
3. Ajouter une méthode afficheJoueur à la classe Tueur permettant d'afficher les informations sur un joueur. Par exemple : Blondin, 31 ans.

4. Écrire, dans une nouvelle classe Principale, une méthode main qui doit créer deux objets instances de la classe Tueur (modélisant respectivement les joueurs Sentenza, 42 ans et Tuco, 35 ans) et en afficher les informations.
5. Que faut-il pour afficher uniquement le nom de Sentenza depuis la méthode main tout en respectant le principe d'encapsulation ?

### 3 Références

Chaque joueur a un contrat à exécuter, c'est à dire un autre joueur, nous allons modéliser ceci grâce à des références à d'autres objets joueurs.

1. Ajouter à la classe Tueur une nouvelle variable d'instance de nom contrat et de type Tueur modélisant l'autre tueur à exécuter.
2. Ajouter à la classe Tueur une nouvelle méthode pour affecter un contrat à un joueur, dont la signature est : `void setContrat(Tueur contrat)`. Cette méthode prend en argument le tueur que le joueur doit tuer.
3. Ajouter à la classe Tueur un nouveau constructeur qui prendra aussi comme argument le contrat de type Tueur.
4. Dans la méthode main de la classe Principale, utilisez la méthode setContrat pour donner des contrats aux deux premiers joueurs et le nouveau constructeur pour créer un troisième joueur. Il est indispensable que les contrats forment une boucle fermée.

### 4 Méthodes et variables de classe

1. Créer dans la classe Tueur une méthode `void afficheContrat(Tueur t)` qui prend en argument un Tueur et qui affiche le nom de son contrat, puis le nom du contrat de son contrat et ainsi de suite. Il faut s'arrêter au moment où on retombe sur le premier joueur. *Indice : le plus simple est d'utiliser une boucle while testant à chaque étape si le tueur dont on va imprimer le contrat est le premier qu'on a affiché, et de n'afficher son nom que si ce n'est pas le cas.*
2. Pourquoi la méthode afficheContrat peut (doit !) être définie comme *statique* ? Utiliser cette méthode depuis le main en l'appelant de deux manières différentes.
3. On souhaite garder en mémoire un entier nombreTueurs correspondant au nombre de joueurs déjà créés. Mettre en place ce nombre dans la classe Tueur, faire en sorte qu'il soit initialisé correctement et mis à jour à chaque fois qu'un tueur est créé.

### 5 Liste chaînée

Pour gérer un grand nombre de joueurs, nous allons utiliser le principe de la *liste chaînée*. La méthode main ne connaîtra que le premier élément  $t_1$  de la liste, qui lui, connaît le deuxième, qui connaît le troisième et ainsi de suite. Le dernier élément de la liste a pour contrat le premier élément  $t_1$ . Si on veut connaître l'ensemble des joueurs, il faut partir de  $t_1$  et parcourir la liste en passant d'un élément à l'autre et en s'arrêtant lorsqu'on est revenu à  $t_1$ .

Pour ajouter un nouveau joueur, on tire une position au hasard dans la liste et on y insère le nouveau tueur (en prenant garde de correctement mettre à jour les autres joueurs).

1. Dans la classe Principale (contenant le main), créez une méthode dont la signature correspond à `void inserer(String nom, int age, Tueur t1)`, qui crée un nouveau Tueur avec les informations données, tire une position au hasard dans la liste des tueurs (pour tirer au hasard un entier entre 0 et *nombre*, on peut utiliser l'instruction `(int)(Math.random()*nombre)`), et insère le nouveau tueur juste après l'élément dont on a tiré la position au hasard.

## 6 Que la partie commence !

Créer dans la classe Principale une méthode de signature `void partie(Tueur t1)` qui lance la partie et prend un seul argument : le premier tueur de la liste. Le jeu se déroule de la manière suivante : on choisit un joueur au hasard et on supprime son contrat. Le contrat du joueur exécuté devient alors celui du tueur qui l'a éliminé. Attention : si le joueur éliminé est le premier de la liste, il faudra remplacer le joueur de référence servant à parcourir la liste. À chaque fois qu'un joueur est éliminé, afficher un message (par exemple Blondin a tué Sentenza). La méthode affichera enfin le nom du tueur survivant et vainqueur du jeu.