

# La pierre et le sabre

## Java faire mal !

### 1 Environnement GitHub

#### *Création de votre dépôt distant pour ce sujet de TP*

Connectez-vous et placez-vous sur la page de vos dépôts (*repositories*). Cliquez sur le bouton New pour démarrer la création du dépôt. Renseignez :

- le nom du dépôt (pourquoi pas "PierreSabre" ?),
- la description ("Dépôt pour le projet Java de TP La pierre et le sabre en ILU1" vous conviendrait ?),
- la visibilité 'Public' (au moins pour celui-ci),
- l'ajout d'un fichier 'README' initial (qui définit ce qu'il faut lire en premier sur le dépôt),
- l'ajout d'un fichier .gitignore en utilisant le template java (pour spécifier les fichiers qui ne seront pas automatiquement rajoutés car considérés comme inutiles sur le dépôt)
- pas de précision de licence pour le moment.

#### *Clonage local initial de votre projet*

- Avec un **explorateur de fichier** retrouver sous votre dossier workspace le fichier .metadata puis cliquer droit (sans rien sélectionner) et sélectionner 'Git Bash Here'.
- Dans votre navigateur, sur la page de GitHub (illustration ci-dessus)
  - Appuyer sur le bouton 'Code' :
  - En restant sur le mode HTTPS, on peut copier l'URL du dépôt présentée dans la pop-up au moyen du bouton à droite du champ de texte :  
(<https://github.com/userilufredo/PierreSabre.git> pour ce qui concerne cet exemple).
- Sous Git Bash : utiliser la commande git 'git clone'+ l'URL copié ci-dessus en cliquant droit puis sur 'paste'.

**exemple :** `git clone https://github.com/userilufredo/PierreSabre.git`

#### *Créer le projet Java*

Dans Eclipse, créer un projet Java dans votre workspace avec le même nom que votre dépôt.

File > New > Java Project puis "PierreSabre".

Dans la partie "Module" décocher l'option : "Create module-info.java file".

Cliquer sur le bouton "Finish"

## 2 Java

On désire réaliser un programme Java permettant d'écrire facilement des histoires de Samouraïs dans lesquelles apparaissent des commerçants, des ronins et des yakuzas.

=> créer une classe **HistoireTP4**, dans le package **histoire**, qui contient une méthode **main**, et qui vous servira de scénario pour tester tous les objets et les méthodes que vous allez écrire. Toutes les autres classes seront placées sous le package **personnages**.

### 1- Tous humains !

Tous les intervenants sont des humains. Un humain est caractérisé par son nom, sa boisson favorite (de type `String`), et la quantité d'argent qu'il possède (de type `int`). Tous les attributs de la classe **Humain** sont privés. Pour manipuler les attributs, un humain propose des accesseurs en lecture (méthodes **get**) sur les attributs `nom` et `argent` ;

Un humain peut :

- dire bonjour, via une méthode **direBonjour()**
  - **Bonjour ! Je m'appelle ... et j'aime boire du ...**
- boire, via une méthode **boire()**
  - **Mmmm, un bon verre de .... ! GLOUPS !**
- s'acheter un bien, via la méthode **acheter(String bien, int prix)**,
- gagner ou perdre de l'argent, via deux méthodes **gagnerArgent(int gain)** et **perdreArgent(int perte)**

Les méthodes **direBonjour**, **boire** et **acheter** font appel à la méthode **parler(texte)** qui permet de faire parler les humains pendant le scénario

La méthode **parler()**, comme pour les Gaulois des TP précédents, ne semble utile que pour factoriser du code d'autres méthodes de la classe **Humain**. Que peut-on en déduire sur sa visibilité ?

- Écrire la classe **Humain** avec ses champs et ses méthodes selon la description précédente. Cette classe possède un constructeur à trois paramètres : le nom de l'humain, sa boisson préférée, et la quantité d'argent qu'il possède.
- Coder les méthodes **gagnerArgent**, **perdreArgent**, **direBonjour**, **boire**, **acheter** et **parler**.
- Tester dans la classe **HistoireTP4** pour obtenir le scénario ci-dessous.

(Prof) - Bonjour ! Je m'appelle Prof et j'aime boire du kombucha.

(Prof) - J'ai 54 sous en poche. Je vais pouvoir m'offrir une boisson à 12 sous

(Prof) - Mmmm, un bon verre de kombucha ! GLOUPS !

(Prof) - J'ai 42 sous en poche. Je vais pouvoir m'offrir un jeu à 2 sous

(Prof) - Je n'ai plus que 40 sous en poche. Je ne peux même pas m'offrir un kimono à 50 sous.

d. Sauvegarder votre travail sous GitHub (cf p1 - Historisation périodique de votre projet - exemple de message pour le commit "création de la classe Humain")

## 2- Commerçants, Ronins, et Yakuzas

Les commerçants, les ronins et les yakuzas sont tous des humains, mais avec des particularités supplémentaires. On parle de *spécialisation*, ou d'*héritage* de la classe `Humain`.

### 2.1 Commerçant

Un commerçant est un humain dont la boisson préférée est le thé. Il peut se faire extorquer son argent : il perd tout son argent et il parle pour dire que le monde est vraiment trop injuste. Il peut aussi recevoir de l'argent d'un ronin qu'il remercie alors avec déférence.

a. Écrire la classe `commercant` : coder les méthodes `int seFaireExtorquer()` et `void recevoir(int argent)`. Concernant la méthode `seFaireExtorquer` vous n'arriverez pas à visualiser l'attribut `argent` de votre classe mère `Humain`. Pour résoudre ce problème vous aurez le choix entre 2 solutions, lesquelles ?

Solution 1 : \_\_\_\_\_

Solution 2 : \_\_\_\_\_

Réfléchissez bien, votre encadrant contrôlera que vous avez choisi la solution préférable 😊. Cette solution sera également appliquée à `parler()`.

b. Tester dans la classe `HistoireTP4` pour obtenir le scénario ci-dessous. Le marchand à 20 sous en poche à sa création.

(Marco) - Bonjour ! Je m'appelle Marco et j'aime boire du thé.

(Marco) - J'ai tout perdu! Le monde est trop injuste...

(Marco) - 15 sous ! Je te remercie généreux donateur!

(Marco) - Mmmm, un bon verre de thé ! GLOUPS !

c. Sauvegarder votre travail sous GitHub (cf p1 - Historisation périodique de votre projet - exemple de message pour le commit "création de la classe Commerçant").

## 2.2 Yakuza

Un yakuza est un humain appartenant à un clan (chaîne de caractères), et caractérisé par sa réputation (un entier par défaut égal à 0).

Il peut extorquer un commerçant : il prend tout l'argent du commerçant, gagne un point de réputation et annonce ce qu'il vient de faire.

Le constructeur de cette classe a les mêmes paramètres que ceux de sa classe mère plus une chaîne de caractères représentant le clan du yakuza.

- Écrire la classe **Yakuza** : coder la méthode `void extorquer(Commerçant victime)`.
- Tester dans la classe **HistoireTP4** pour obtenir le scénario ci-dessous. Le yakuza à 30 sous en poche à sa création et fait partie du clan Warsong.

*(Yaku Le Noir) - Bonjour ! Je m'appelle Yaku Le Noir et j'aime boire du whisky.*

*(Yaku Le Noir) - Tiens, tiens, ne serait-ce pas un faible marchand qui passe par là ?*

*(Yaku Le Noir) - Marco, si tu tiens à la vie donne moi ta bourse !*

*(Marco) - J'ai tout perdu! Le monde est trop injuste...*

*(Yaku Le Noir) - J'ai piqué les 15 sous de Marco, ce qui me fait 45 sous dans ma poche. Hi ! Hi !*

- Sauvegarder votre travail sous GitHub (cf p1 - Historisation périodique de votre projet - exemple de message pour le commit "création de la classe Yakuza")

## 2.3 Ronin

Le ronin est un humain, il a un attribut d'honneur initialisé à 1.

On lui ajoute une méthode lui permettant de donner 10% de son argent à un commerçant,

- Écrire la classe **Ronin** : coder la méthode `void donner(Commerçant beneficiaire)`.
- Tester dans la classe **HistoireTP4** pour obtenir le scénario ci-dessous. Le ronin à 60 sous en poche à sa création.

*(Roro) - Bonjour ! Je m'appelle Roro et j'aime boire du shochu.*

*(Roro) - Marco prend ces 6 sous.*

*(Marco) - 6 sous ! Je te remercie généreux donateur!*

- Sauvegarder votre travail sous GitHub (cf p1 - Historisation périodique de votre projet - exemple de message pour le commit "création de la classe Ronin").

### 3- Honneur, réputation et duel entre les Ronins et les Yakuza

Au cours d'un duel avec un ronin, le Yakuza peut :

- perdre (méthode `int perdre()`) il perd tout son argent, perd un point de réputation, et annonce sa défaite.
- gagner (méthode `gagner(int gain)`) : il prend l'argent gagné à son adversaire, gagne en réputation et crie sa victoire.

Le Ronin peut provoquer un yakuza en duel (méthode `provoquer(Yakuza adversaire)`). Sa force détermine l'issue du duel. Elle vaut le double de son honneur.

- Si la force du ronin est plus grande ou égale à la réputation du yakuza, le ronin gagne : il prend l'argent du yakuza, gagne en honneur et annonce sa victoire.
- Sinon, il perd ; on décrémente son compteur honneur, il perd tout son argent, et il râle à cause de sa défaite.

En cas de victoire on fait appel à la méthode `perdre` du yakuza et inversement en cas de défaite on fait appel à la méthode `gagner` du yakuza.

a. Écrire la classe **Ronin** : coder la méthode `perdre` et `gagner` dans la classe **Yakuza** et `provoquer` dans la classe **Ronin**. Vous aurez peut-être besoin d'un getteur dans l'une des deux classes 😊.

b. Tester dans la classe **HistoireTP4** pour obtenir le scénario ci-dessous.

*(Roro) - Je t'ai retrouvé vermine, tu vas payer pour ce que tu as fait à ce pauvre marchand!*

*(Roro) - Je t'ai eu petit yakusa!*

*(Yaku Le Noir) - J'ai perdu mon duel et mes 45 sous, snif... J'ai déshonoré le clan de Warsong.*

Modifier dans la classe yakuza l'initialisation de sa réputation pour 4. Relancer le main, vous devez obtenir le scénario ci-dessous.

*(Roro) - J'ai perdu contre ce yakuza, mon honneur et ma bourse ont en pris un coup.*

*(Yaku Le Noir) - Ce ronin pensait vraiment battre Yaku Le Noir du clan de Warsong ? Je l'ai dépouillé de ses 54 sous.*

Remettre l'initialisation de la réputation du yakuza à 0.

c. Sauvegarder votre travail sous GitHub (cf p1 - Historisation périodique de votre projet - exemple de message pour le commit "duels entre Ronin et Yakuza").