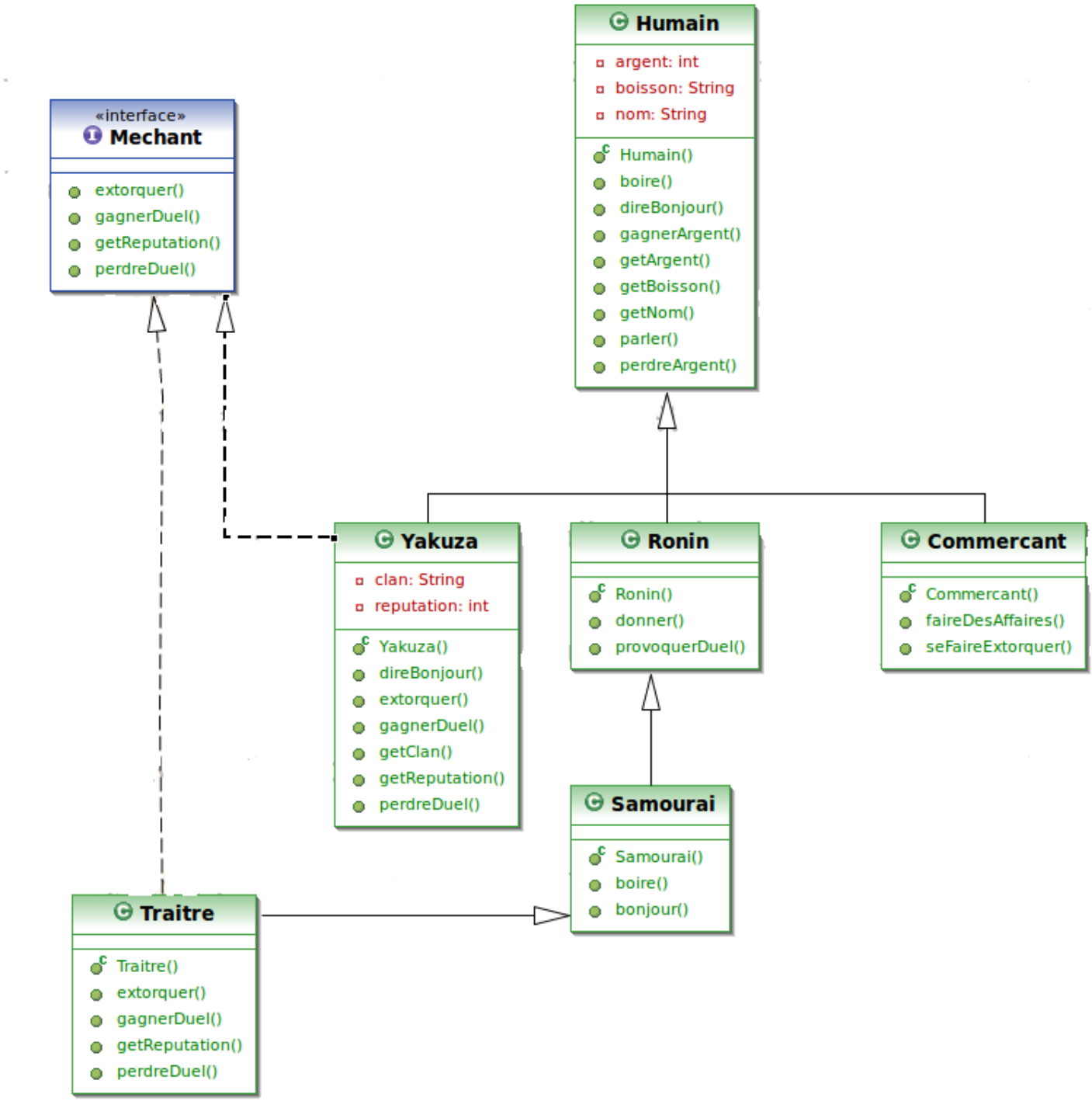
TP1 – Partie 2
« Des histoires de Samouraïs »

On désire réaliser un programme Java qui permet de décrire des histoires de Samouraïs.

Ci-dessous, le diagramme de classes correspondant aux classes que vous devrez écrire lors du TP. Il est incomplet et va évoluer au fur et à mesure de votre avancement. Pensez à le mettre à jour pour assurer une homogénéité de la documentation de votre projet.



1ère étape : tous humains.

Tous les personnages de l'histoire sont des humains. Un **humain** est caractérisé par son nom, sa boisson favorite, et la quantité d'argent qu'il possède.

⇒ Écrivez la classe Humain en spécifiant ses attributs, puis utilisez les outils de *NetBeans* pour générer automatiquement le constructeur.

⇒ Ajoutez les méthodes permettant à un humain de :

- parler : affiche le nom de l'humain suivi du texte passé en paramètre
- dire bonjour : utilise la méthode précédente pour afficher une message contenant le nom, la boisson favorite et la somme d'argent de cet humain
- boire : utilise la méthode parler pour afficher un message « Ahhh, un bon verre de [boisson] ! GLOUPS ! »
- gagner et perdre de l'argent (le montant est passé en paramètre)

2ème étape : on spécialise.

Un **commerçant** est un humain dont la boisson préférée est le thé. Il peut se faire extorquer : il perd tout son argent et il parle pour dire que le monde est vraiment trop injuste. Il peut aussi faire des affaires avec un humain et pour un montant, donnés en paramètre. Dans ce cas l'humain perd la somme d'argent correspondante et le commerçant la gagne, puis il affiche un message remerciant l'humain.

⇒ Écrivez la classe Commercant conformément au diagramme de classes.

- Vous aurez besoin d'utiliser le mot-clé super pour accéder au constructeur de la classe humain
- Testez deux techniques pour accéder dans la classe Commercant aux attributs de la classe humain, expliquez pourquoi celle mise en œuvre dans le diagramme UML est préférable. Modifiez la classe humain.

Un **yakuza** est un humain appartenant à un clan (représenté par une 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. Lorsqu'il dit bonjour, il indique également son clan d'appartenance.

Un yakuza peut participer à des duels : s'il perd le duel alors il perd tout son argent, perd un point de réputation, et annonce sa défaite. Dans le cas contraire, il gagne un point de réputation et crie sa victoire (et ne gagne pas d'argent).

⇒ Écrivez la classe Yakuza

- La méthode direBonjour() a déjà été définie dans la classe Humain. Retrouvez le nom et le principe associés à ce concept.
- On peut utiliser l'annotation @Override pour indiquer au compilateur qu'il peut faire des vérifications supplémentaires liées à ce principe (entre autres).

Le **ronin** est un humain qui a un attribut d'honneur entier initialisé à 1. Il peut donner de l'argent à un commerçant (commerçant et montant passés en paramètres). Il peut également provoquer un yakuza (passé en paramètre) en duel.

Les règles du duel sont les suivantes : si le double de l'honneur du ronin est plus grand que la réputation du yakuza, le ronin gagne : il prend l'argent du yakuza, gagne un point d'honneur et annonce sa victoire. S'il perd, on décrémente son compteur honneur, et il râle à cause de sa défaite (et ne perd pas d'argent). 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.

⇒ Écrivez la classe Ronin

Les **samouraïs** sont des ronins liés à un seigneur (représenté par une chaîne de caractères, contenant le nom). Le nom de son seigneur est donné au constructeur. Lorsqu'un samouraï se présente (dit bonjour), il annonce quel seigneur il sert.

Alors que tous les autres humains ne peuvent boire que leur boisson favorite, le samouraï a une super capacité spéciale : il peut boire n'importe quelle boisson. C'est le seul à pouvoir boire du thé en journée, prendre des bières à l'apéro et accompagner son repas de saké.

⇒ Écrivez la classe Samourai

- La méthode boire() a déjà été définie dans la classe Humain. Retrouvez le nom et le principe associés à ce concept.
- Si dans la méthode direBonjour() on utilise le mot-clé super pour appeler la méthode direBonjour() de la classe parente, quelle méthode de quelle classe sera en réalité utilisée ?

3ème étape : on teste, on vérifie et on valide.

⇒ Dans la classe **MonHistoire**, ajoutez une méthode main() qui permet de tester les différentes classes écrites jusqu'à présent, par exemple :

```
public static void main(String[] args) {
    Humain h = new Humain("Prof", "Porto", 10);
    h.direBonjour();
    h.boire();

    Commercant c = new Commercant("Marchant", 35);
    c.direBonjour();

    Yakuza y = new Yakuza("Yaku le noir", "biere", 42, "WarSong");
    y.extorquer(c);

    Ronin r = new Ronin("Roro", "sake", 61);
    r.donner(c, 10);
    r.provoquerDuel(y);
    r.direBonjour();
}
```

4ème étape : on implémente de nouvelles fonctionnalités.

On veut ajouter à notre histoire des samouraïs traîtres qui peuvent extorquer des commerçants et qui peuvent être provoqués en duel comme un yakuza. De plus, lorsqu'un traître extorque un pauvre commerçant son honneur diminue.

D'après cette description, on aurait tendance à vouloir faire hériter cette classe à la fois de Samourai et de Yakuza. Mais en java, l'héritage multiple est interdit. Heureusement, on peut s'en sortir en utilisant une interface qui regroupe toutes les fonctionnalités qui nous intéressent, ici celles qui correspondent à un comportement de méchant.

⇒ Écrivez l'interface **Mechant** comme spécifié dans le diagramme de classes.

- Exprimez qu'un Yakuza est méchant
- Un ronin peut provoquer n'importe quel type de méchant, que faut-il modifier ?

⇒ Écrivez la classe **Traitre**, utilisez les outils de *NetBeans* pour générer automatiquement les squelettes des méthodes, puis complétez le code manquant. Ici apparaît un petit problème : le traître est supposé avoir une réputation (qui est uniquement définie dans Yakuza), et non un honneur (qui est hérité de Ronin). On peut créer un attribut réputation, et il faudrait dans ce cas gérer 2 attributs (réputation et honneur). Ou on peut assimiler l'honneur pré-existant à la réputation. Choisissez cette option et discutez des modifications nécessaires dans le diagramme de classes.

⇒ Complétez votre classe de test pour prendre en compte ce nouveau personnage.