

Implémentation OO

Java

1 But

Le but de ces exercices est de mettre en pratique l'implémentation des concepts de base de l'orienté-objet en Java.

Parfois, il s'agira également de réaliser l'analyse et la conception avant d'en faire l'implémentation.

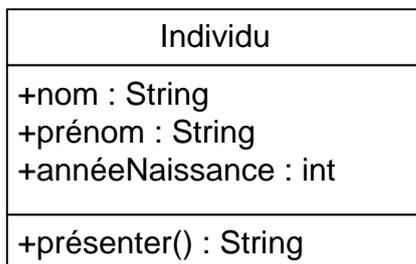
2 Prérequis

Connaître le diagramme de classe en UML.

3 Travail à réaliser

3.1 L'individu

Soit le diagramme de classes suivant :

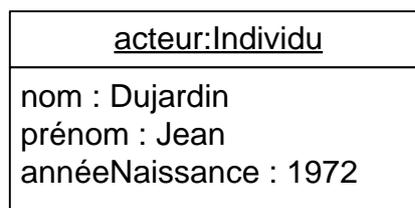
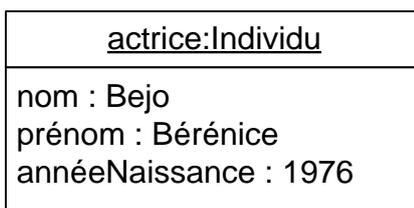


Implémentez cette classe en Java.

La méthode présenter() doit retourner une chaîne de caractères de ce format :

Je m'appelle *prénom nom* et je suis né(e) en *annéeNaissance*.

Ensuite, créez une classe Application contenant la méthode statique main() et instanciez les objets selon ce diagramme d'objets :



Les faire se présenter.



EXERCICE		INF3	 CEJEF DIVISION TECHNIQUE ÉCOLE DES MÉTIERS TECHNIQUES
M226 / Implémentation OO			
ICT – INF 3 – 226 ED 01	JCO	Mise à jour : 29.08.2016	

Affichez ensuite celui qui est le plus jeune.

Voici un exemple d'affichage :

```
Je m'appelle Jean Dujardin. Je suis né(e) en 1972.
Je m'appelle Bérénice Bejo. Je suis né(e) en 1976.
L'individu le plus jeune : Bejo
```

3.2 L'individu, bis

À partir des classes créées à l'exercice 3.1, écrivez un programme qui réalise séquentiellement les opérations suivantes :

- Créer les acteurs *Jean Dujardin* et *Bérénice Bejo* ;
- Les faire se présenter ;
- Lassé de sa notoriété, *Dujardin* change de nom en *Dupotager*.
- *Bejo* choisi de prendre le nom de son mari. Elle s'appelle désormais *Hazanavicius*.
- Suite à ces changements, ces acteurs se présentent.

Voici un exemple d'affichage :

```
Je m'appelle Jean Dujardin. Je suis né(e) en 1972.
Je m'appelle Bérénice Bejo. Je suis né(e) en 1976.
Nous changeons de noms.
Je m'appelle Jean Dupotager. Je suis né(e) en 1972.
Je m'appelle Bérénice Hazanavicius. Je suis né(e) en 1976.
```

3.3 L'individu, bis repetita

En reprenant les classes de l'exercice 3.1, vous décidez d'améliorer leur robustesse en encapsulant les attributs privés.

Modifiez la classe *Individu* de telle sorte que les attributs *nom*, *prénom* et *annéeNaissance* ne puissent plus être directement modifiés, mais uniquement au travers de mutateurs (méthodes *set*) et accesseurs (méthodes *get*).

Les mutateurs devront valider les données fournies avant de modifier les attributs de la classe :

- Les noms et prénoms doivent contenir au moins un caractère.
- L'année de naissance ne doit pas être inférieure à 1900 ni supérieure à 2200.

En cas de donnée invalide, aucun changement n'est effectué et un message d'erreur est inscrit sur la console de sortie.

Voici un exemple d'affichage :

```
Je m'appelle Jean Dujardin. Je suis né(e) en 1972.
Je m'appelle Bérénice Bejo. Je suis né(e) en 1976.
Le prénom doit contenir au moins un caractère.
Le nom doit contenir au moins un caractère.
2300 n'est pas une année valide.
Je m'appelle Jean Dujardin. Je suis né(e) en 1972.
Je m'appelle Bérénice Bejo. Je suis né(e) en 1976.
L'individu le plus jeune : Bejo
```

EXERCICE		INF3	 CEJEF DIVISION TECHNIQUE ÉCOLE DES MÉTIERS TECHNIQUES
M226 / Implémentation OO			
ICT – INF 3 – 226 ED 01	JCO	Mise à jour : 29.08.2016	

3.4 L'individu, encore et toujours

Complétez la classe Individu de l'exercice précédent afin d'y ajouter différents constructeurs :

- un constructeur prenant un nom, un prénom et une année de naissance ;
- un constructeur prenant un nom et un prénom, l'année de naissance étant fixée à 2000.

Attention à ne pas dupliquer inutilement le code.

Modifier la classe Individu de sorte que l'individu se présente en écrivant le code suivant :

```
Individu monIndividu = new Individu(...);
System.out.println(monIndividu);
```

Modifier la phrase de présentation pour qu'elle affiche ceci :

```
Je m'appelle prénom nom. Cette année, je fête mes x ans.
```

Toutefois, si l'individu n'est pas encore né, afficher :

```
Je m'appelle prénom nom. Je ne suis pas encore né(e).
```

Si l'individu est né cette même année, afficher :

```
Je m'appelle prénom nom. Je suis né(e) cette année !
```

Pour connaître l'année en cours, utiliser le code suivant (uniquement depuis *Java 8*) :

```
LocalDate maintenant = LocalDate.now();
int cetteAnnee = maintenant.getYear();
```

Pour tester tous les cas de figure possibles, créez les individus suivants :

- L'acteur *Jean Dujardin*, né en 1972.
- L'actrice *Bérénice Bejo*, née en 1976.
- Aidan Clinton*, né en 2016¹.
- La biologiste de fiction, *Alexa Komarova* née en 2051 (ou 2052 mais peu importe).

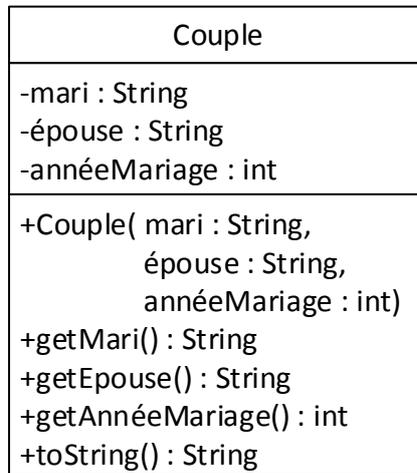
Voici un exemple d'affichage :

```
Je m'appelle Jean Dujardin. Cette année, je fête mes 41 ans.
Je m'appelle Bérénice Bejo. Cette année, je fête mes 37 ans.
Je m'appelle Charlotte de Cambridge. Je suis né(e) cette année !
Je m'appelle Alexa Komarova. Je ne suis pas encore né(e).
```

¹ Si cet exercice est fait à un autre moment que l'année 2016, utiliser un autre personnage, dont l'année de naissance correspond à l'année scolaire en cours.

3.5 Le couple

Créez une classe Couple implémentant le diagramme de classes suivant :

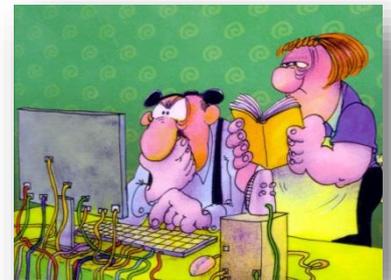


La fonction toString() doit retourner une chaîne de caractères sous la forme suivante :

épouse et *mari* se sont mariés en *annéeMariage*.

Testez votre classe avec, par exemple, les couples suivants :

- *Pierre et Marie Curie*, mariés en 1895.
- *Michel Berger et France Gall*, mariés en 1976.
- *Robert et Raymonde Bidochon*, mariés en 1977.



Déboguez ensuite le code suivant :

```
Couple lesClintons;
System.out.println("Bill et Hillary se sont mariés en" +
    lesClintons.getAnnéeMariage());
```

3.6 Les calories

L'entreprise qui vous embauche vous demande de créer un programme permettant de compter les calories absorbées en consommant des aliments.

Les aliments contiennent une certaine quantité de lipides, de glucides et de protéines (généralement données en grammes). Ces quantités permettent de savoir combien de calories contient un aliment.

Un compteur de calories se chargera, chaque fois qu'un aliment est mangé, d'accumuler les quantités de lipides, glucides et protéines absorbées et d'en déduire le nombre de calories que cela représente. Si ce nombre dépasse une limite préalablement fixée, un message est affiché.

Plus précisément, un aliment est désigné par un nom et contient un certain nombre de lipides, de glucides et de protéines.



EXERCICE		INF3	 CEJEF DIVISION TECHNIQUE ÉCOLE DES MÉTIERS TECHNIQUES
M226 / Implémentation OO			
ICT – INF 3 – 226 ED 01	JCO	Mise à jour : 29.08.2016	

Il faut pouvoir afficher les caractéristiques d'un aliment avec `System.out.println()`. En plus de ces caractéristiques, il faut indiquer si l'aliment en question est sain ou non, au moyen de la méthode de calcul suivante² :

*Si nombre lipides [g] * 7 < nombre glucides [g] + nombre protéines : aliment sain.
 Sinon : aliment malsain.*

De plus, il faut pouvoir vérifier si deux aliments sont identiques.

Le compteur de calories mémorise la quantité totale de lipides, glucides et protéines absorbées en mangeant des aliments (soit en fournissant directement l'aliment à manger, soit en fournissant la quantité de lipides, glucides et protéines). Il mémorise également une limite (en kilocalories [kcal]) à ne pas dépasser.

Chaque fois qu'un nouvel aliment est mangé, il faut vérifier si la limite de calories n'a pas été dépassée. Si c'est le cas, un message d'avertissement s'affiche sur la console.

Le tableau suivant permet de calculer les calories :

	lipides	glucides	protéines
Nombre de kcal/g	9	4	4

Il faut pouvoir également remettre les compteurs à zéro (sans toucher à la limite).

Le compteur de calories doit également pouvoir nous renseigner sur le pourcentage de graisses (les lipides) absorbées.

Toutes les classes développées doivent implémenter la méthode `toString()`.

Travail à réaliser :

1. À partir du cahier des charges, établir le diagramme de classes.
2. Implémenter l'application selon votre diagramme de classes.
3. Tester l'application avec les informations suivantes :

	lipides	glucides	protéines
Un croissant	4	38	5
Une tasse de lait	101	166	23
Un verre de coca	0	35	0
Une portion de frites	30	78	7
Une côte de porc	75	0	39

Limite journalière (par exemple) : 2700 kcal.

4. Une fois tous les aliments avalés, afficher le pourcentage de graisse que cela représente.

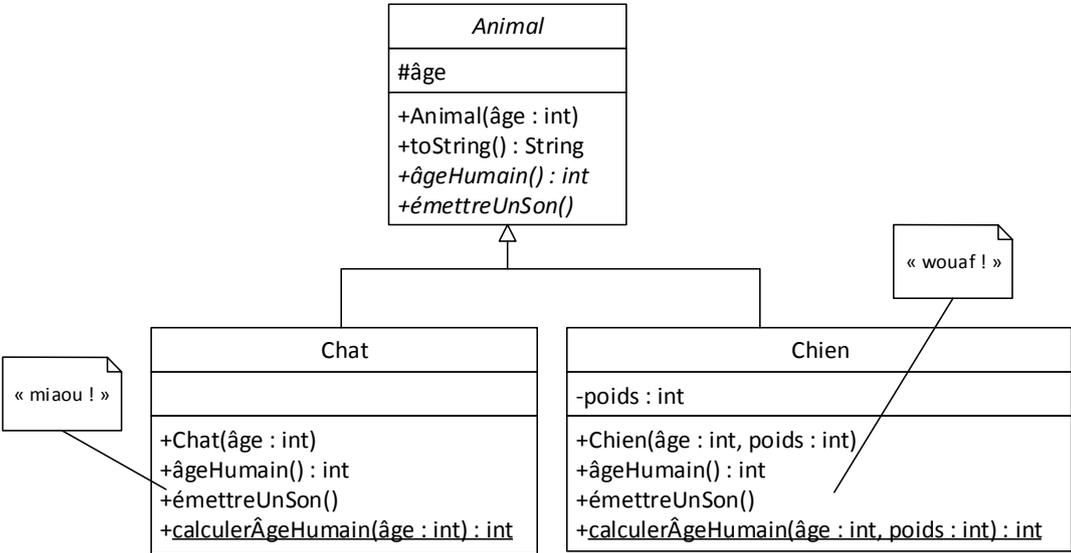
² Qui ne se base sur aucune rigueur scientifique et qui n'est donc pas digne de confiance.

Voici un exemple d'affichage :

```
=> Déjeuner
croissant avalé(e) !
tasse de lait avalé(e) !
=> Dîner
verre de coca avalé(e) !
portion de frites avalé(e) !
côte de porc avalé(e) !
Attention, limite de 2700 cal dépassée.
Pourcentage de graisse : 34%
```

3.7 Chien et chat

Implémenter en Java le diagramme de classes suivant :



Créer une application qui instancie les animaux suivants, leur fait émettre un son, puis les affiche sur la console :

médor:Chien
âge : 7 ans Poids : 15 kilos

minet:Chat
âge : 5 ans

Ça vous embête pas si on fait un petit somme pendant que vous bossez ?



Voici les formules³ permettant d'obtenir l'âge humain équivalent pour les chiens et les chats :

Chien de moins de 15 kilos	$f(\text{âge}) = \begin{cases} 20, & \text{âge} = 1 \\ 28 + (\text{âge} - 2) \cdot 4, & \text{âge} \geq 2 \end{cases}$
Chien de 15 kilos à 40 kilos.	$f(\text{âge}) = \begin{cases} 18, & \text{âge} = 1 \\ 27 + (\text{âge} - 2) \cdot 6, & \text{âge} \geq 2 \end{cases}$

³ Chiffres tirés de <http://www.age-humain.com/>

Chien de plus de 40 kilos.	$f(\hat{age}) = \begin{cases} 16, & \hat{age} = 1 \\ 22 + (\hat{age} - 2) \cdot 9, & \hat{age} \geq 2 \end{cases}$
Chat	$f(\hat{age}) = \begin{cases} 19, & \hat{age} = 1 \\ 24 + (\hat{age} - 2) \cdot 4, & \hat{age} \geq 2 \end{cases}$

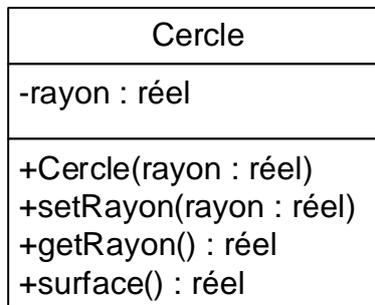
Voici un exemple d'affichage :

```

wouaf !
miaou !
J'ai 7 ans (57 ans humain).
J'ai 5 ans (36 ans humain).
```

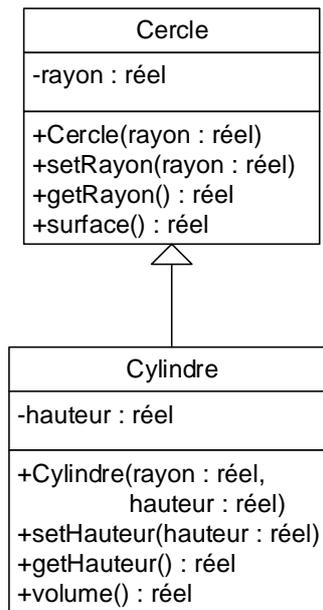
3.8 Cercle et cylindre

Une application qui gère quelques formes géométriques 2D possède la classe suivante :



Vous êtes chargé de créer une classe représentant un cylindre et permettant de définir la hauteur du cylindre et de calculer son volume.

Votre chef vous propose d'implémenter le diagramme de classes suivant :



Que pensez-vous de sa suggestion ? Quels sont les avantages et inconvénients de cette solution ? Y aurait-il une meilleure solution ?

3.9 Ulysse 31

Vous êtes chargé par une entreprise informatique de participer à la conception d'un jeu vidéo basé sur l'univers d'une vieille série animée dénommée *Ulysse 31*.



Plus particulièrement, vous êtes en charge de modéliser tout ce qui concerne l'équipage d'un vaisseau et ses membres. Voici les informations vous permettant d'y arriver :

Un vaisseau est caractérisé par un nom et contient un équipage.

Les membres de l'équipage peuvent être de différentes races : les humains, les zotriens et les robots.

Ils ont tous un nom et un âge. Un membre de l'équipage est capable de dire si oui ou non il sait piloter un vaisseau.

Un humain n'a pas de pouvoir particulier. Un zotrien est capable de lire dans les pensées et un robot est capable de calculer.

Un robot possède une caractéristique particulière : sa vitesse de processeur (un nombre réel).

Selon leur race et leurs caractéristiques, les membres de l'équipage sont capables de faire preuve de plus ou moins de sagesse. Cela est représenté par des points de sagesse (un nombre entier), calculés ainsi :

Race	Points de sagesse
Humain	Si âge < 12 : zéro Si âge ≥ 90 : zéro Sinon : âge + 20
Zotrien	$\sqrt{\text{âge}}$
Robot	(Vitesse du processeur) ^π

Il est possible de connaître le membre de l'équipage qui fait preuve de moins de sagesse (si plusieurs membres ont la même sagesse, renvoyer le dernier de la liste).

Lorsqu'un équipage est affecté à un vaisseau, le premier membre sachant piloter est désigné comme pilote du vaisseau (si au sein de l'équipage, personne ne sait piloter, le vaisseau refuse d'utiliser l'équipage proposé).

Un vaisseau doit pouvoir indiquer qui est son pilote.

La méthode `toString()` du vaisseau doit retourner une chaîne ayant le format suivant :

Nom vaisseau : taille équipage membres d'équipage. Pilote : nom pilote

Travail à réaliser :

1. À partir du cahier des charges, établir le diagramme de classes.
2. Implémenter l'application selon votre diagramme de classes.
3. Tester l'application avec les informations suivantes :

Nom du vaisseau : Odysseus

Membres de l'équipage :

Ulysse, 40 ans, humain, sait piloter.

Télémaque, 13 ans, humain, ne sait pas piloter.



EXERCICE		INF3	 CEJEF DIVISION TECHNIQUE ÉCOLE DES MÉTIERS TECHNIQUES
M226 / Implémentation OO			
ICT – INF 3 – 226 ED 01	JCO	Mise à jour : 29.08.2016	

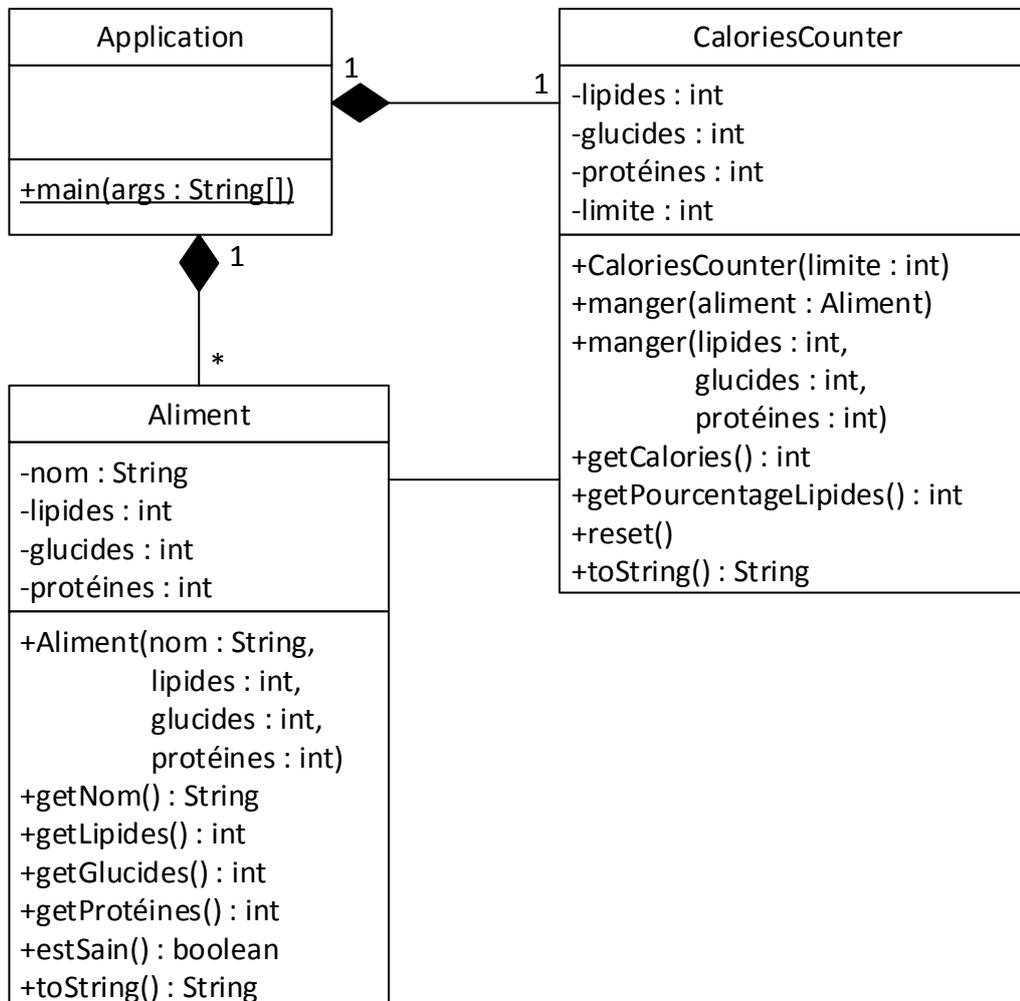
Thémis, 11 ans, zotrienne, ne sait pas piloter.
 Nono, 1 ans, robot, sait piloter, vitesse du processeur : 3.44.

Voici un exemple d'affichage :

Odysseus : 4 membres d'équipage. Pilote : Ulysse Le membre d'équipage ayant le moins de sagesse est Thémis.
--

M226 / Implémentation OO**4 Aides****4.1 Les calories**

Voici un diagramme de classe envisageable :



M226 / Implémentation OO

ICT – INF 3 – 226 ED 01

JCO

Mise à jour : 29.08.2016

4.2 Ulysse 31