

Conception et programmation objets avancées

Sujet 1 : révisions et prise en main de l'environnement de travail

Vous utiliserez l'environnement de développement intégré (IDE) *IntelliJ IDEA* qui offre de nombreuses fonctionnalités. Vous penserez à tirer profit au maximum de l'IDE : création automatique des constructeurs, modificateurs et accesseurs, création de sous-classes,... Vous testerez votre code au fur et à mesure.

Le logiciel est normalement installé sur les machines du département. Si nécessaire, il est téléchargeable à partir de <https://www.jetbrains.com>. Il s'agit d'un outil payant, mais mis gratuitement à disposition des étudiants après inscription sur le site de *JetBrains* : <https://www.jetbrains.com/shop/eform/students>. **Attention, vous devez utiliser votre adresse mail de l'Université d'Artois.** Vous recevrez un mail de demande de confirmation. Lors de la première utilisation d'*IntelliJ IDEA*, il vous faut fournir vos identifiants de connexion et vous pourrez utiliser le produit pendant un an.

Exercice 1

1. Créez une interface *Animal* déclarant les quatre méthodes suivantes.
 - *alimentation()* : retourne une valeur décrivant l'alimentation de l'animal en question : carnivore, herbivore ou omnivore. Les trois précédentes valeurs doivent être regroupées dans un type d'énumération (enum) dans un fichier à part.
 - *vertebre()* : retourne un booléen, vrai si l'animal en question est un vertébré, faux dans le cas contraire.
 - *criAnimal()* : retourne la chaîne de caractères correspondant au cri de l'animal en question.
 - *afficher()* : affiche à l'écran les valeurs que retournent les trois précédentes méthodes.
2. Visualisez le diagramme de classe directement dans l'IDE. Depuis le diagramme de classe, créez une classe *Tigre* qui implémente l'interface *Animal* (un tigre est un animal carnivore dont le cri est le rugissement et il s'agit d'un vertébré) puis complétez le code pré-écrit.
3. Procédez de même pour créer une classe *Zebre* qui implémente l'interface *Animal* (un zèbre est un vertébré, herbivore et dont le cri est le hennissement).
4. Créez une classe *Zoo* utilisée pour décrire les zoos. On retient comme caractéristiques le nom du zoo et un tableau d'animaux, qui ne sont pas modifiables à l'extérieur de cette classe. On considère ici qu'un zoo ne peut contenir plus de 100 animaux, cette valeur devra être déclarée en tant que constante dans la classe. La classe *Zoo* doit contenir les méthodes suivantes.
 - un constructeur fournissant un nom au zoo et créant le tableau en utilisant la constante correspondant à la taille maximale,
 - un accesseur («getter») pour le nom du zoo,
 - une méthode qui affiche les caractéristiques de tous les animaux dans le zoo.
5. Créez une exception *TableauComplet* traduisant le fait que le tableau des animaux est complet. Le message d'erreur de toute instance de cette exception doit toujours être « Le zoo est complet ».
6. Ecrivez une méthode *AjoutAnimal* qui ajoute au zoo un animal fourni en paramètre. Cette méthode doit lever une exception (une instance de *TableauComplet*) dans le cas où il n'y a plus d'espace pour réaliser l'ajout.
7. Pour récupérer le nom d'une classe d'un objet, on peut utiliser la méthode `getClass().getName()`. Ajoutez une méthode retournant le nombre d'un type d'animaux (*Tigre*, *Zebre*, etc) d'un zoo.