

Instruction 5 :

Pour continuer notre programme, le chef de projet vous demande de créer une classe nommée “Zoo” et une autre classe nommée “Animal” comme suit :

La classe “Animal” contient les attributs suivants :

- String family
- String name
- int age
- boolean isMammal

La classe “Zoo” contient les attributs suivants :

- Animal [] animals
- String name
- String city
- int nbrCages

Maintenant créez deux objets dans la classe principale, un animal (lion) et un zoo (myZoo) et affectez à leurs attributs des valeurs.

NB : le zoo va contenir 25 animaux maximum.

Instruction 6 :

Comme vous le remarquez, initialiser les attributs d’un objet animal ou zoo dans la méthode main se fait sur plusieurs lignes, votre chef vous demande de changer ceci en utilisant un constructeur paramétré, pour ce faire ajoutez ces deux constructeurs dans les classe Zoo et Animal

```
public Animal(String family, String name, int age, boolean
isMammal) {
    this.family = family;
    this.name = name;
    this.age = age;
    this.isMammal = isMammal;
}
```

```
public Zoo(String name, String city, int nbrCages) {
    animals = new Animal[nbrCages];
    this.name = name;
    this.city = city;
    this.nbrCages = nbrCages;
}
```

Que remarquez-vous au niveau de la méthode main ?

Instruction 7 :

Résolvez le problème précédent puis et créez des animaux avec le constructeur paramétré.

Instruction 8 :

Créez la méthode “displayZoo()” dans la classe Zoo qui permet d’afficher la valeur des attributs d’un zoo(name, city et nbrCages), et l’invoquer dans la méthode “main” pour le zoo que vous avez créé.

Une fois que vous avez afficher les informations d’un zoo via cette méthode, essayez de directement afficher le zoo via ces deux instructions :

```
System.out.println(myZoo);  
System.out.println(myZoo.toString());
```

Que remarquez-vous ?

Instruction 9 :

Faites en sorte d’afficher les informations correctement avec ces deux instructions. Une fois résolu, faites la même chose pour la classe Animal.

Instruction 10 :

Créez, dans la classe Zoo, la méthode “boolean addAnimal(Animal animal)” qui permet d’ajouter un Animal dans le tableau “animals” et de retourner **true** si l’ajout a bien été fait.

Nb : accédez case par case. (Utilisez un compteur d’animaux)

Maintenant testez la méthode dans la classe principale en ajoutant plus d’animaux qu’il y a de place dans le zoo (invoquez la méthode plusieurs fois), que remarquez-vous ?

Instruction 11 :

Ajoutez une méthode qui vous affiche les animaux d’un zoo dans la classe Zoo et utilisez-la dans la classe principale.

Ajoutez une méthode “int searchAnimal(Animal animal)”qui cherche un animal, **selon son nom**, dans le zoo et retourne l’indice de l’animal s’il est trouvé sinon retourne “-1”.

Dans la classe principale, ajoutez un animal dans un zoo et testez la méthode de recherche en le cherchant, que retourne la méthode ?

Maintenant créez un autre animal identique au premier et cherchez le dans le zoo, que remarquez-vous ?

Instruction 12 :

Régalez le problème de l'instruction précédente et re testez la recherche
Appliquez les changements pour faire en sorte de respecter les conditions ci-dessous :

- Ajouter un animal une seule fois (un animal est unique) dans un zoo
- Interdire de dépasser le nombre maximal d'animaux dans un zoo

Instruction 13 :

Créez la méthode "boolean removeAnimal(Animal animal)" qui permet de supprimer un animal et de renvoyer le succès de la suppression.