**Prosit 3**



**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 :**

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.

**Instruction 14 :**

Modifier l'attribut nbrCages de la classe Zoo, en le rendant une constante.

N’oubliez pas qu’un Zoo peut contenir 25 animaux maximum.

Faites les changements nécessaires au niveau du constructeur.

**Instruction 15 :**

Créer la méthode **« boolean isZooFull() »,** une méthode qui vérifie si le zoo est plein en fonction du nombre d'animaux et du nombre de cages.

**Instruction 16 :**

Créer une méthode **« Zoo comparerZoo(Zoo z1, Zoo z2) »**, dans la classe Zoo, qui prend 2 objets de type Zoo comme paramètres et renvoie le Zoo avec le plus d'animaux.