

PROSIT 2

Instruction 5 :

Création d'une Classe Zoo :

```
public class Zoo {
    Animal[] animals;
    String name;
    String city;
    final int NB_CAGES = 25;
    int nbrCages;
}
```

Création d'une Classe Animal :

```
public class Animal {
    String family;
    String name;
    int age;
    boolean isMammal;
}
```

Classe Principale :

```
public class MainClass {
    public static void main(String[] args)
    {
        // Création d'un animal (lion) et attribution de valeurs aux attributs
        Animal lion=new Animal();
        lion.family = "Félin";
        lion.name = "Lion";
        lion.age = 5;
        lion.isMammal = true;

        // Création d'un zoo (myZoo) et attribution de valeurs aux attributs
        Zoo myZoo = new Zoo();
        myZoo.name = "Mon Zoo";
        myZoo.city = "VilleZoo";
    }
}
```

Instruction 6 :

```
public class Zoo {
    Animal[] animals;
    String name;
    String city;
    final int NB_CAGES = 25;
    public Zoo(String name, String city, int nbrCages) {
        this.animals = new Animal[NB_CAGES];
        // Initialisation du tableau d'animaux avec une taille maximale de 25
        this.name = name;
        this.city = city;
        this.nbrCages=nbrCages;
    }
}

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

Erreurs dans la classe main :

```
1 'Animal(java.lang.String, java.lang.String, int, boolean)' in 'Animal' cannot be applied to '()' :5
2 'Zoo(java.lang.String, java.lang.String, int)' in 'Zoo' cannot be applied to '()' :12
```

Ces erreurs se produisent parce que la classe Animal et la classe Zoo ont désormais des constructeurs paramétrés.

Lorsque des constructeurs paramétrés sont définis dans une classe, le constructeur par défaut n'est plus généré automatiquement.

Dans la méthode main, les objets Animal et Zoo sont créés sans utiliser les nouveaux constructeurs paramétrés, ce qui provoque ces erreurs.

Instruction 7 :

Résolvez le problème précédent

```
public class Zoo {
    Animal[] animals;
    String name;
    String city;
    final int NB_CAGES = 25;
    public Zoo() {}
    public Zoo(String name, String city) {
        animals = new Animal[NB_CAGES];
        this.name = name;
        this.city = city;
        this.nbrCages=nbrCages;
    }
}

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

Créez des animaux avec le constructeur paramétré.

```
public class MainClass {
    public static void main(String[] args)
    {
        Zoo myZoo = new Zoo("Belvidere", "Tunis",6);
        Animal chat = new Animal("Carnivore","Chat",2,true);
        Animal elephant = new Animal("Elephant", "Elephant", 10, true);
        Animal giraffe = new Animal("Giraffidae", "Giraffe", 8, true);
    }
}
```

Instruction 8 :

Création de la méthode “displayZoo ()” dans la classe Zoo et son invocation dans la méthode “main” pour le zoo que vous avez créé.

Classe Zoo :

```
public void displayZoo()
{
```

```

    System.out.println("Zoo: " + name + ", cité: " + city+", nombre de cages: "+nbrCages);
}

```

Classe Principale :

```

public class MainClass {
    public static void main(String[] args)
    {
        ...
        Zoo myZoo = new Zoo("Belvidere", "Tunis",6);
        ...
        myZoo.DisplayZoo();
    }
}

```

```

C:\Users\Ferdaws\.jdk\jbr-17.0.9\bin\java.exe "-javaagent:C:\Program Files\JetB
Zoo: Belvidere, cité: Tunis, nombre de cages: 6

Process finished with exit code 0

```

Essayez de directement afficher le zoo via ces deux instructions :

```

public class MainClass {
    public static void main(String[] args)
    {
        ...
        Zoo myZoo = new Zoo("Belvidere", "Tunis");
        Animal lion = new Animal("Félin", "Lion", 5, true);
        //myZoo.displayZoo();
        System.out.println(myZoo);
        System.out.println(myZoo.toString());
    }
}

```

```

C:\Users\Ferdaws\.jdk\jbr-17.0.9\bin\java.exe "-javaagent:C:\Program Files\JetB
Zoo@3cb5cdba
Zoo@3cb5cdba

Process finished with exit code 0

```

Lorsque on essaye d'afficher le zoo directement avec `System.out.println(myZoo)` ou `System.out.println(myZoo.toString())`, on remarquera que le résultat est différent de l'invocation de `displayZoo ()`.

Lorsque on utilise `System.out.println(myZoo)`, cela utilise la méthode `toString ()` par défaut de la classe `Object`, qui renverra une représentation de l'objet au format "nomDeLaClasse@adresseMémoire".

Instruction 9 :

```

public class Animal {
    ...
    @Override
    public String toString() {
        return "Family:"+family+" ,name:"+name+", age:"+age+", isMammal:"+isMammal;
    }
}
public class Zoo {
    ...
    @Override
    public String toString() {
        return "Zoo: " + name + ", cité: " + city+",Nombre de cages" + nbrCages;
    }
}

```

Instruction 10 :

```

public class Zoo {
    ...
    int nbAnimaux;
    ...
    public Zoo(String name, String city) {
        ...
        this.nbAnimaux = 0;
    }
    public void displayZoo()
    {
        System.out.println("Zoo: " + name + ", cité: " + city+", nombre d'animaux:
        "+nbAnimaux+", nombre de cages: "+nbrCages);
    }

    ...
    public boolean addAnimal(Animal animal) {
        if (nbAnimaux < nbrCages) {
            animals[nbAnimaux] = animal;
            nbAnimaux++;
            return true;
        } else {
            System.out.println("Le zoo est plein. Impossible d'ajouter plus d'animaux.");
            return false;
        }
    }
}

```

Classe Principale

```

public class MainClass {
    public static void main(String[] args)
    {
        ...
        Zoo myZoo = new Zoo("Belvidere", "Tunis",2);
        Animal lion = new Animal("Félin", "Lion", 5, true);
        Animal chat = new Animal("Carnivore", "Chat",2,true);
        Animal elephant = new Animal("Elephant", "Elephant", 10, true);
        Animal giraffe = new Animal("Giraffidae", "Giraffe", 8, true);
        myZoo.addAnimal(lion);
        myZoo.addAnimal(chat);
        myZoo.addAnimal(elephant);
        myZoo.addAnimal(giraffe);

        myZoo.displayZoo();
        ...
    }
}

```

```

C:\Users\Ferdaws\.jdk\jbr-17.0.9\bin\java.exe "-javaagent:C:\Program Files\Jet
Le zoo est plein. Impossible d'ajouter plus d'animaux.
Le zoo est plein. Impossible d'ajouter plus d'animaux.
Zoo: Belvidere, cité: Tunis, nombre d'animaux: 2, nombre de cages: 2

Process finished with exit code 0

```

Dans cet exemple, le zoo a une capacité de 2 cages. Le lion et le chat sont ajoutés avec succès car le zoo a des cages disponibles. Cependant, lorsque vous essayez d'ajouter un éléphant et une girafe, la méthode `addAnimal()` retourne false pour indiquer que le zoo est plein et qu'aucun autre animal ne peut être ajouté.

Instruction 11 :

Ajout d'une méthode qui affiche les animaux d'un zoo dans la classe Zoo

```
public void displayAnimalZoo(){
    for(int i=0;i<NB_CAGES;i++){
        if (animals[i]!=null)
        {
            System.out.println("animal["+i+"]=Family: "+animals[i].family+"
,name:"+animals[i].name+", age:"+animals[i].age+" ,isMammal:"+animals[i].isMammal);
        }
    }
}
```

Classe principale.

```
public class MainClass {
    public static void main(String[] args)
    {
        ...
        Zoo myZoo = new Zoo("Belvidere", "Tunis",6);
        Animal lion = new Animal("Félin", "Lion", 5, true);
        Animal chat = new Animal("Carnivore","Chat",2,true);
        Animal elephant = new Animal("Elephant", "Elephant", 10, true);
        Animal giraffe = new Animal("Giraffidae", "Giraffe", 8, true);
        myZoo.addAnimal(lion);
        myZoo.addAnimal(chat);
        myZoo.addAnimal(elephant);
        myZoo.addAnimal(giraffe);
        ...
        myZoo.displayAnimalZoo();
        //System.out.println(myZoo);
        //System.out.println(myZoo.toString());
    }
}
```

```
public class Zoo {
    ...
    public int searchAnimal(Animal animal) {
        for (int i=0;i<nbrCages;i++)
        {
            if (animals[i]!=null && animals[i].name==animal.name)
                return i;
        }
        return -1;
    }
}
```

```
public class MainClass {
    public static void main(String[] args)
    {
        ...
        Zoo myZoo = new Zoo("Belvidere", "Tunis",6);
        Animal lion = new Animal("Félin", "Lion", 5, true);
        Animal chat = new Animal("Carnivore","Chat",2,true);
        Animal elephant = new Animal("Elephant", "Elephant", 10, true);
        Animal giraffe = new Animal("Giraffidae", "Giraffe", 8, true);
        Animal chat_n = new Animal("Carnivore","Chat_n",2,true);
        myZoo.addAnimal(lion);
        myZoo.addAnimal(chat);
        myZoo.addAnimal(elephant);
        myZoo.addAnimal(giraffe);
        myZoo.displayZoo();
        myZoo.displayAnimalZoo();
        int search= myZoo.searchAnimal(chat_n);
        if (search != -1) {
            System.out.println("L'animal a été trouvé à l'indice : " + search);
        } else {
            System.out.println("L'animal n'a pas été trouvé dans le zoo.");
        }
        ...
    }
}
```

```
C:\Users\Ferdaws\.jdk\jbr-17.0.9\bin\java.exe "-javaagent:C:\Program Files\JetB
Zoo: Belvidere, cité: Tunis, nombre d'animaux: 4, nombre de cages: 6
animal[0]=Family: Félin ,name:Lion, age:5 ,isMammal:true
animal[1]=Family: Carnivore ,name:Chat, age:2 ,isMammal:true
animal[2]=Family: Elephant ,name:Elephant, age:10 ,isMammal:true
animal[3]=Family: Giraffidae ,name:Giraffe, age:8 ,isMammal:true
L'animal n'a pas été trouvé dans le zoo.
```

Process finished with exit code 0

```
public class MainClass {
    public static void main(String[] args)
    {
        ...
        Zoo myZoo = new Zoo("Belvidere", "Tunis",6);
        Animal lion = new Animal("Félin", "Lion", 5, true);
        Animal chat = new Animal("Carnivore", "Chat", 2, true);
        Animal elephant = new Animal("Elephant", "Elephant", 10, true);
        Animal giraffe = new Animal("Giraffidae", "Giraffe", 8, true);
        Animal giraffe_x = new Animal("Giraffidae", "Giraffe", 8, true);
        Animal chat_n = new Animal("Carnivore", "Chat_n", 2, true);

        myZoo.addAnimal(lion);
        myZoo.addAnimal(chat);
        myZoo.addAnimal(elephant);
        myZoo.addAnimal(giraffe);
        myZoo.addAnimal(giraffe_x);

        myZoo.displayZoo();
        myZoo.displayAnimalZoo();
        int search= myZoo.searchAnimal(giraffe_x);
        if (search != -1) {
            System.out.println("L'animal a été trouvé à l'indice : " + search);
        } else {
            System.out.println("L'animal n'a pas été trouvé dans le zoo.");
        }
        ...
    }
}
```

```
C:\Users\Ferdaws\.jdk\jbr-17.0.9\bin\java.exe "-javaagent:C:\Program Files\JetB
Zoo: Belvidere, cité: Tunis, nombre d'animaux: 5, nombre de cages: 6
animal[0]=Family: Félin ,name:Lion, age:5 ,isMammal:true
animal[1]=Family: Carnivore ,name:Chat, age:2 ,isMammal:true
animal[2]=Family: Elephant ,name:Elephant, age:10 ,isMammal:true
animal[3]=Family: Giraffidae ,name:Giraffe, age:8 ,isMammal:true
animal[4]=Family: Giraffidae ,name:Giraffe, age:8 ,isMammal:true
L'animal a été trouvé à l'indice : 3
```

Process finished with exit code 0

Lorsque on crée un autre animal identique au premier et que vous le cherchez dans le zoo en utilisant la méthode searchAnimal(), on constate que la méthode retournera l'indice du premier animal trouvé qui correspond au critère de recherche (dans ce cas, le nom de l'animal). Cela signifie que la méthode searchAnimal() ne distingue pas entre deux animaux ayant les mêmes attributs.

Elle retournera simplement le premier animal trouvé qui correspond au critère de recherche.

Instruction 12 :

```
public class Zoo {
    ...
    int nbTotal;
    ...
    public boolean addAnimal(Animal animal) {
        if (searchAnimal(animal) == -1) {
            if (isZooFull()==false) {
                animals[nbAnimaux] = animal;
                nbAnimaux++;
                nbTotal++;
                return true;
            } else {
                System.out.println("Nombre maximal");
                return false;
            }
        } else {
            System.out.println("Animal existe déjà");
            return false;
        }
    }
    public boolean isZooFull(){
        /*    if(nbAnimaux < NB_CAGES)
            return false;
            return true;

        */
        return (nbAnimaux == NB_CAGES);
    }
    ...
    public int searchAnimal(Animal animal) {

        for (int i=0;i<nbrCages;i++)
        {
            if (animals[i]!=null && animals[i].name==animal.name)
                return i;
        }

        return -1;
    }
}
```

Instruction 13 :

```
public boolean removeAnimal(Animal animal){
    int pos = searchAnimal(animal);
    if(pos != -1){
        for(int j=pos; j<nbAnimaux-1 ; j++){
            animals[j] = animals[j+1];
        }
        nbAnimaux--;
        animals[nbAnimaux]=null;
        System.out.println("delete avec succes");
        return true;
    }
    return false;
}
```

```

Zoo: Belvidere, cité: Tunis, nombre d'animaux: 4, nombre de cages: 6
animal[0]=Family: Félin ,name:Lion, age:5 ,isMammal:true
animal[1]=Family: Carnivore ,name:Chat, age:2 ,isMammal:true
animal[2]=Family: Elephant ,name:Elephant, age:10 ,isMammal:true
animal[3]=Family: Giraffe ,name:Giraffe, age:8 ,isMammal:true
Animal existe déjà
L'animal a été trouvé à l'indice : 1
delete avec succes
```