

# OOP

Comprendre la Programmation  
Orientée Objet





**Compétence demandée :**  
**Comprendre ce qu'est une classe et  
les 4 principes de la POO**

1. Les classes et les objets
2. L'instanciation
3. L'abstraction
4. L'héritage
5. Le polymorphisme
6. L'encapsulation



# LES CLASSES ET LES OBJETS



# La définition d'un objet Versus L'objet concret



# La **classe** d'un objet Versus L'**objet**

Une **classe regroupe** des membres,  
communs à un ensemble d'objets.

Ces membres peuvent être des  
méthodes ou des propriétés

Les **propriétés** définissent les **caractéristiques** d'un ensemble d'objets

Les **méthodes** définissent les **comportements** d'un ensemble d'objets



# Exemples en programmation

Les classes peuvent aussi définir des propriétés

```
class Produit  
{  
    public $id;  
    public $titre;  
    public $adresse;  
    public $ville;  
    public $cp;  
    public $surface;  
    public $prix;  
    public $photo;  
    public $type;  
    public $description;  
}
```

Logement

Les classes peuvent aussi définir des **méthodes**

```
class Vendeur {  
    public $nom;  
    public $prenom;  
  
    public function vendre(Produit $produit): bool {  
        echo "Je vends le produit : " . $produit->titre;  
        return true;  
    }  
}
```

## Les classes peuvent être **abstraites**

```
class Produit extends ElementVendable  
{  
    public $id;  
    public $titre;  
    public $adresse;  
    public $ville;  
    public $cp;  
    public $surface;  
    public $photo;  
    public $type;  
    public $description;  
}
```

```
abstract class ElementVendable {  
    public $prix;  
}
```





# Exemples en UML

## CHIEN

- + age
- + groupe sanguin
- + taille
- + poids

- + mange()
- + dort()
- + aboie()



Une **interface** permet de  
lister les fonctionnalités  
attendues d'une brique

On dit qu'un objet  
**implémente** une  
**interface**

On dit qu'un objet  
**implémente** une  
**interface**



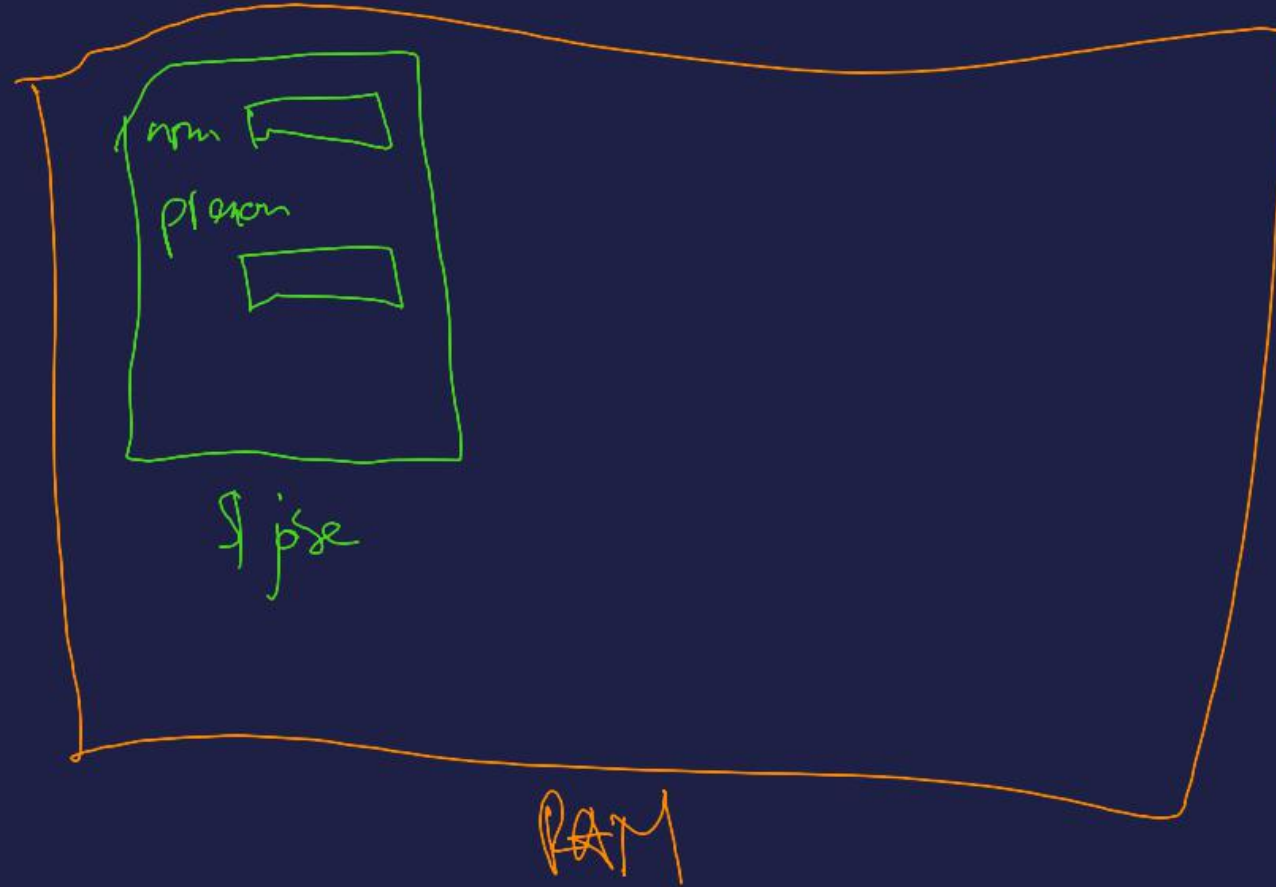




# L'INSTANCIATION

**Et dans un ordinateur ?**

**Que se passe-t-il dans la RAM ?**



créer un objet  
instancier

↓  
\$pse = new Person;

nom  
prenom

Instantiation



**Sinon** ... on s'est  
fiche ..

# **Touche de clavier d'ordinateur**

# Une batterie d'ordinateur

# **Une souris d'ordinateur**



# Un vidéo-projecteur

# Des hauts-parleurs

# Un microphone

# Une pédale de frein



# **Abstraction**

**Utiliser des briques sans  
connaître leurs détails  
techniques**

# Abstraction

Utiliser des briques sans  
connaître leurs détails  
techniques







# L'HERITAGE

## CHIEN

- + age
- + groupe sanguin
- + taille
- + poids

- + mange()
- + dort()
- + aboie()

## CHAT

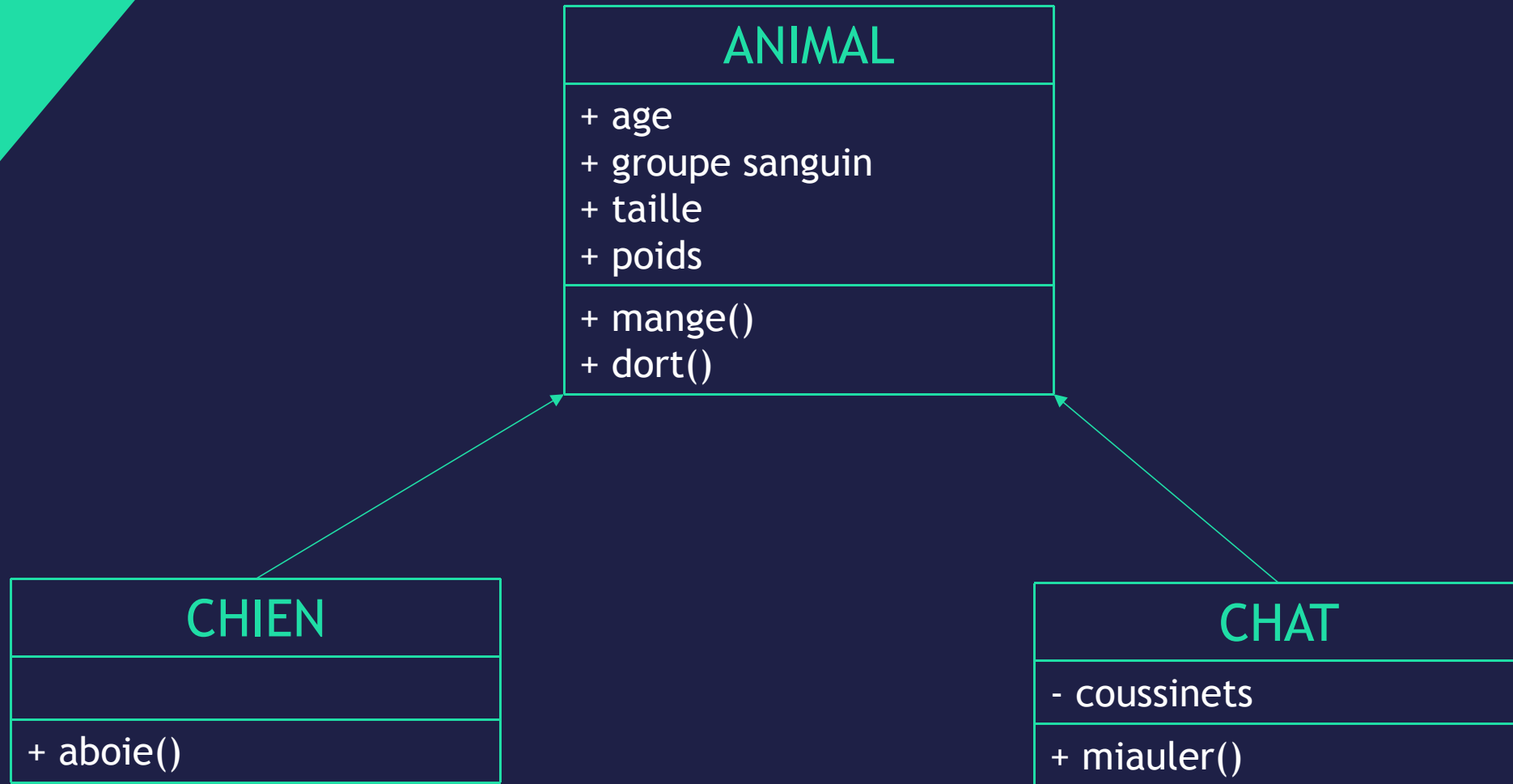
- Coussinets
- + age
- + groupe sanguin
- + taille
- + poids

- + mange()
- + dort()
- + miauler()

| CHIEN     |
|-----------|
|           |
| + aboie() |

| ???  |
|--|
| + age<br>- groupe sanguin<br># taille<br>+ poids |
| + mange()<br>+ dort()                            |

| CHAT         |
|--------------|
| - coussinets |
| + miauler()  |



L'héritage permet d'organiser les classes en groupe plus spécifiques qui ont des propriétés et des méthodes qui leur sont propres

## L'héritage se fait avec le mot-clé **extends**

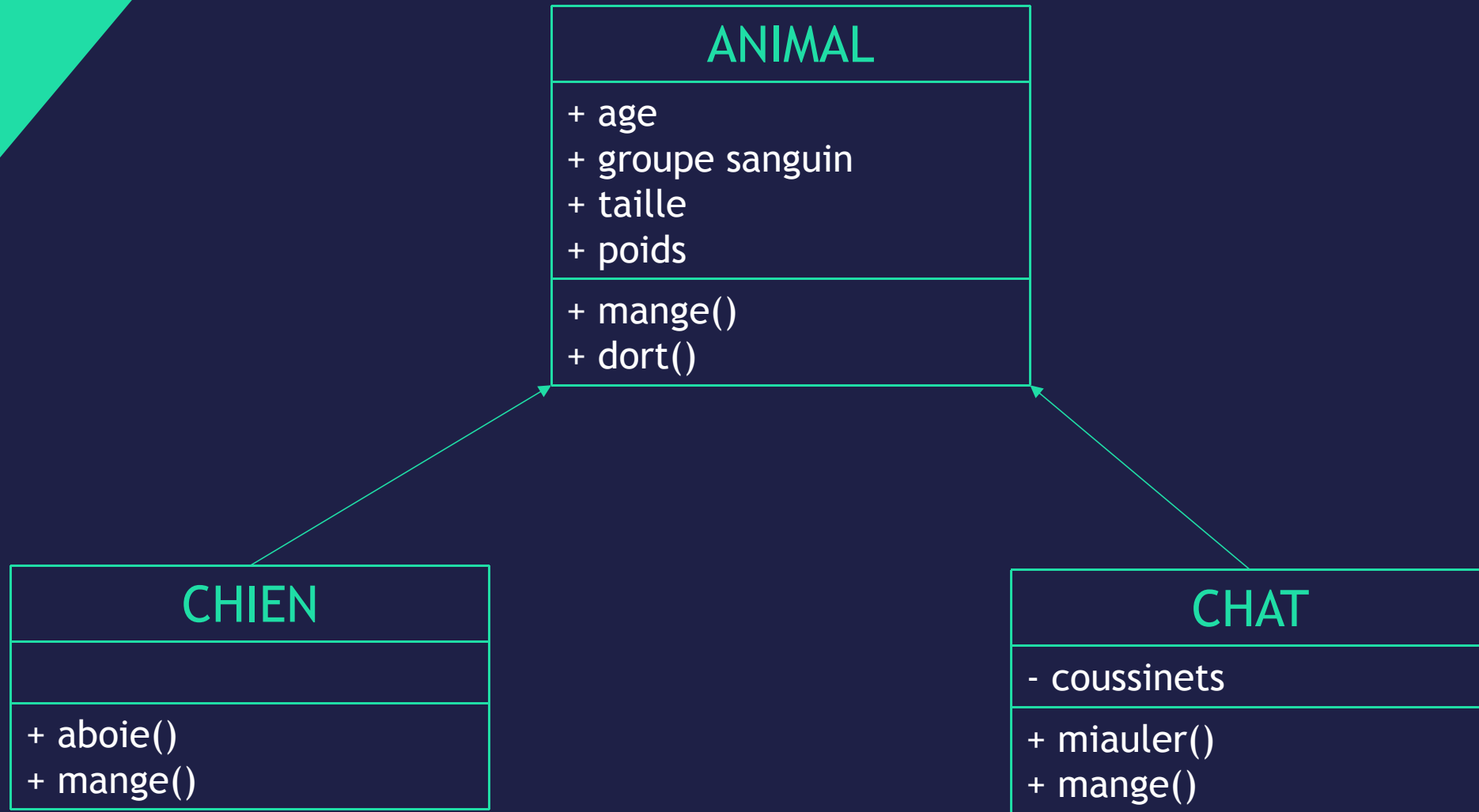
```
class Produit extends ElementVendable  
{  
    public $id;  
    public $titre;  
    public $adresse;  
    public $ville;  
    public $cp;  
    public $surface;  
    public $photo;  
    public $type;  
    public $description;  
}
```

```
abstract class ElementVendable {  
    public $prix;  
}
```



# LE POLYMORPHISME

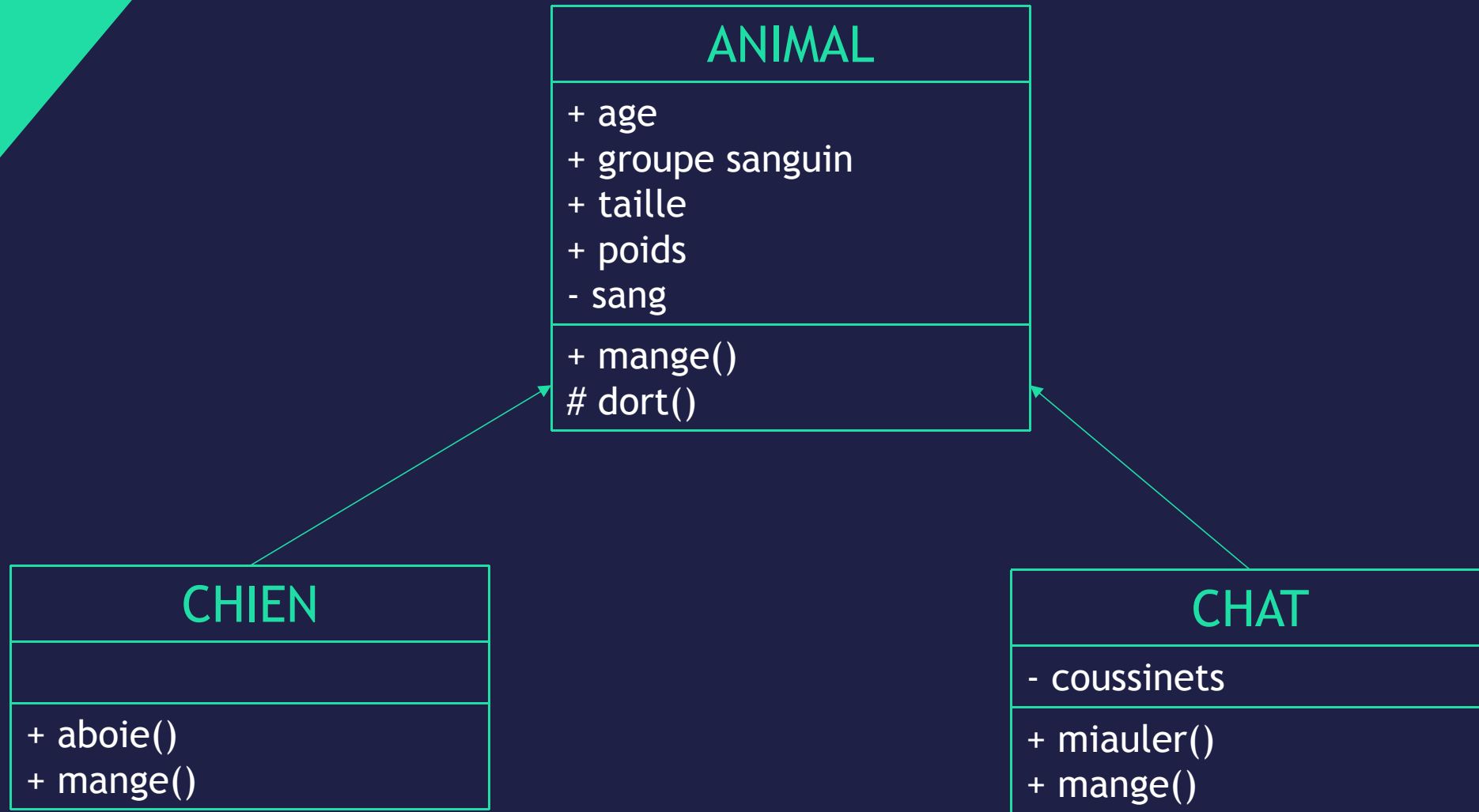




Le polymorphisme (de méthode)  
permet d'utiliser le même  
verbe pour décrire 2 méthodes  
différentes



# L'ENCAPSULATION



L'encapsulation permet de définir une propriété ou une méthode **interne**

Privé : interne à la classe

Protégé : interne à la famille

L'encapsulation se fait avec les mots-clés **public**, **protected** ou **private**

```
class Homme
{
    protected function pleurer()
    {
    }
}

class HommeFier extends Homme
{
    private function seRemettreEnQuestion() {
    }
}
```

# 4 PRINCIPES

## ABSTRACTION



# 4 PRINCIPES

**ABSTRACTION  
HERITAGE**

# 4 PRINCIPES

**ABSTRACTION  
HERITAGE  
POLYMORPHISME**

# 4 PRINCIPES

**ABSTRACTION  
HERITAGE  
POLYMORPHISME  
ENCAPSULATION**

# 4 PRINCIPES

**ABSTRACTION  
HERITAGE  
POLYMORPHISME  
ENCAPSULATION**

# 4 PRINCIPES

**ABSTRACTION  
HERITAGE  
POLYMORPHISME  
ENCAPSULATION**



