



Concepts objets

Objectifs Pédagogiques

À l'issue de cette formation, vous serez en mesure de :

- ✓ Comprendre les notions de base de la Programmation Orientée Objet, à savoir ce que sont:
 - ✓ *Les méthodes abstraites*
 - ✓ *La redéfinition*
 - ✓ *Les attributs d'instance et les attributs de classe*

Programme détaillé ou sommaire

Langage et abstraction

Héritage

Comportement

Méthode abstraite

Redéfinition

Attributs

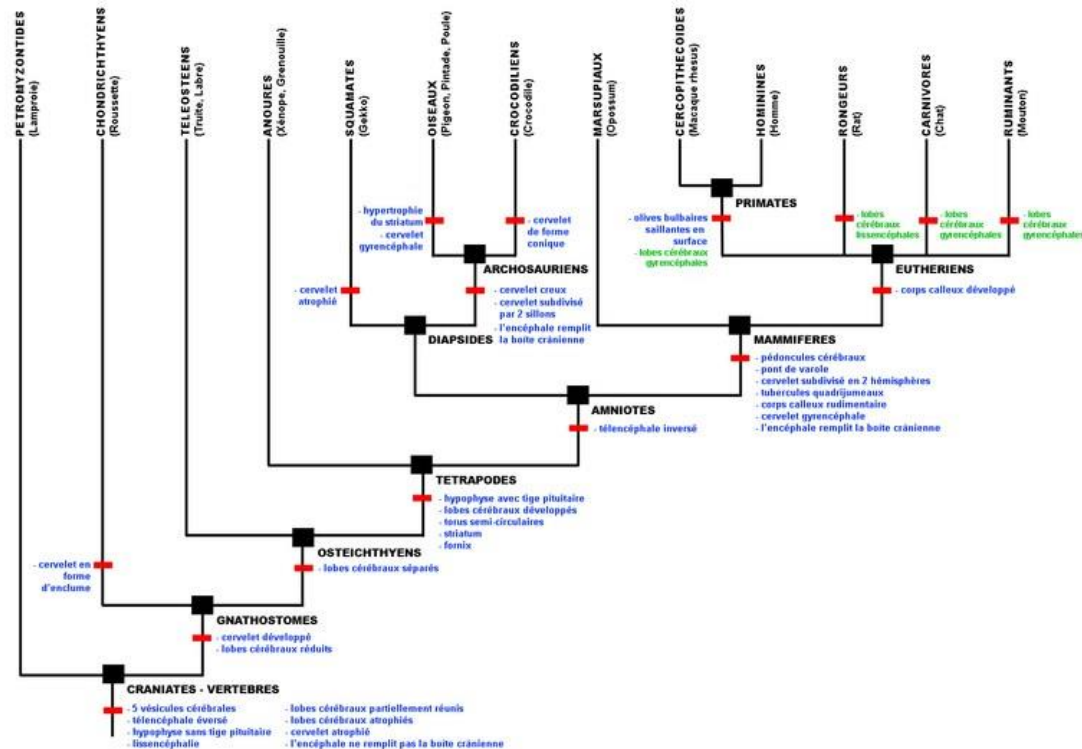
Les concepts abstraits

Un concept abstrait ne représente pas (forcément) une chose physique.

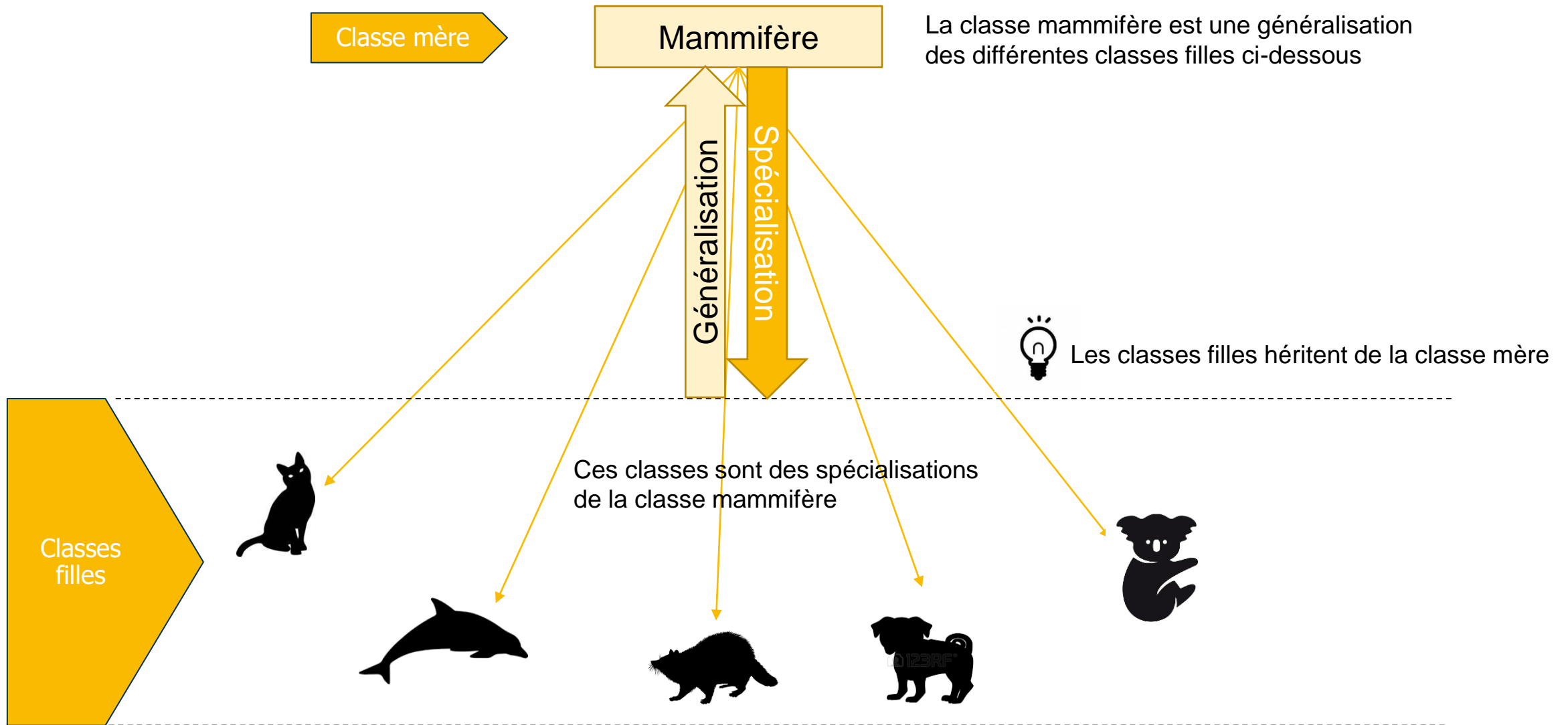
Exemple: le concept de Mammifère désigne une famille d'animaux qui ont des traits et caractéristiques communes.

Un mammifère concret n'existe pas. C'est un concept inventé par les biologistes.

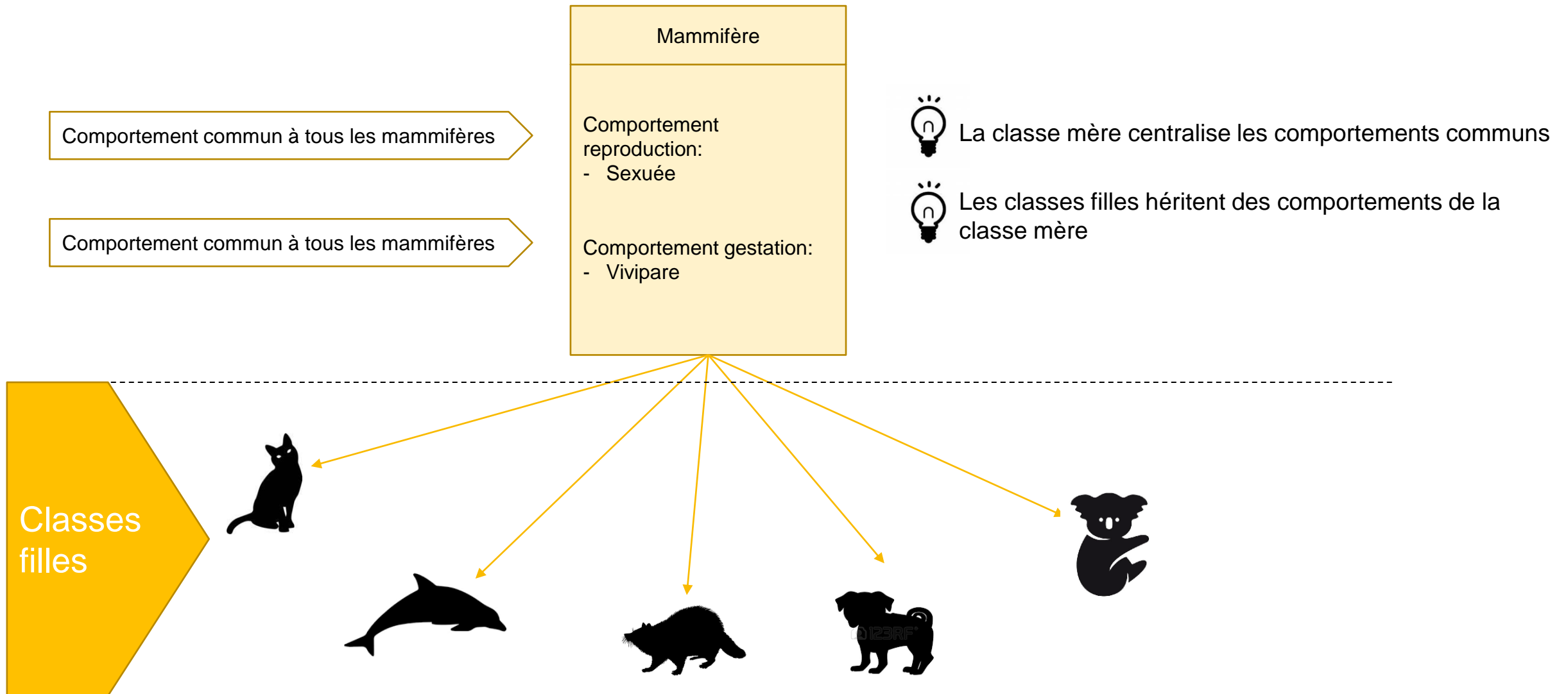
Arbre phylogénétique



HERITAGE



HERITAGE: Les comportements



HERITAGE: méthode abstraite

Classe mère

Mammifère

Comportement reproduction:

- Sexuée

Comportement gestation:

- Vivipare

Comportement alimentaire:

- ?

On ne peut rien dire de ce comportement.

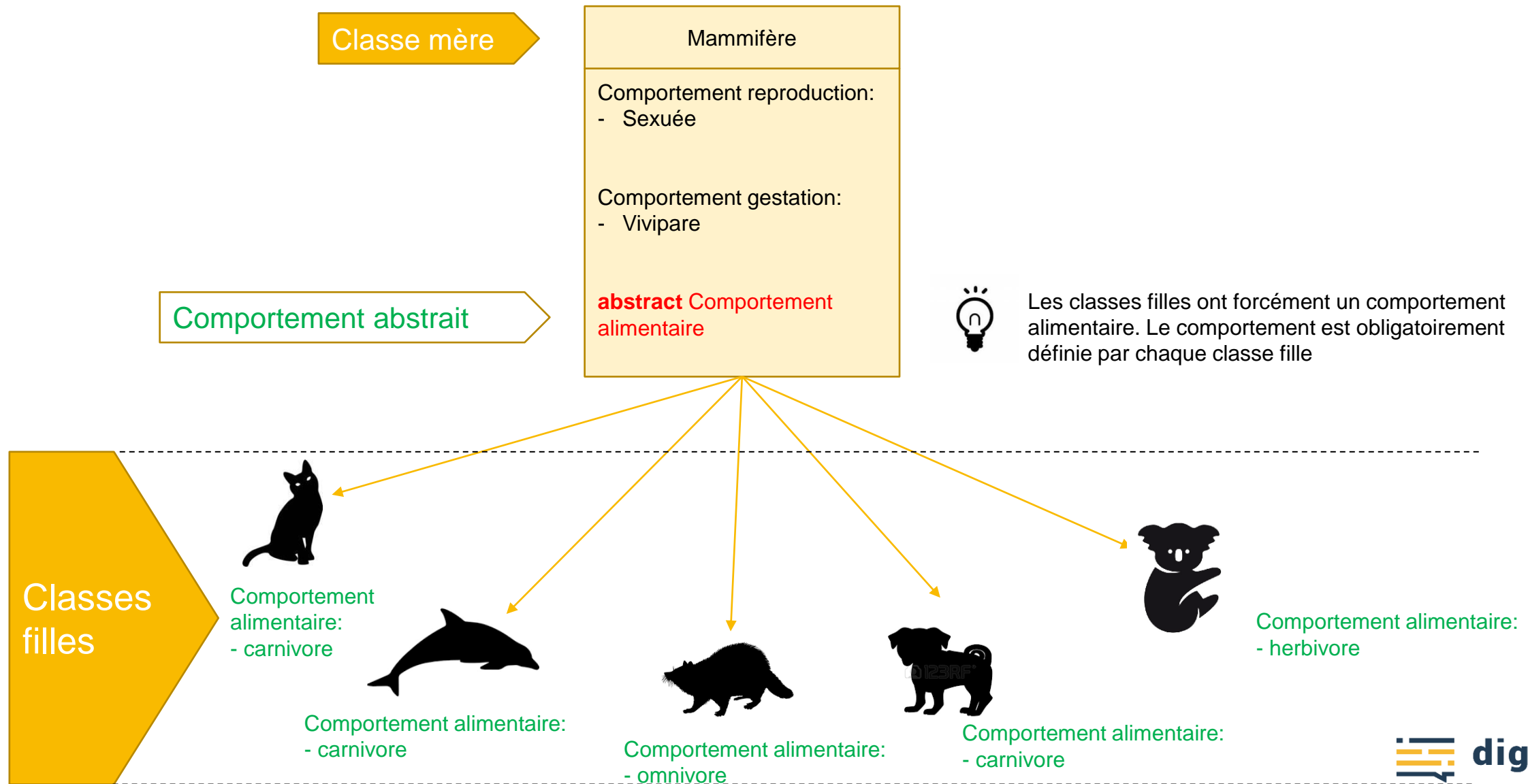


Un comportement est dit abstrait lorsqu'on sait qu'il existe mais qu'on ne peut pas le décrire dans la classe mère

Classes filles



HERITAGE: méthode abstraite



HERITAGE: méthode abstraite

Classe mère

abstract Mammifère

Comportement reproduction:

- Sexuée

Comportement gestation:

- Vivipare

abstract Comportement
alimentaire



Une classe est dite abstraite si elle possède au moins une méthode abstraite

Classes
filles



Comportement
alimentaire:
- carnivore



Comportement alimentaire:
- carnivore



Comportement alimentaire:
- omnivore

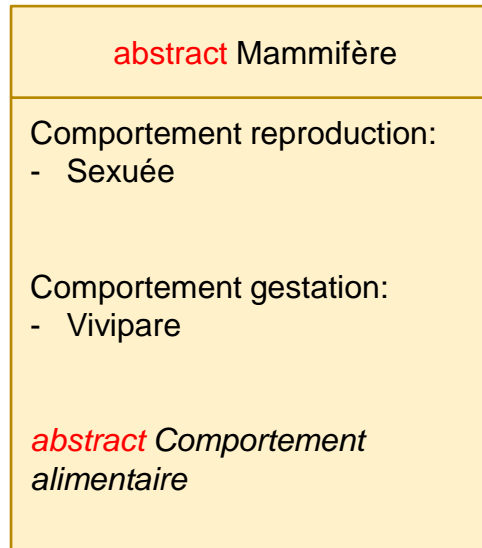


Comportement alimentaire:
- carnivore



Comportement alimentaire:
- herbivore

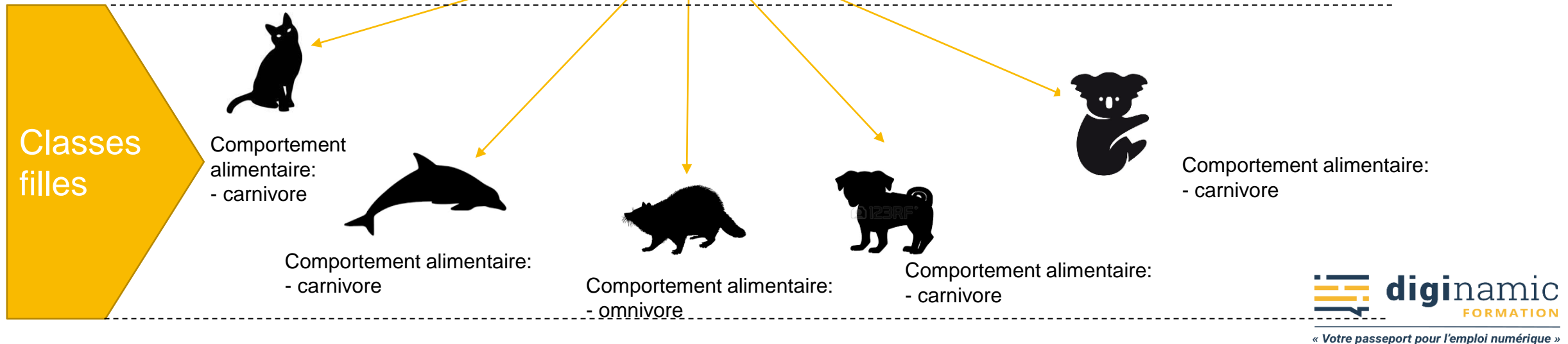
HERITAGE: méthode abstraite



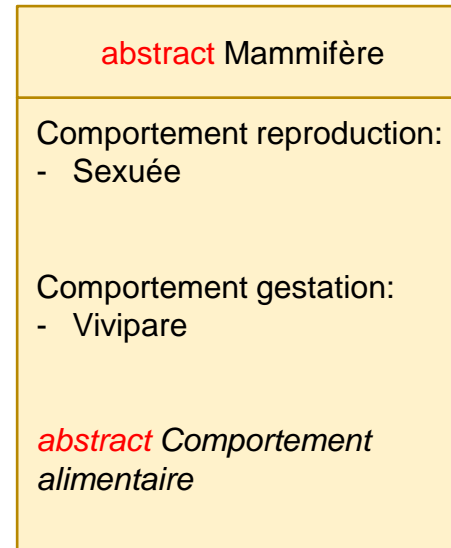
On ne peut pas créer d'objet à partir d'une classe abstraite. Dans la nature il n'existe pas de mammifère **concret**.



Une classe abstraite représente un concept abstrait.



HERITAGE: redéfinition



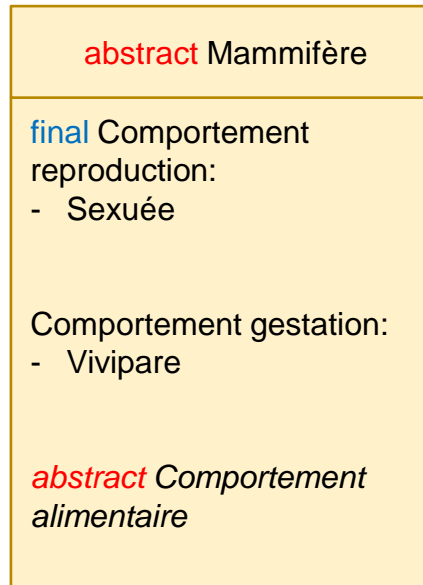
La classe fille Ornithorinque est une exception. C'est le seul mammifère à pondre des œufs. On va donc redéfinir le comportement de gestation par défaut.

Classe
s filles



Comportement gestation:
- Ovipare

HERITAGE: redéfinition



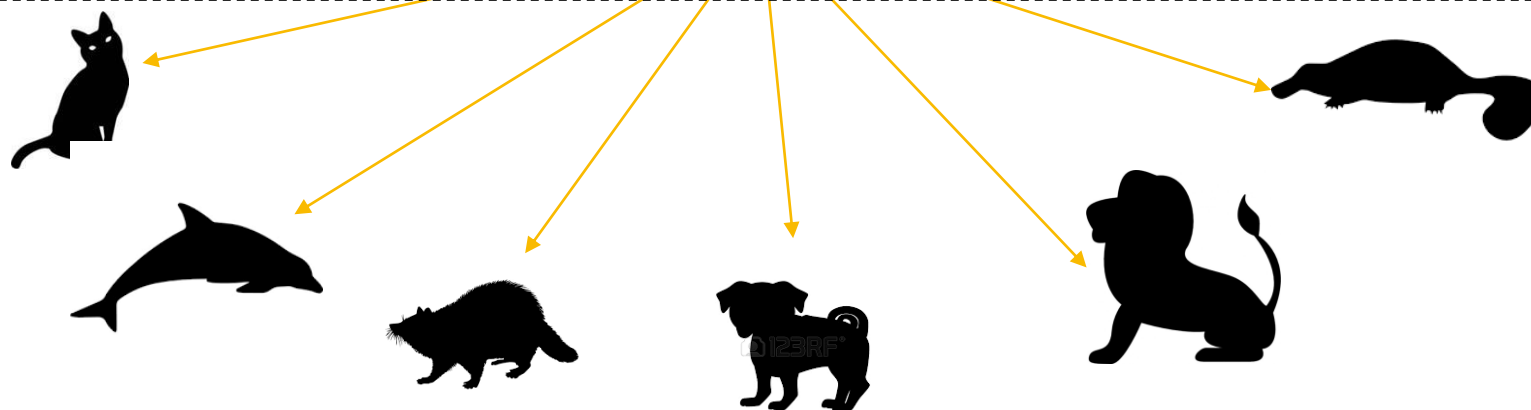
Une méthode finale est une méthode qui ne peut pas être redéfinie par une classe fille.



Tous les mammifères se reproduisent de manière sexuée, sans exception



Cette méthode peut être déclarée « finale »
Cela permet d'affiner le modèle.



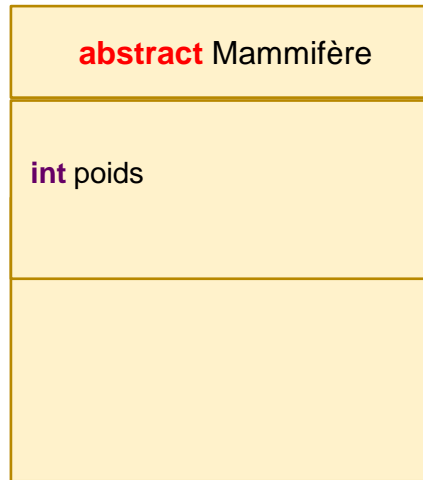
ATTRIBUTS: Les caractéristiques d'un objet...

Un attribut est un élément de la structure de la classe qui porte une valeur.

Attribut d'instance vs attribut de classe :

- 2 chats auront un **poids différent**: le poids est un **attribut d'instance**

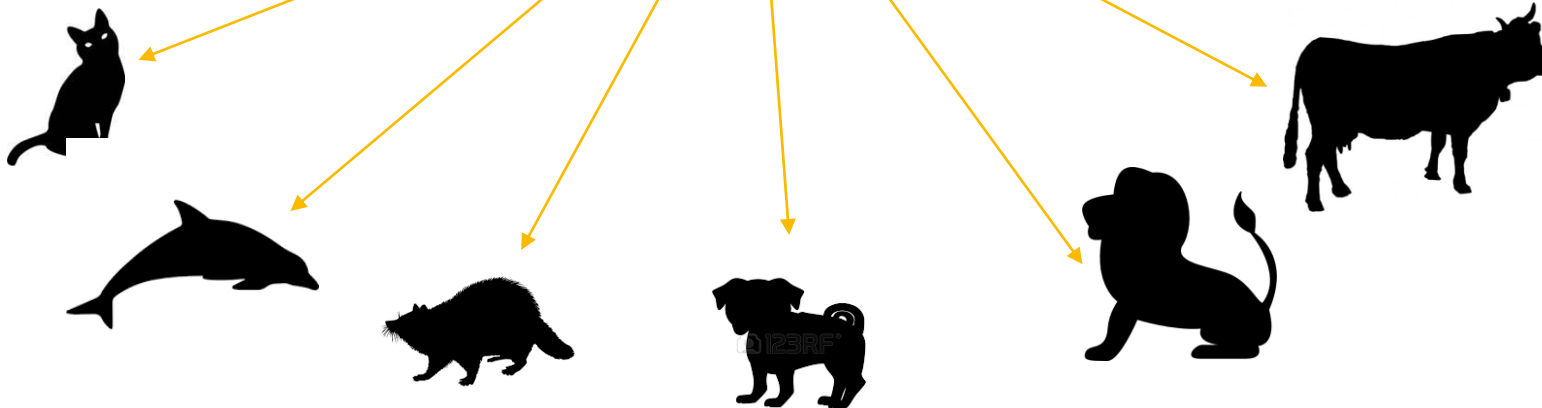
- 2 chats auront la **même durée de gestation**: la durée de gestation est un **attribut de classe** (**static**), partagée par tous les chats.



Tous les mammifères ont un poids, on peut donc déclarer cet attribut au niveau de la classe mère. En revanche la durée de gestation est spécifique à chaque classe fille



Chaque classe fille disposera de cet attribut par héritage.



HERITAGE

```
/** Classe mammifère
 * @author DIGINAMIC
 *
 */
public abstract class Mammifere {

    /**
     * Affiche le comportement alimentaire du mammifère
     */
    public abstract void seNourrir();

}
```

```
/** Représente un koala
 * @author DIGINAMIC
 *
 */
public class Koala extends Mammifere {

    @Override
    public void seNourrir() {
        System.out.println("Le Koala mange des fruits.");
    }

}
```

```
/** Représente un chat
 * @author DIGINAMIC
 *
 */
public class Chat extends Mammifere {

    @Override
    public void seNourrir() {
        System.out.println("Le chat est carnivore.");
    }

    /**
     * Le chat miaule
     */
    public void miauler(){
        System.out.println("Miaou");
    }

}
```

X monInstance = new X();
Services.afficherComportementAlimentaire(monInstance);

Qu'est ce que la classe X?

```
/** Classe de services
 * @author DIGINAMIC
 *
 */
public class Services {

    /** Affiche le comportement alimentaire d'un mammifère
     * @param mammifere
     */
    public void afficherComportementAlimentaire(Mammifere mammifere){
        mammifere.seNourrir();
    }

}
```

INTERFACES

```
/** Représente un chasseur.  
 * @author DIGINAMIC  
 *  
 */  
public interface Chasseur {  
  
    /**  
     * Méthode de chasse  
     */  
    void chasser();  
}
```

```
public class Chouette implements Chasseur {  
  
    @Override  
    public void chasser() {  
        System.out.println("Chasseur nocturne");  
    }  
}
```

```
/** Représente un chat  
 * @author DIGINAMIC  
 *  
 */  
public class Chat extends Mammifere implements Chasseur {  
  
    @Override  
    public void seNourrir() {  
        System.out.println("Le chat est carnivore.");  
    }  
  
    /**  
     * Le chat miaule  
     */  
    public void miauler(){  
        System.out.println("Miaou");  
    }  
  
    @Override  
    public void chasser() {  
        System.out.println("Chasseur diurne");  
    }  
}
```

X monInstance = new X();
Services.afficherComportementChasse(monInstance);

Qu'est ce que la classe X?

```
/** Classe de services  
 * @author DIGINAMIC  
 *  
 */  
public class Services {  
  
    /** Affiche le comportement alimentaire d'un mammifere  
     * @param mammifere  
     */  
    public void afficherComportementAlimentaire(Mammifere mammifere){  
        mammifere.seNourrir();  
    }  
  
    /** Affiche le comportement alimentaire d'un mammifere  
     * @param mammifere  
     */  
    public void afficherComportementChasse(Chasseur chasseur){  
        chasseur.chasser();  
    }  
}
```