



Une présentation pour le Collège de Rosemont

Héritage

Hugo Leblanc
25 Septembre 2019

Héritage

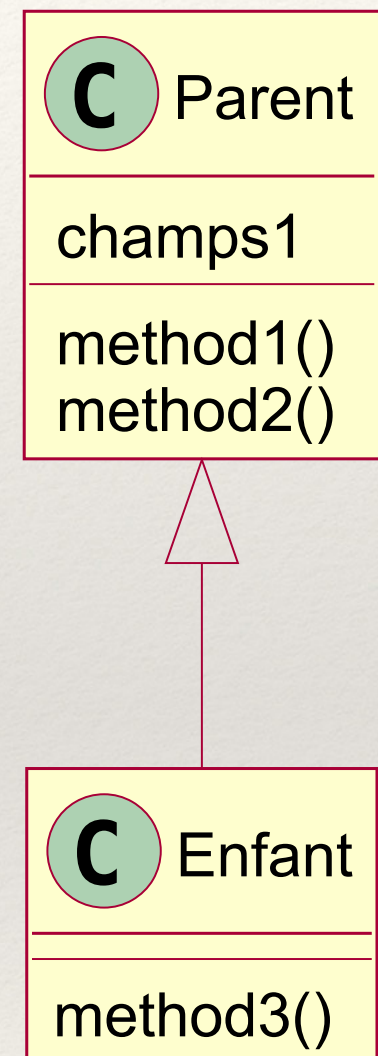
- ❖ Description
- ❖ Syntaxe
- ❖ Avantages

Héritage

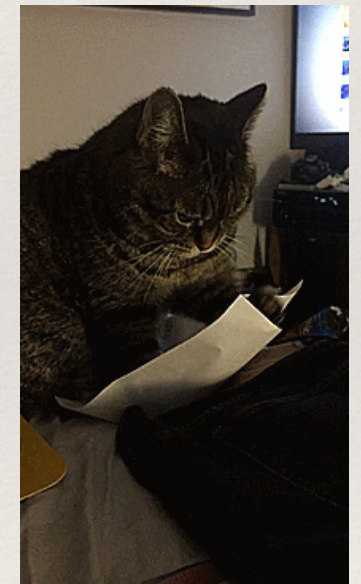
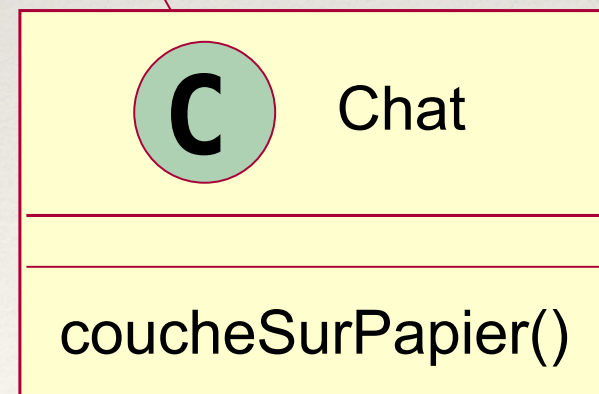
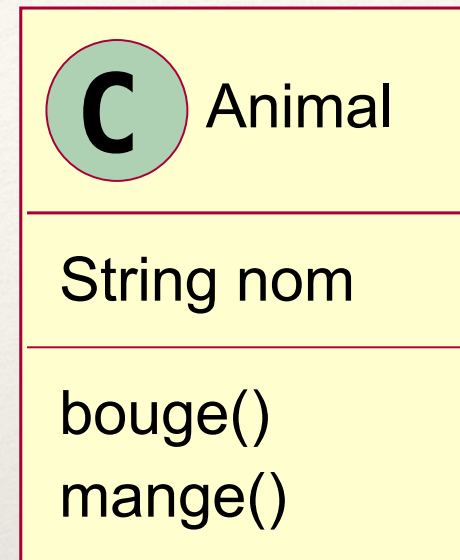
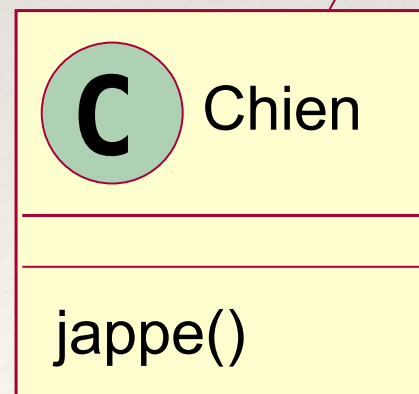
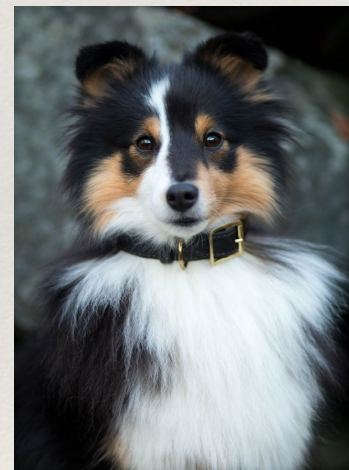
- ❖ **Description**
- ❖ Syntaxe
- ❖ Avantages

Description

- ❖ Le mécanisme d'héritage nous permet de “copier” les propriétés (champs, méthodes, etc.) d'une classe dans une autre.
- ❖ On parle d'une classe **parent** et une classe **enfant**. La classe Enfant **spécialise** la classe Parent.



Exemple



Héritage

- ❖ Description
- ❖ Syntaxe
- ❖ Avantages

Syntaxe

```
public class Animal{  
    String nom;  
    String couleur;  
  
    public void bouge () {  
        // Animal bouge  
    }  
  
    public void mange () {  
        // Animal mange  
    }  
}
```

```
public class Chien extends Animal{  
    public void jappe () {  
        // Chien jappe  
    }  
}
```

```
public class Chat extends Animal{  
    public void coucheSurPapier () {  
        // Chat se couche  
    }  
}
```

Héritage

- ❖ Description
- ❖ Syntaxe
- ❖ **Avantages**

Avantages

- ❖ Réutilisation de code et maintenance
 - ❖ Un seul endroit pour l'implémentation de propriétés
- ❖ Sous-typage
 - ❖ Le sous-typage permet d'écrire des méthodes qui accepteront une classe générale et ainsi toutes ses classes spécialisées.

Example

```
public class Main{  
    public static void capturerAnimal (Animal animal) {  
        //Capture l'animal  
    }  
  
    public static void main (String[] args) {  
        Chat chat = new Chat();  
        capturerAnimal (chat); // Permis!  
    }  
}
```

Dans la prochaine séance

- ❖ Contraintes
- ❖ Modificateurs d'accès (public, private, etc.)
- ❖ Héritage à plusieurs niveaux
- ❖ Composition