

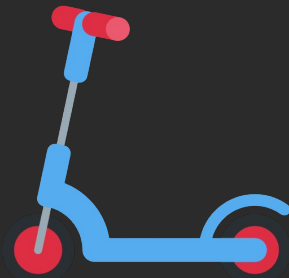
Programmation Orientée Objet

Héritage



Une histoire de famille

- Un **objet parent** peut transmettre des caractéristiques à un **objet enfant**
- L'objet enfant **héríte** / **dérive** de l'objet parent



Héritage en Kotlin

- Par défaut, aucune classe ne peut être dérivée
- Il faut marquer la classe comme “ouverte à l’héritage”
- La classe enfant explicite son lien avec la classe parent
- Pour une classe enfant, une seule classe parente autorisée

Syntaxe d'une classe parente

```
open class Parent
```

Syntaxe d'une classe enfant

```
open class Parent
```

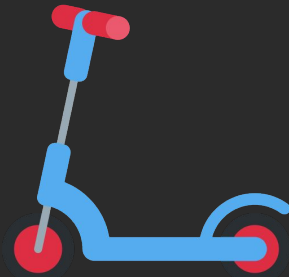
```
class Child : Parent()
```

Appel d'un constructeur parent

```
open class Parent(val var1: Int)  
  
class Child : Parent(4)
```

Polymorphisme

- Hériter d'une action parent, mais la faire différemment
- La méthode a donc plusieurs formes (elle est polymorphe)
- Si une classe enfant modifie une action, elle *surcharge la méthode*



Polymorphisme en Kotlin

- Par défaut, aucune méthode ne peut être surchargée
- Il faut marquer la méthode comme “ouverte à la surcharge”
- La méthode enfant explicite la surcharge avec la méthode parente

Syntaxe de surcharge de méthodes

```
open class Parent {  
    open fun myMethod() {  
        print("Hello from parent")  
    }  
}  
  
class Child : Parent() {  
    override fun myMethod() {  
        println("Hello from child")  
    }  
}
```