



# Analyse et Programmation Orientées Objets / C++

## La relation d'héritage

# Classification par héritage (1)

---

## ➡ Origine et justification

- Principes de la classification en S.V.T.
- Factorisation hiérarchique des caractéristiques
- Arbre n-aire de classification
- Exemples :
  - Invertébrés (pas de colonne vertébrale)
  - Arthropodes (corps dur et symétrie axiale)
  - Insectes (1 paire antennes et 3 paires de pattes)
  - ---

# Classification par héritage (2)

---

## ➡ Objectifs recherchés pour la relation "**is\_a**"

- Favoriser la réutilisation de codes sources
- Réutilisation des données et des traitements
- Respecter les règles d'encapsulation
- Favoriser la conception par spécialisation progressive
  - Spécialisation par enrichissement
  - Spécialisation par substitution (redéfinition)
- Exemple : IHM Windows

# Classification par héritage (3)

---

## ➡ Vocabulaire attaché à la relation

- Relation exclusivement entre classes
- Classe amont : classe mère
- Classe aval : classe fille (classe dérivée)
- Relation hiérarchique à plusieurs niveaux
- Ensemble des classes aval : famille
- Transitivité restreinte

# La classe mère

---

## ➡ **Attributs et méthodes**

- Attributs strictement privés
- Attributs délégués (dans la descendance)
- Définition d'une nouvelle section de contrôle d'accès
- Toutes les méthodes sont exportées

# Définition de la classe mère

---

## ➡ Langage C++

```
class CX {
```

```
  private :
```

```
  --- Définition des attributs spécifiques à CX
```

```
  protected :
```

```
  --- Définition des attributs délégués
```

```
  public :
```

```
  --- Déclaration des méthodes de CX
```

```
};
```

# La classe fille

---

## ➡ Attributs et méthodes

- Attributs hérités de la classe mère (**private**)
- Attributs spécifiques à la classe fille
- Attributs délégués par la classe mère (**protected**)
- Méthodes hérités de la classe mère
- Méthodes redéfinies
- Méthodes propres à la classe fille

# Définition de la classe fille

---

## ➡ Langage C++

```
class CY : public CX {
```

```
private :
```

```
--- Définition des attributs spécifiques à CY
```

```
public :
```

```
CY (list_types);
```

```
--- Déclaration des méthodes propres à CY
```

```
};
```



# Constructeur d'instances de CY

---

## ➡ Syntaxe en langage C++

```
# include "CX.h "
```

```
# include "CY.h "
```

```
CY::CY (list args ) : CX (sub_list args) {  
    --- Affectation des attributs propres à CY  
}
```