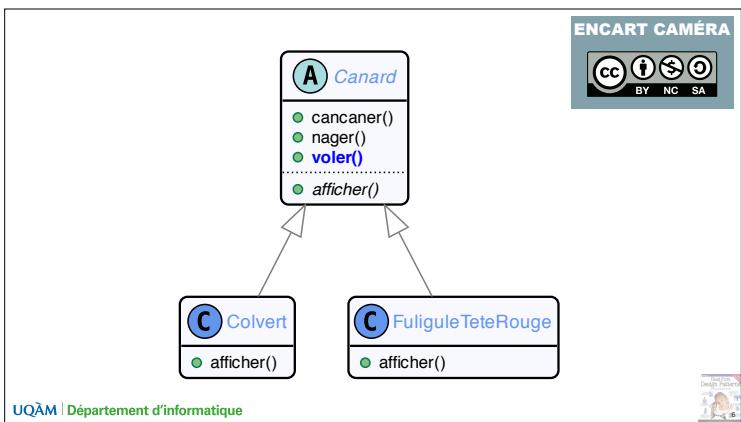
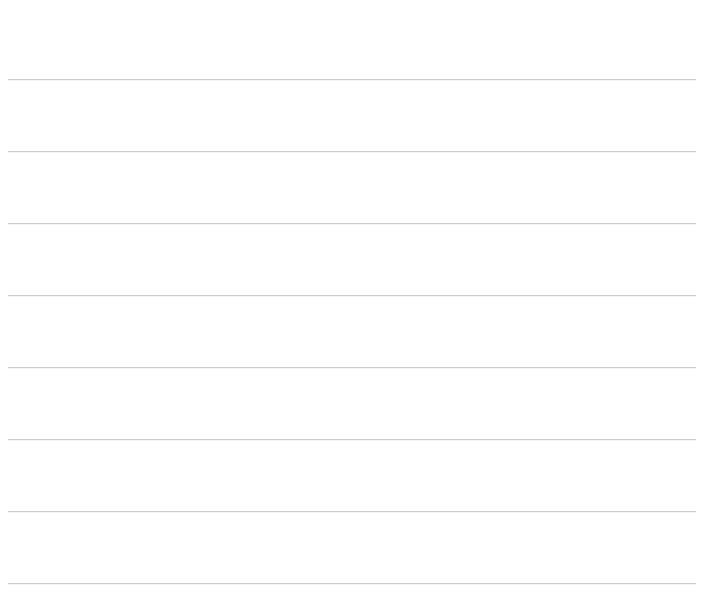
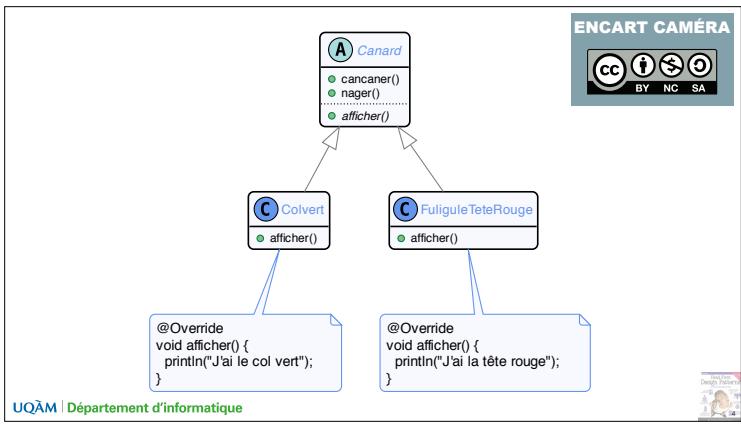


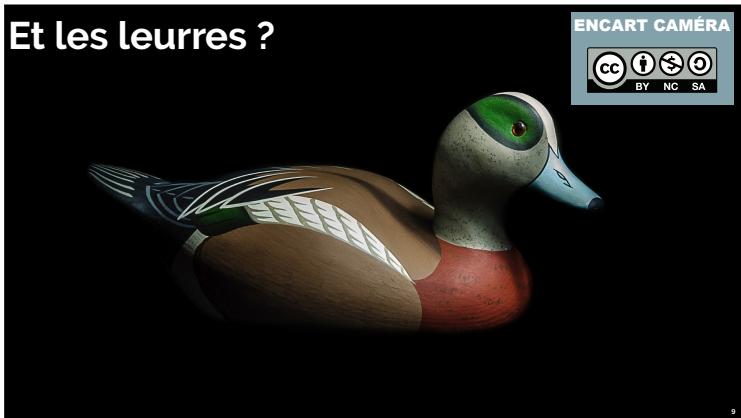
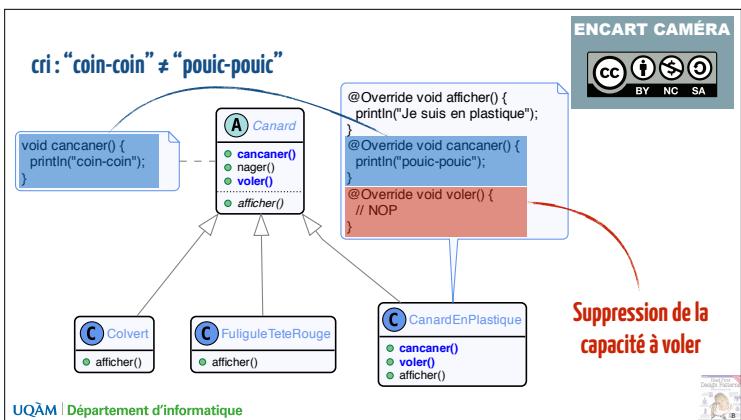


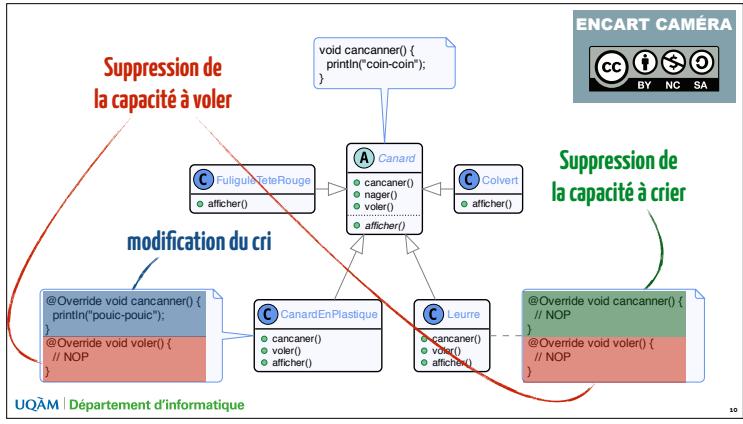
ENCART CAMÉRA



BY NC SA







Est-ce raisonnable ?



Pourtant on respecte tous les principes !

(abstraction, encapsulation, héritage, polymorphisme, ...)

UQÀM | Département d'informatique

11

Travail aux interfaces !



Design Principle

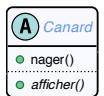
Program to an interface, not an implementation.

UQÀM | Département d'informatique



On identifie trois abstractions

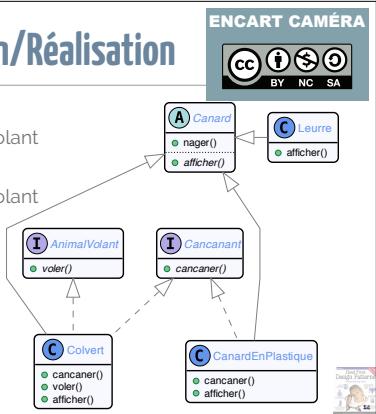
- Une abstraction **structurelle** :
 - Être un canard
- Deux abstraction **fonctionnelles** :
 - Cancaner
 - Voler



13

Définition par Généralisation/Réalisation

- Colvert** :
 - Canard ⊕ Cancanant ⊕ Animal Volant
- Fuligule à tête rouge** :
 - Canard ⊕ Cancanant ⊕ Animal Volant
- Canard en plastique** :
 - Canard ⊕ Cancanant
- Leurre** :
 - Canard



On définit les types de canards en composant les interfaces lors de la compilation

Définition par Généralisation/Réalisation

- Colvert** :
 - Canard ⊕ Cancanant ⊕ Animal Volant
- Fuligule à tête rouge** :
 - Canard ⊕ Cancanant ⊕ Animal Volant
- Canard en plastique** :
 - Canard ⊕ Cancanant
- Leurre** :
 - Canard



```
public class Colvert
    extends Canard
    implements AnimalVolant, Cancanant { ... }

public class CanardEnPlastique
    extends Canard
    implements Cancanant { ... }

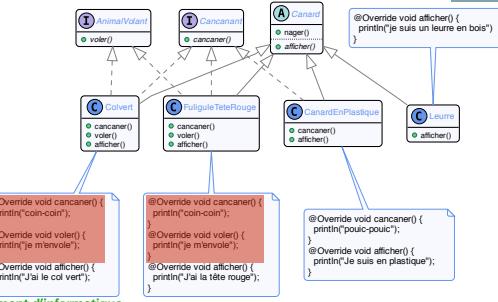
public class Leurre
    extends Canard { ... }
```

15

Au bout du compte ...

DRY:
Don't
Repeat
Yourself

ENCART CAMÉRA

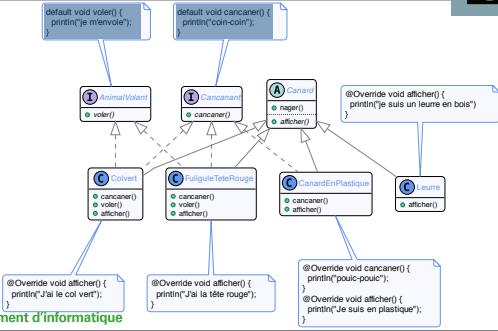


UQÀM | Département d'informatique

16

On peut faire mieux avec des "default method"

ENCART CAMÉRA



UQÀM | Département d'informatique

17

On aurait pas juste décalé le problème ?

ENCART CAMÉRA



```
Colvert colvert = new Colvert();
c.voler();

Canard unCanard = (Canard) new Colvert();
unCanard.voler() // Ne compile pas !!

if (unCanard instanceof AnimalVolant) {
    ((AnimalVolant) unCanard).voler();
}
```

UQÀM | Département d'informatique

18

Conclusions

ENCART CAMÉRA



- Notre conception souffre fondamentalement d'un défaut majeur
 - Pour l'utiliser, on doit reposer sur des **casts** & des **instance of** !
 - **C'est à l'encontre des principes objets'**
- **On peut s'acharner et essayer de la hacker**
- Ou on peut faire levier sur cet échec pour :
 - **caractériser les propriétés** dont on a vraiment besoin
 - **Définir une nouvelle conception**, qui les garantie

UQÀM | Département d'informatique

* pourtant il y a des classes, des interfaces...*



UQÀM | Département d'informatique

FACULTÉ DES SCIENCES
Université du Québec à Montréal

ENCART CAMÉRA



<https://mosser.github.io/>



<https://ace-design.github.io/>

**Abonne toi à la chaîne,
et met un pouce bleu !**

