

Un exemple de conception de cartes

ENCART CAMÉRA



- Description extraite depuis la **spécification** :
 - Il y a **neufs (9)** cartes **clan** par couleurs
 - Il y a **six (6)** couleurs de cartes
- On **encode** l'information dans un **entier** $card \in [0, 9 \times 6[$
 - Pour trouver la **valeur** de la carte : On **divise** par neuf, plus un

```
public static int value(int card) { return (card / 9) + 1; }
```
 - Pour trouver la **couleur** : on prend le **modulo** neuf

```
public static int color(int card) { return card % 9; }
```

“Hé tavu, il est opti mon code...”

Quels sont les problèmes ?

ENCART CAMÉRA



- Représentation** dans le **code différente** du **domaine**
 - Vous jouez avec des cartes, ou avec des nombres entiers ?*
- Représentation **couplée** à l'implémentation
 - Dans tout le programme, on utilise le type int pour une carte*
- On peut **corrompre** la structure de donnée aisément
 - Est-ce que -42 est une carte valide ? Pourtant c'est un int ..*

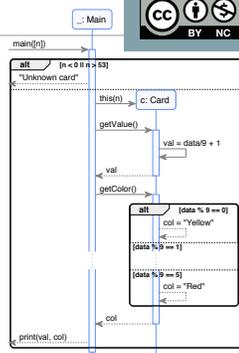
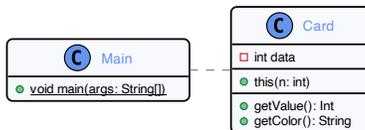
On parle de **"Primitive Obsession"** quand un code utilise de manière inadéquate des types primitifs au lieu d'objets du domaine. C'est une **représentation possible** dans le code, mais **pas acceptable** du point de vue de la conception.

On va encapsuler alors !

ENCART CAMÉRA



Classes : Modèles UML :
C'est facile la conception !

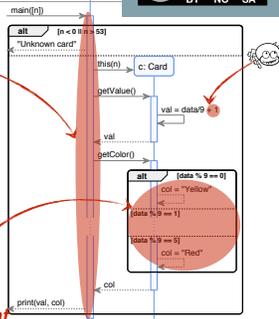
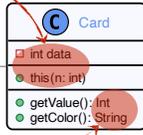


Quels sont les problèmes?



Choix interne visible

Responsabilité ?



Primitive Obsession

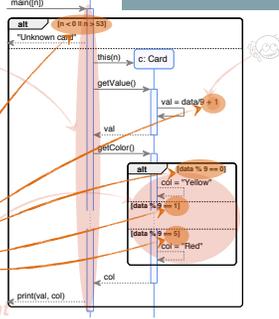
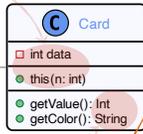
Switch statement

Un dernier problème pour la route



Choix interne visible

Responsabilité ?



Primitive Obsession

Nombres Magiques

Switch statement

Cause racine & amélioration de la conception



- Pour améliorer notre conception, **il ne faut pas s'arrêter aux symptômes**
- La cause racine de nos problèmes ici est la **Primitive Obsession**
 - **En faisant le choix de reposer sur des Ints et des Strings, on fragilise tout le système**
- Pour palier ce problème, il faut **réifier** les concepts associés :
 - Une valeur de carte n'est plus un **Int** mais une **Value**
 - Une couleur de carte n'est plus une **String** mais une **Color**
