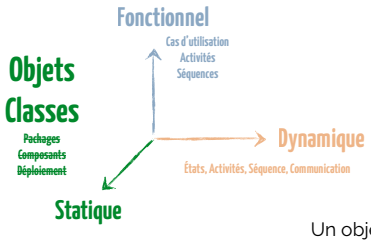
Des Objets aux Classes

Sébastien Mosser - INF5153
Chapitre 1 - Capsule 4
Automne 2020

crédits photos: Pixabay

On commence par la dimension statique



Les objets sont des
citoyens de premier ordre

On parle de conception
"**orientée objet**", pas
"**orientée classes**"

Un objet est une instance d'une classe

Qu'est-ce qu'un objet ?

- Un objet est une **boite noire**
  - On ne connait que son **interface publique** (son type)
  - Son **interface expose des messages** que l'objet sait recevoir
    - *En Java, les messages sont des invocations de fonctions/méthodes*
- Comment **travailler avec un objet** ?
  - Il faut commencer par **trouver sa référence** (l'instancier, le réutiliser)
  - Une fois qu'on l'a, on va **lui envoyer des messages**

Objet = Données + Traitement

Un exemple de code objet

ENCART CAMÉRA

```
public void netflixAndChill(SmartHome s) {

    HomeCinema hc = s.findTV("TV Salon");
    SmartLight light = s.findLight("Philips Light Hue 1");

    hc.startTv();
    hc.setSoundMode(Modes.DOLBY_5_1);
    hc.setChannel(Services.NETFLIX)

    light.setLumPercentage(0.40);

}
```

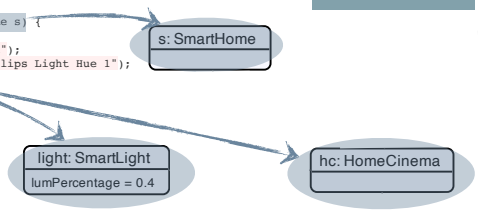
Objets Principaux

Envois de message

Représentation en UML : Diagramme d'objet

ENCART CAMÉRA

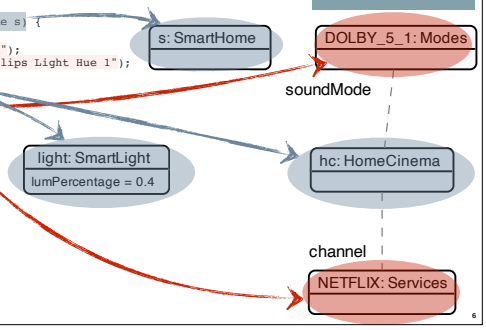
```
public void netflixAndChill(SmartHome s) {

  HomeCinema hc = s.findTV("TV Salon");
  SmartLight light = s.findLght("Philips Light Hue 1");

  hc.startTv();
  hc.setSoundMode(Modes.DOLBY_5_1);
  hc.setChannel(Services.NETFLIX)

  light.setLumPercentage(0.40);
}
```



Représentation en UML : Diagramme d'objet

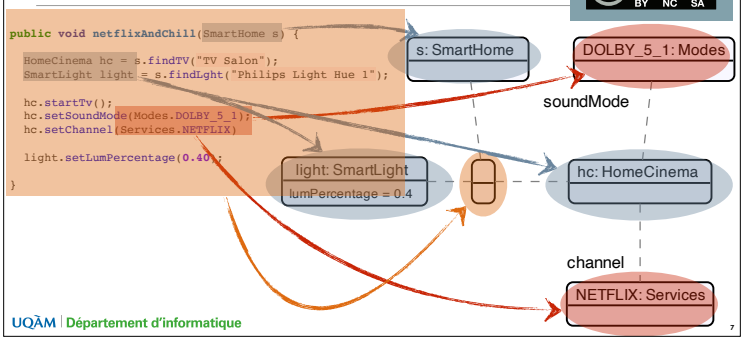
ENCART CAMÉRA

```
public void netflixAndChill(SmartHome s) {

  HomeCinema hc = s.findTV("TV Salon");
  SmartLight light = s.findLght("Philips Light Hue 1");

  hc.startTv();
  hc.setSoundMode(Modes.DOLBY_5_1);
  hc.setChannel(Services.NETFLIX)

  light.setLumPercentage(0.40);
}
```



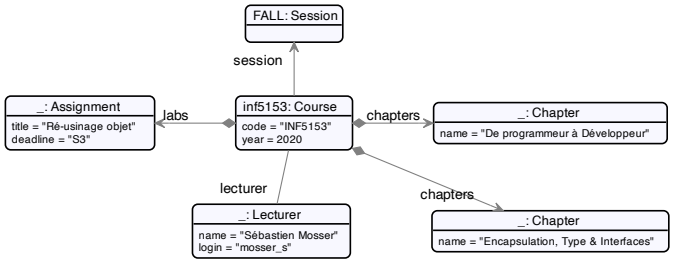
Représentation en UML : Diagramme d'objet

ENCART CAMÉRA



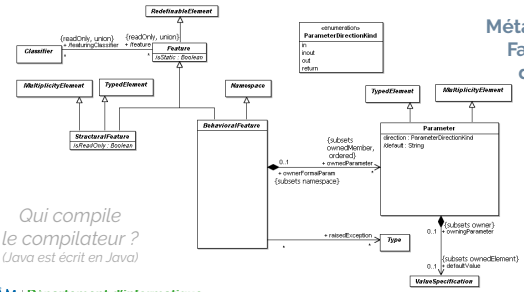
Et si on concevait le cours de conception ?

ENCART CAMÉRA



Rôle de la super-structure UML

ENCART CAMÉRA



Méta-modélisation : Faire le modèle des modèles

Qui compile le compilateur ?
(Java est écrit en Java)

740 pages

Forces et Faiblesses du diagramme d'objets

ENCART CAMÉRA

- Un **diagramme d'objet** permet :
  - De montrer une **représentation exhaustive** des objets qui collaborent lors de **l'exécution du programme**
  - De comprendre comment le **graphe d'objet** est constituée
- Il souffre de faiblesses non négligeables :
  - **Difficile à maintenir à jour** quand le code évolue (très couplé au code)
  - **Ne passe pas à l'échelle** pour visualiser de grosses grappes d'objets

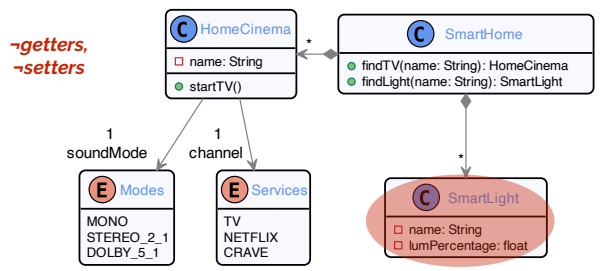
Mais qu'est-ce qu'une classe ?

ENCART CAMÉRA

- Une classe est une **fabrique à objet**
  - Fabriquer un nouvel objet (**new**), c'est "***instancier***" la classe
- Elle **définit l'interface publique** de ses instances :
  - Les **messages** que l'objet sait traiter (**signature** des méthodes)
- Elle **définit les détails d'implémentation, privés**
  - La **logique d'affaire** qui est exécuté à l'intérieur de la boite noire

Représentation en UML : Diagrammes de classes

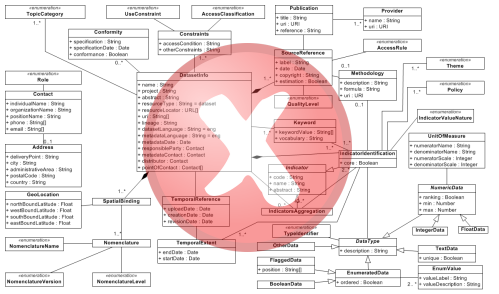
ENCART CAMÉRA



Comment allumer la lumière ?

Attention au niveau de granularité des modèles !

ENCART CAMÉRA

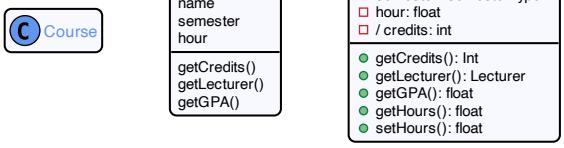


Les modèles doivent être utile à l'activité de conception, et aux développeurs.

Découpez vos modèles !

Différents niveaux de granularité

ENCART CAMÉRA



Il faut adapter la granularité à l'intention que sert le diagramme

On peut faire de l'objet sans classes ?

ENCART CAMÉRA

- La classe est "juste" une **fabrique à objet**
  - **Il existe des langages objets qui n'utilisent pas ce principe**
    - *Par exemple les langages à prototype reposent sur le clonage*
- En conception, c'est l'**utilisabilité / lisibilité** des objets qui est critique
  - **Est-ce qu'ils exposent les bons messages aux développeurs ?**
- **La classe est un moyen** pour définir les objets, **pas une finalité**
  - ***"Faire simple, c'est compliqué"***

Définition des modèles de conception

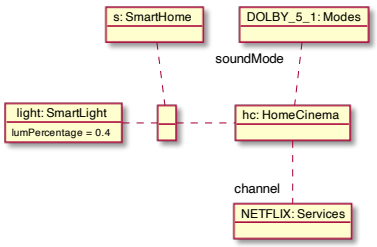
ENCART CAMÉRA

- Dans le cadre du cours, on utilise PlantUML
  - Vous devez passer du temps à **comprendre les principes de conception**, pas à apprendre un outil compliqué comme *VisualParadigm*
- PlantUML est un **langage textuel** pour "coder" des diagrammes UML
  - On écrit son **modèle dans un fichier texte**
  - On "**compile**" le modèle pour obtenir le schéma graphique
- **Les modèles peuvent être versionnés** dans Git facilement !

Modèle PlantUML : diagrammes d'Objets

ENCART CAMÉRA

```
@startuml

' Démonstration PlanTUML pour INF5153
' Auteur: Sébastien Mosser

object "s: SmartHome" as s
object "hc: HomeCinema" as hc
object "light: SmartLight" as light {
  lumPercentage = 0.4
}

object "DOLBY_5_1: Modes" as dolby
object "NETFLIX: Services" as netflix

object " " as main

s .. main
main . hc
light . main
dolby "soundMode".. hc
hc .. "channel" netflix

@enduml
```




```
mosser@lucifer inf5153 % plantuml -tpdf mon_diagramme.puml
```


Modèle PlantUML : diagrammes de Classes

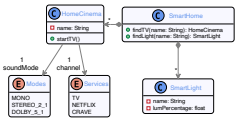
ENCART CAMÉRA

```
@startuml
!include ./../../../_commons.style

class SmartHome {
  + findTV(name: String): HomeCinema
  + findLight(name: String): SmartLight
}
SmartHome *--> "*" SmartLight
HomeCinema "*" <--* SmartHome
```

```
class SmartLight {
  - name: String
  - lumPercentage: float
}

class HomeCinema {
  - name: String
  + startTV()
}
HomeCinema --> "1\nsoundMode" Modes
HomeCinema --> "1\nchannel" Services
```

```
enum Modes {
  MONO
  STEREO_2_1
  DOLBY_5_1
}

enum Services {
  TV
  NETFLIX
  CRAVE
}
@enduml
```

```
mosser@lucifer inf5153 % plantuml -tpdf mon_diagramme.puml
```

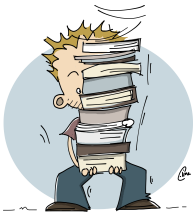


UQÀM | Département d'informatique

FACULTÉ DES SCIENCES
Université du Québec à Montréal

ENCART CAMÉRA

CC BY NC SA



https://mosser.github.io/

ace

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

Abonne toi à la chaine,
et met un pouce bleu !