

# LOG210 SÉANCE #03

## ANALYSE ET CONCEPTION DE LOGICIELS

1. Administration 
2. Équipe: Confiance
3. MDD: Classes, Attributs, Associations
4. DSS: Exercice, résumé
5. Contrats d'opération
6. RDCU: Créateur et Expert GRASP



# ADMINISTRIVIA

## 1. Accès G Suite (**ETSMTL.NET**)

- Google Classrooms
- Google Drive

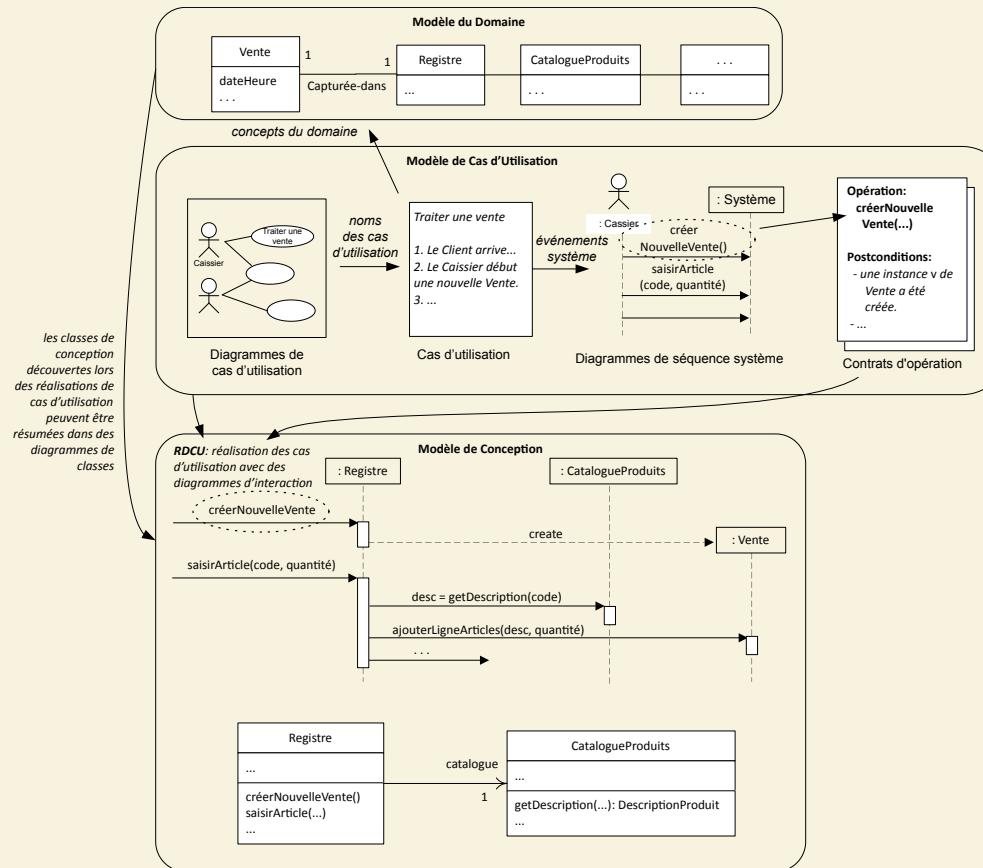
## 2. Invitation Lab 0 (GitHub Classroom)

-  lien différent pour chaque groupe!
- ne pas partager

## 3. Équipes (connaissez-vous vos coéquipiers?)

## 4. ~~Lien Zoom~~ Serveur Discord pour lab

☰ Ajouter patron: Transformer identifiant en objets



# LOG210 SÉANCE #03

## ANALYSE ET CONCEPTION DE LOGICIELS

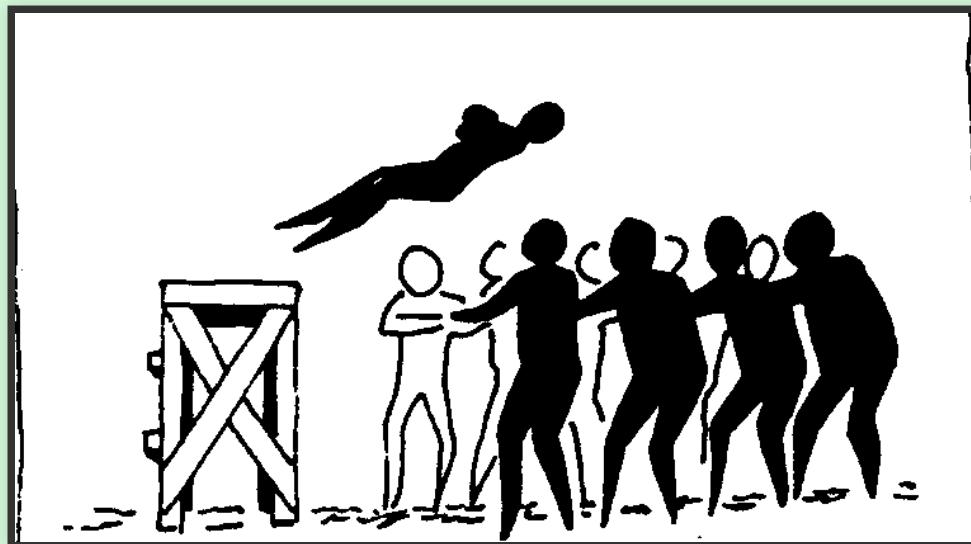


1. Administration
2. Équipe: Confiance ← S20203
3. MDD: Classes, Attributs, Associations
4. DSS: Exercice, résumé
5. Contrats d'opération
6. RDCU: Créateur et Expert GRASP

# CONFiance

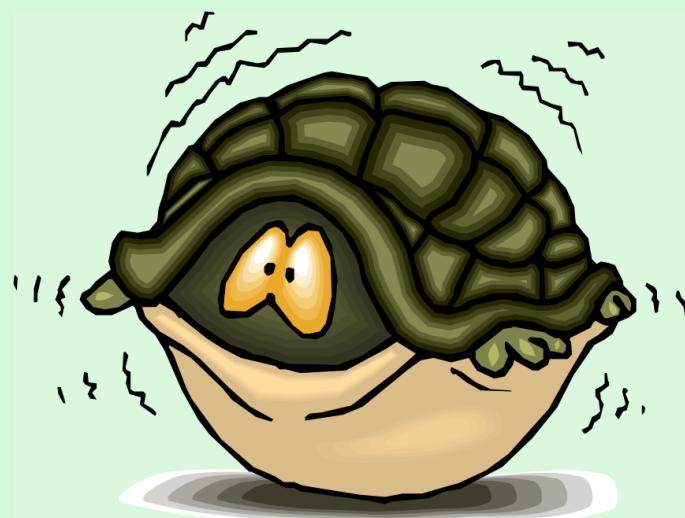
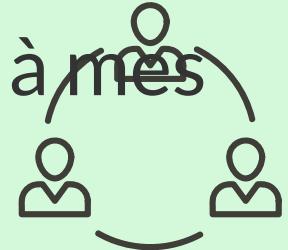


- Je crois que les autres coéquipiers sont compétents et qu'ils feront la bonne chose.
- Je suis à l'aise lorsqu'ils prennent le volant, le cas échéant.



# TENDANCES COMPORTEMENTALES CHEZ LES DÉVELOPPEURS...

- Je ne veux pas montrer mon code source à mes coéquipiers...
    - Mon code n'est pas fini...
    - J'ai trop peur d'être jugé(e)...
    - J'ai peur que quelqu'un vole mon idée...
- Il s'agit de l'insécurité. C'est normal!





# CACHER SES IDÉES AUGMENTE LE RISQUE...



- de faire des erreurs dans la conception initiale.
- de « réinventer la roue ».
- de terminer le travail plus tard que son compétiteur, qui, lui, a collaboré avec son équipe.



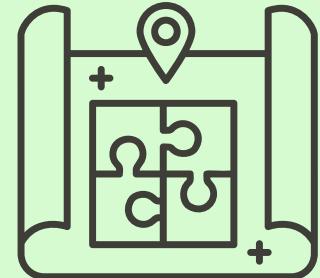
# LOG210 SÉANCE #03

## ANALYSE ET CONCEPTION DE LOGICIELS



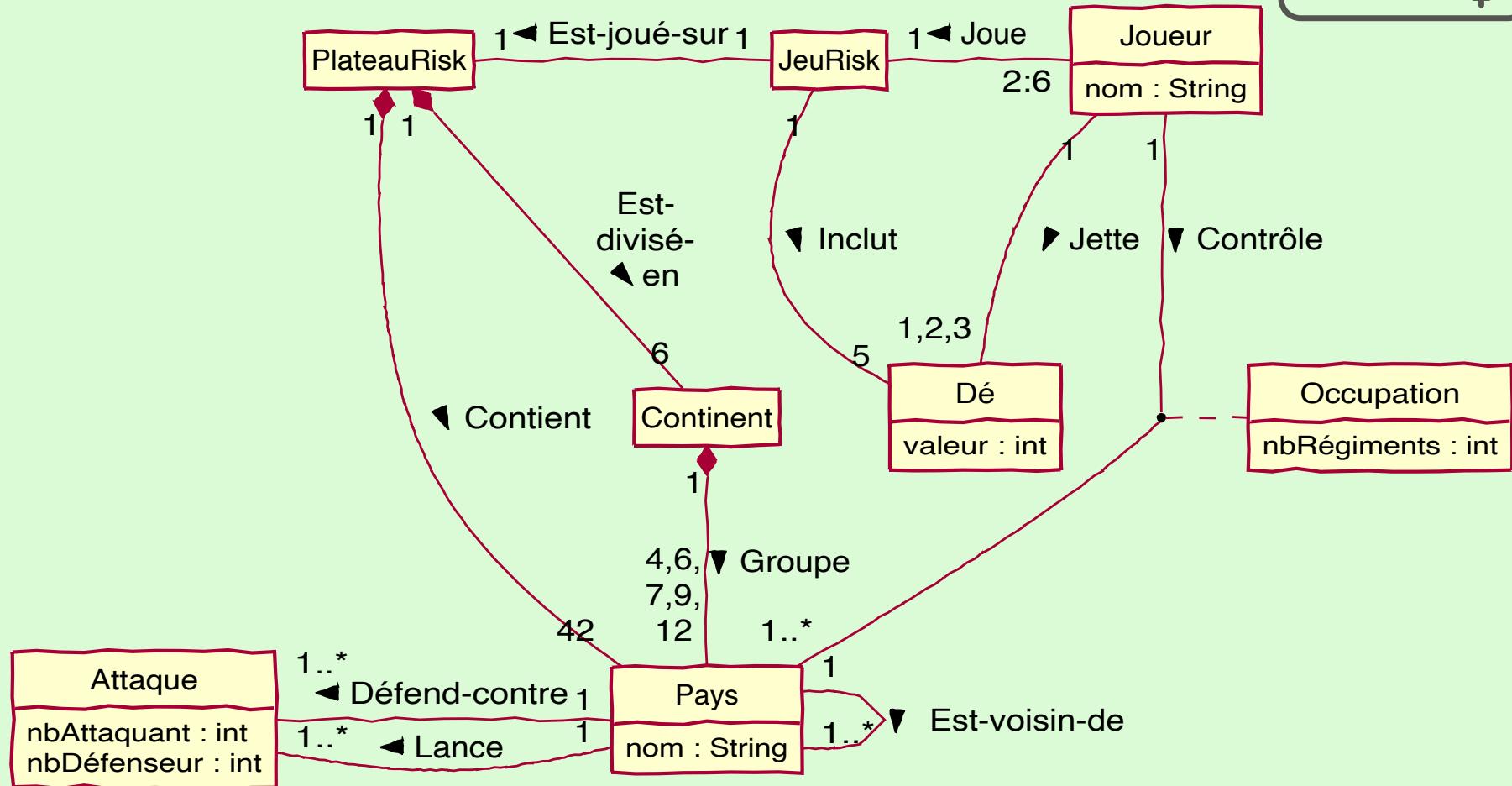
1. Administration
2. Équipe: Confiance
3. MDD: Classes, Attributs, Associations  S20203
4. DSS: Exercice, résumé
5. Contrats d'opération
6. RDCU: Créateur et Expert GRASP

# MDD

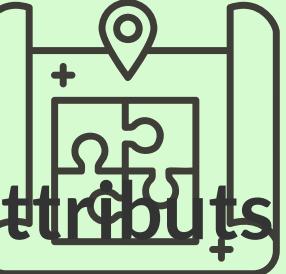


- Classes conceptuelles
- Attributs
- Associations

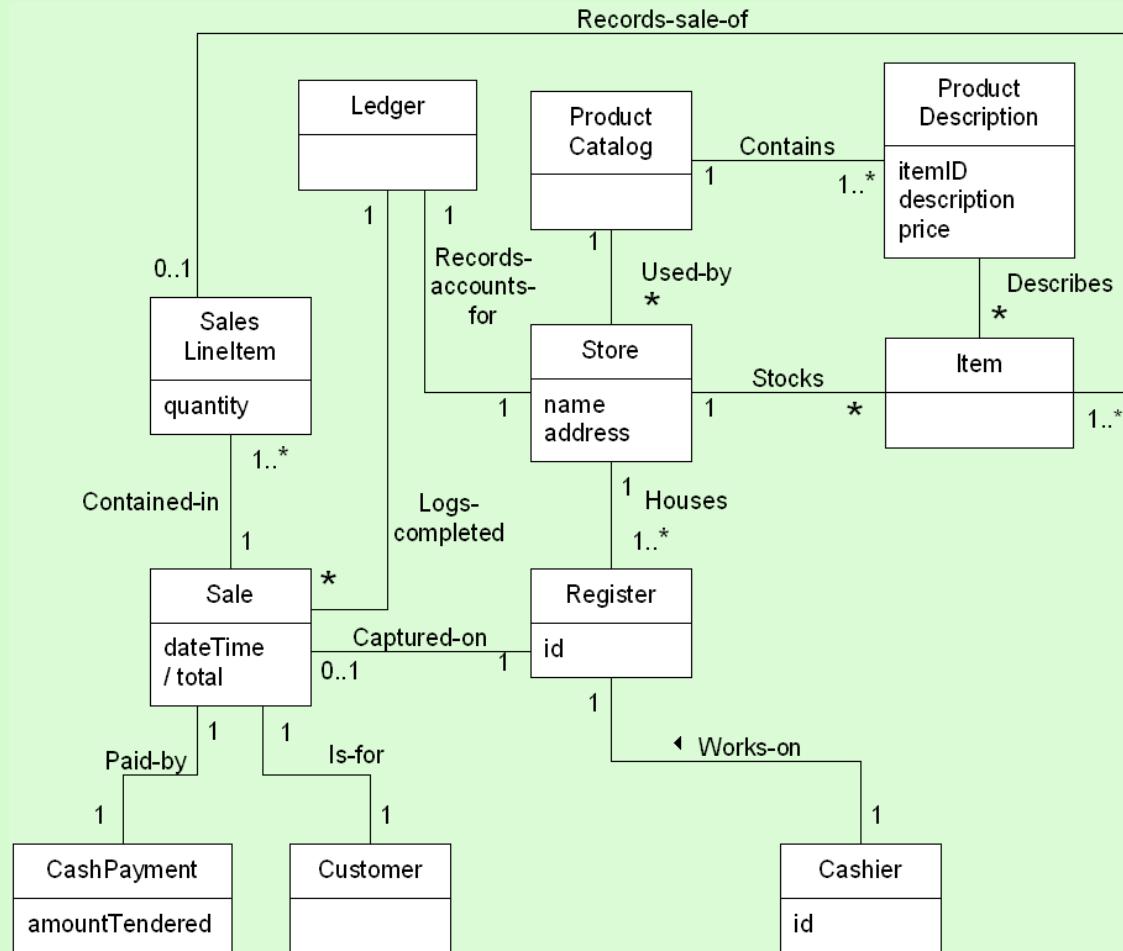
# \*MDD - JEU RISK



# MDD - ATTRIBUTS NEXTGEN (LARMAN)



Dans LOG210 il faut spécifier les types des attributs



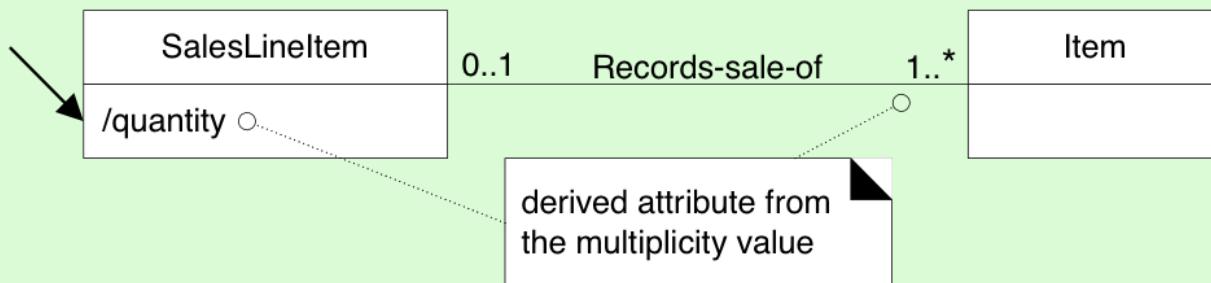
# ATTRIBUTS DÉRIVÉS



Each line item records a separate item sale.  
For example, 1 tofu package.

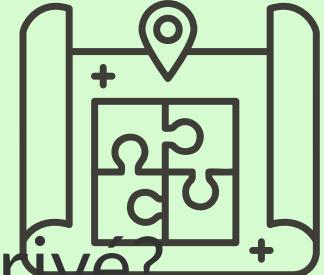


Each line item can record a group of the same kind of items.  
For example, 6 tofu packages.

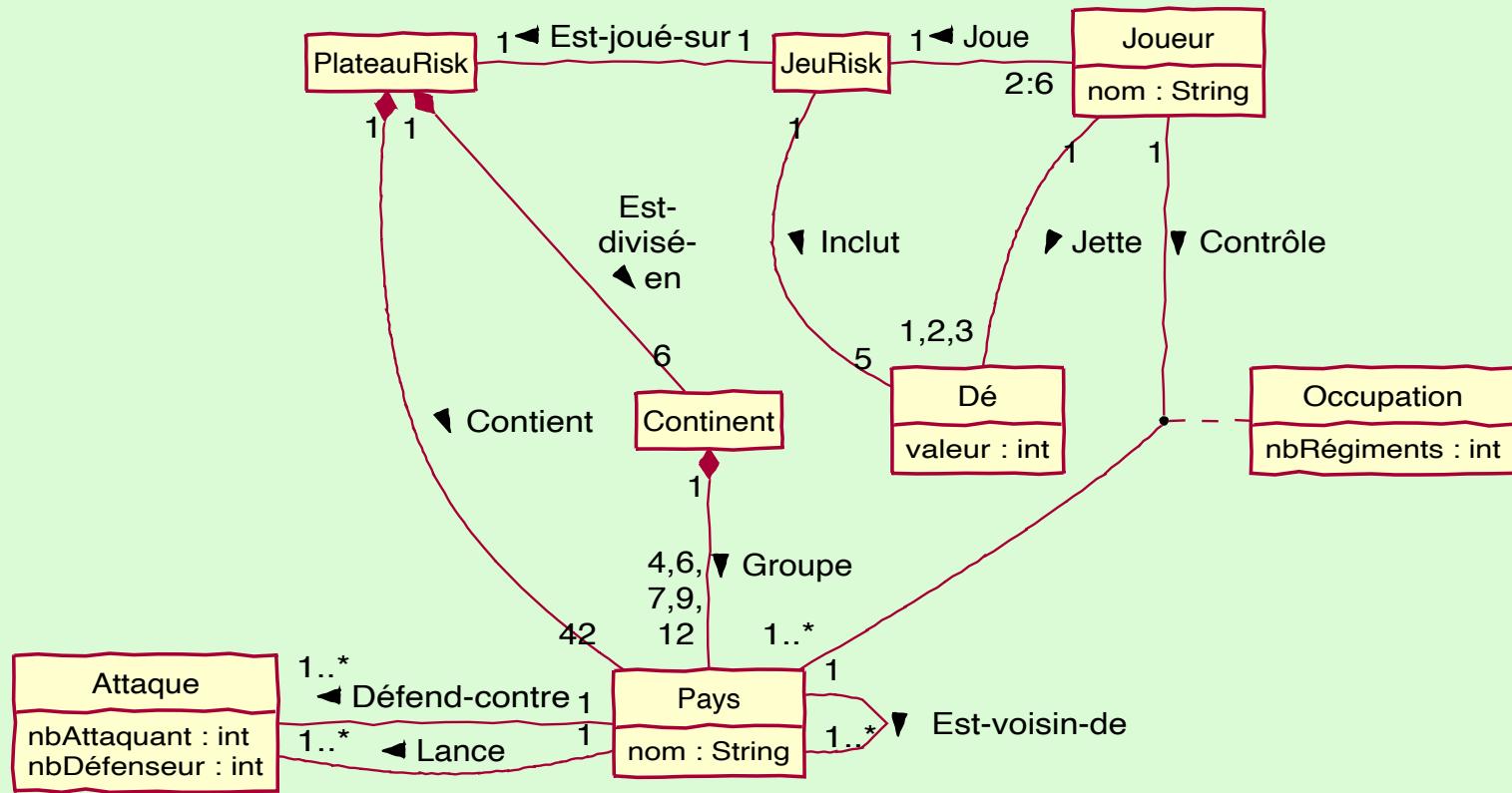


derived attribute from the multiplicity value

# \*ATTRIBUT DÉRIVÉ - RISK

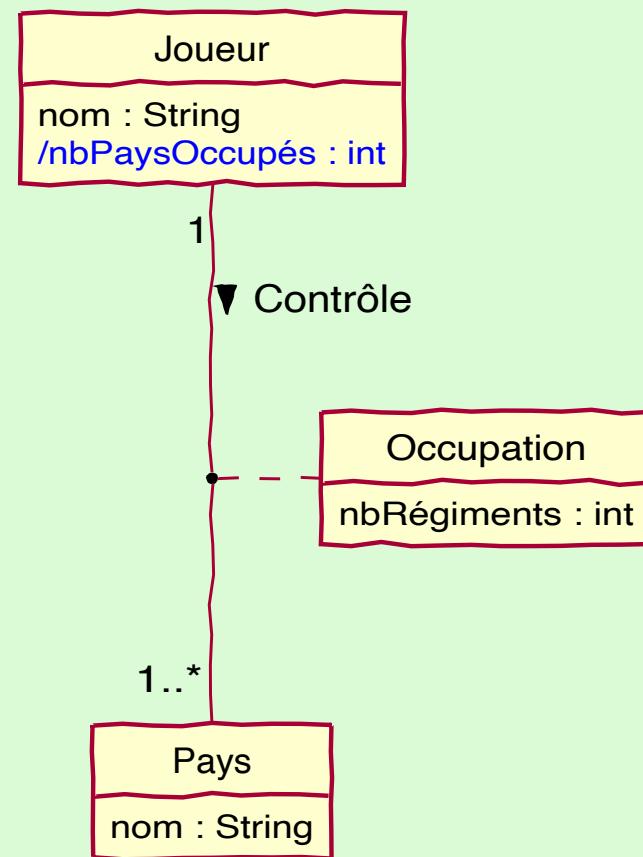
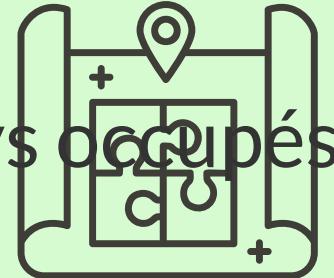


Voyez-vous où appliquer un attribut dérivé?



# CALCUL DES RENFORTS

On reçoit les renforts selon le nombre de pays occupés.  
Utiliser une classe d'association





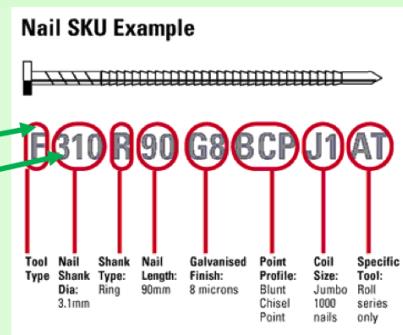
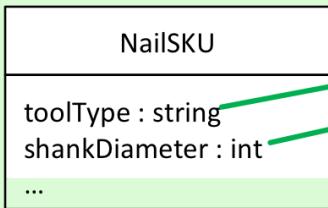
# TYPES DE DONNÉES CH. 9.16



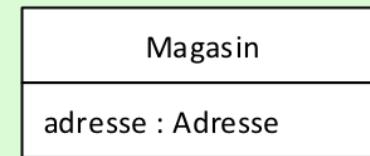
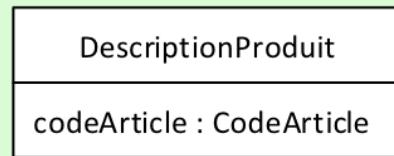
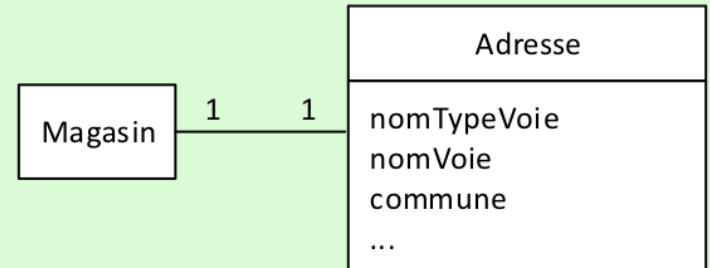
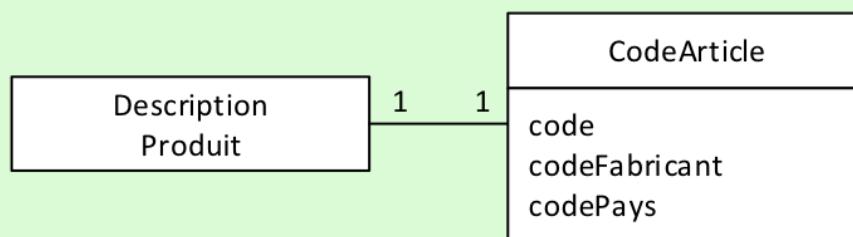
- Définir de nouveaux types quand les types primitifs ne sont pas adéquats
  - ex : Sigle de cours \$\rightarrow\$ LOG120



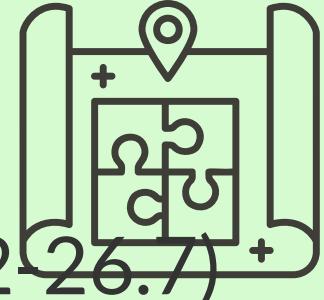
- ex : un code SKU



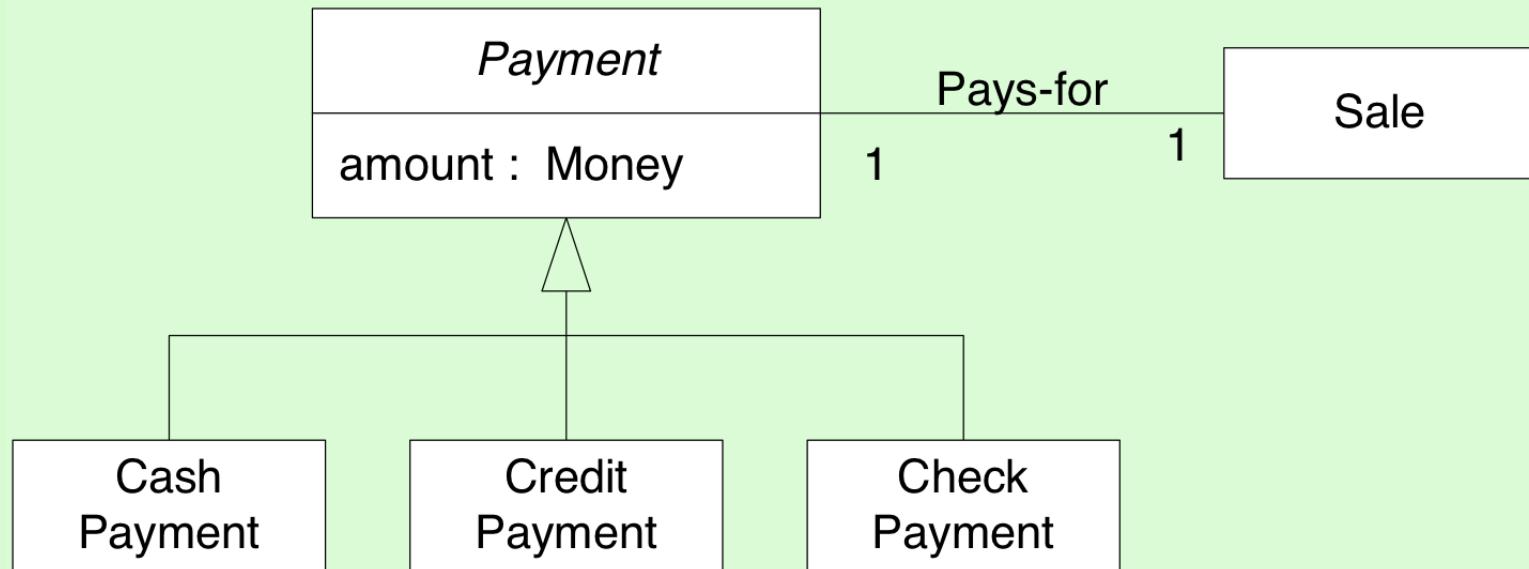
# UML : CLASSES TYPES DE DONNÉES



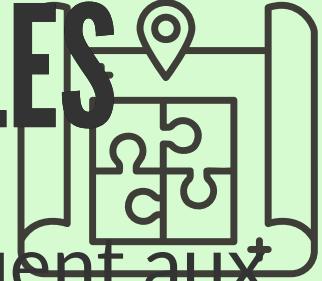
# \*GÉNÉRALISATION



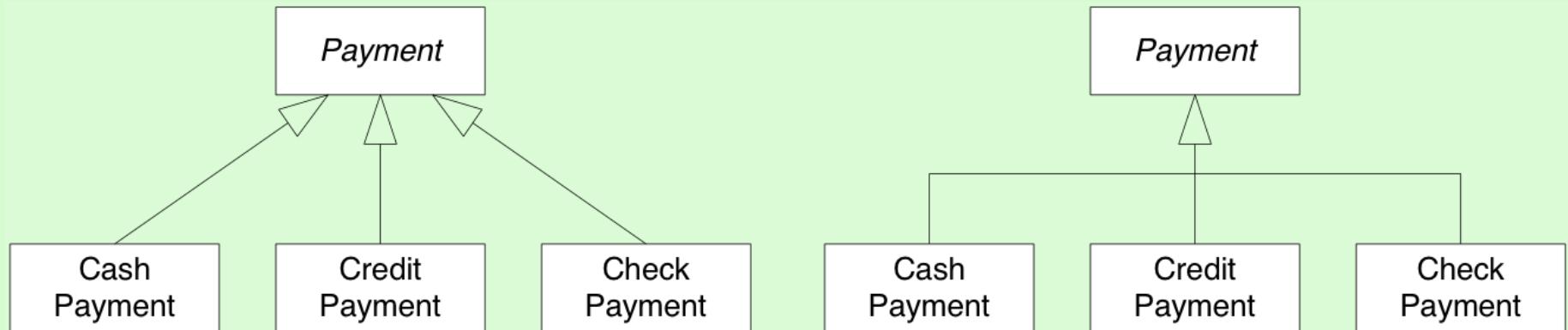
Lire Ch. 32.2-32.7 (version française 26.2-26.7)



# SOUS-CLASSES CONCEPTUELLES



Les énoncés sur les super-classes s'appliquent aux sous-classes



# RÈGLE « À 100% »

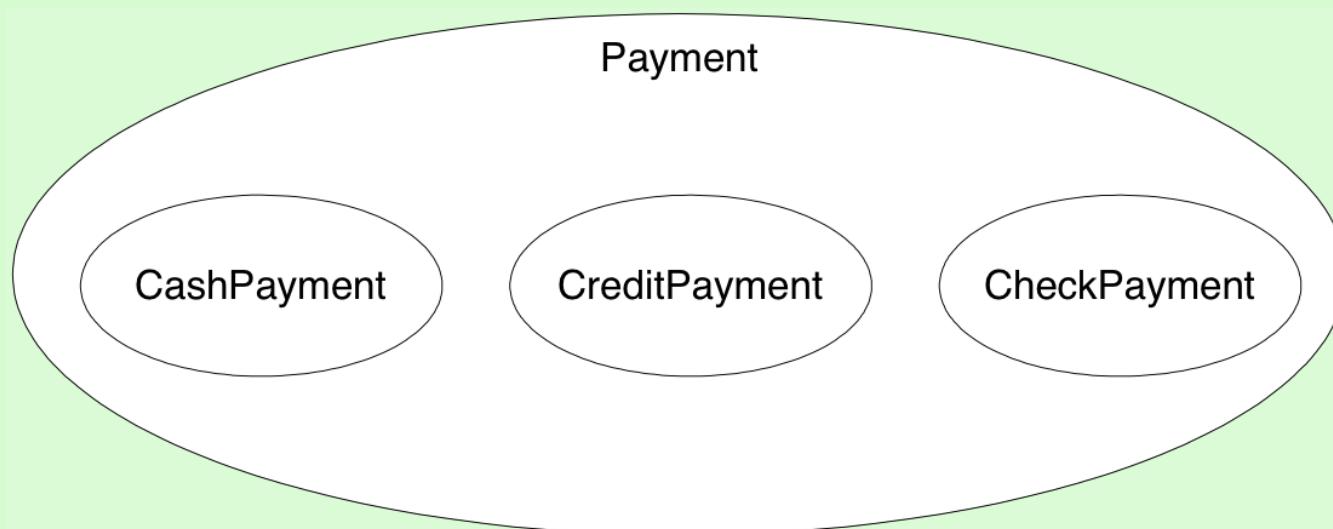


- 100% de la définition d'une classe conceptuelle est applicable aux sous-classes
- Sous-classe
  - conforme à 100% à sa superclasse
    - en termes d'attributs
    - en termes d'associations

# RÈGLE « EST UN »



- Sous-classe conceptuelle
  - est membre de l'ensemble de sa superclasse
  - objet d'une sous-classe **est un** objet de la super-classe

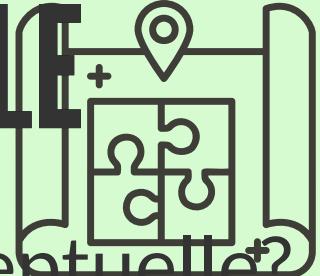


# DÉFINIR UNE SOUS-CLASSE CONCEPTUELLE QUAND ELLE...

- possède des attributs additionnels intéressants
- possède des associations additionnelles intéressantes
- réagit, est manipulée, est traitée d'une façon intéressante et
  - différemment de sa superclasse
  - différemment d'autres sous-classes
- représente un objet animé (humain, animal, robot) ayant un comportement différent et intéressant



# SUPER-CLASSE CONCEPTUELLE



- Quand définit-on une super-classe conceptuelle?
  - Lorsque les classes sont des variations de concepts similaires
  - respectent les règles à 100% et « est un »
  - ont des attributs communs
    - qui pourraient être placés dans la super-classe
  - ont les mêmes associations
    - qui pourraient s'appliquer à la super-classe

# EXEMPLE DE PARTITION INUTILE

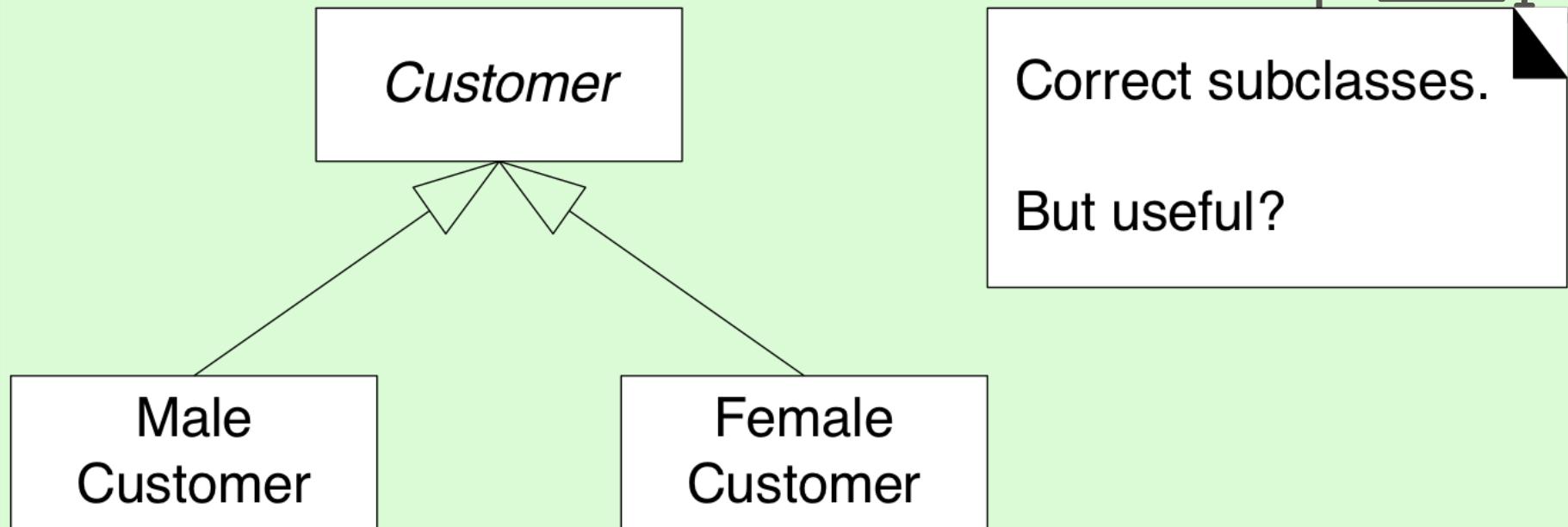


figure: F26.6, A32.6

# CLASSES D'ASSOCIATION

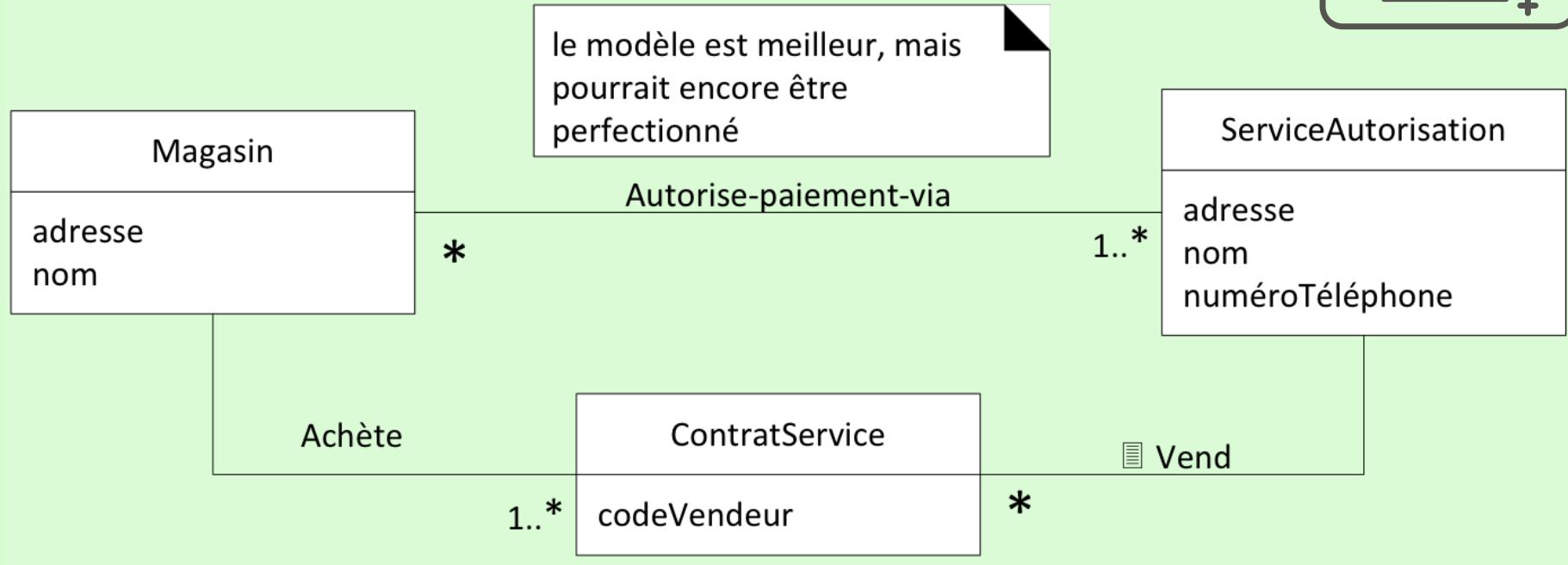
- Services d'autorisation de paiement
  - assignent une identification à chaque magasin
  - demande d'autorisation de paiement  $\square$  codeVendeur
  - le même magasin peut utiliser plusieurs services, chacun lui donnant un codeVendeur différent
- Dans quelle classe conceptuelle mettre codeVendeur?

Magasin
adresse
<b>codeVendeur</b>
nom

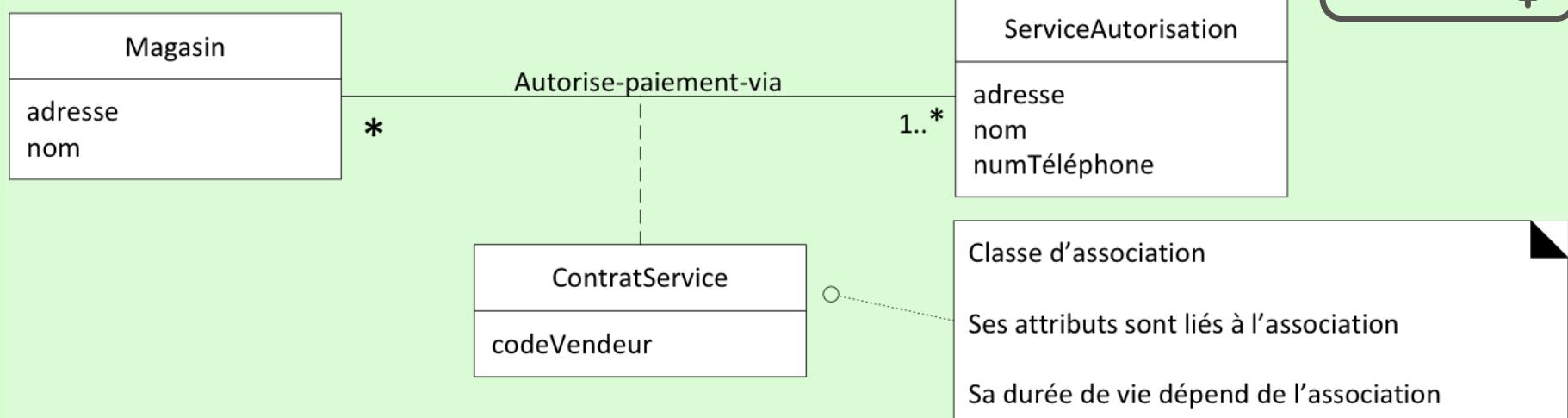
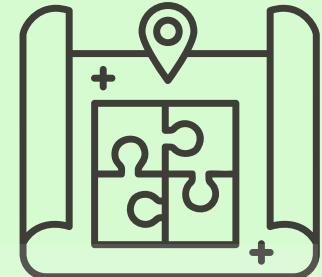
les deux localisations de codeVendeur sont incorrects, puisqu'il peut y en avoir plus d'un

ServiceAutorisation
adresse
<b>codeVendeur</b>
nom
numTéléphone

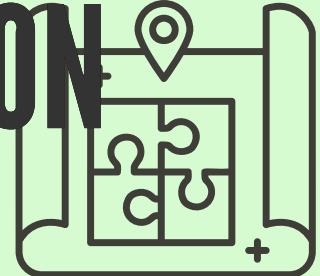
# SOLUTION INTERMÉDIAIRE



# CLASSE D'ASSOCIATION



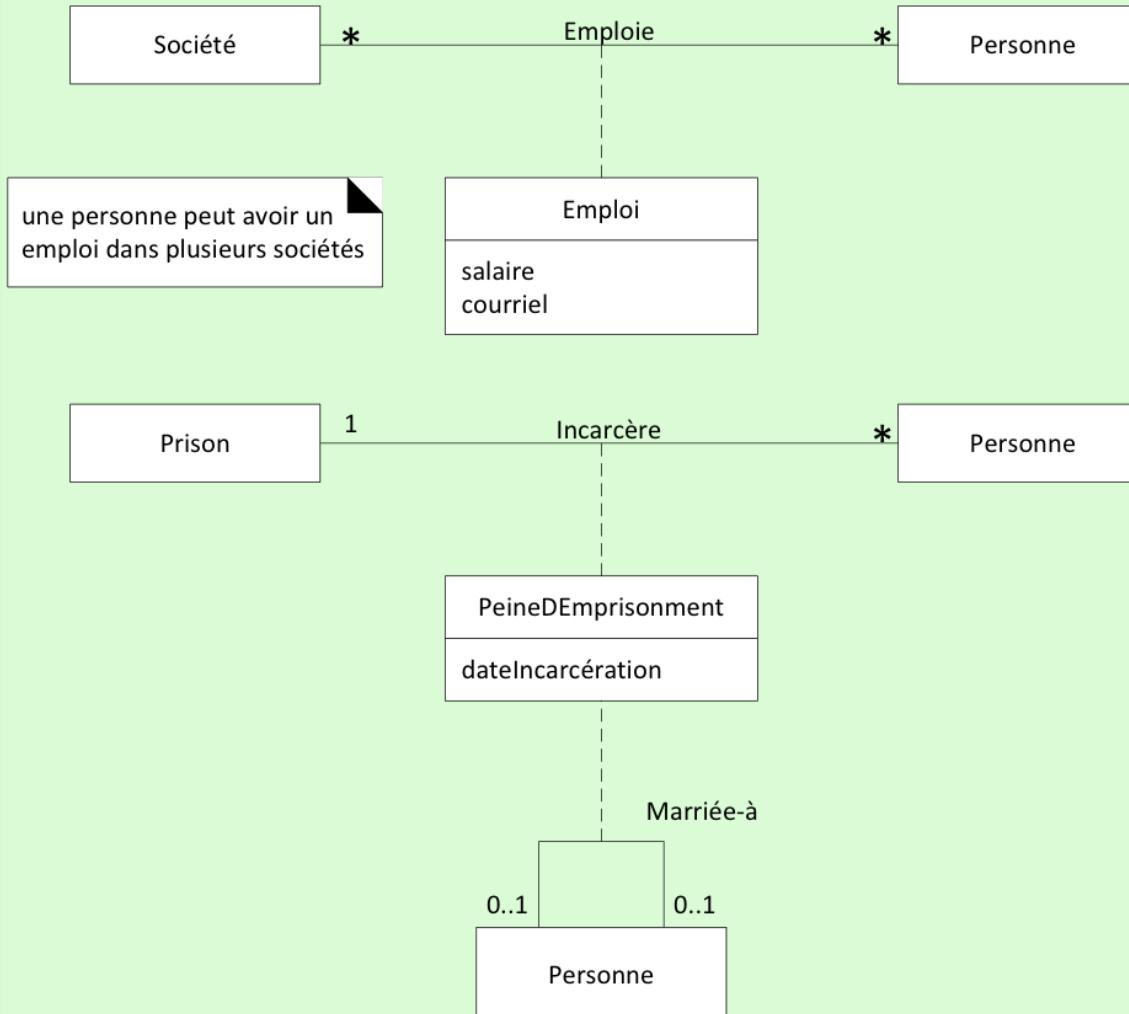
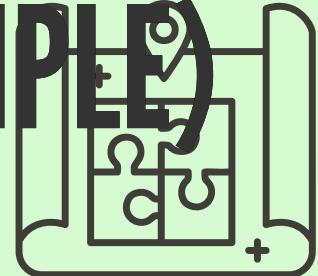
# AJOUT - CLASSE D'ASSOCIATION



- Règles pour l'ajout :
  - un **attribut** est lié à une association
  - la durée de vie des instances de la classe d'association dépend de l'association
    - «historic mapping» (patron d'analyse, M. Fowler)
- il y a une association N-N entre deux concepts et des **informations liées à l'association** elle-même.

<https://martinfowler.com/apsupp/properties.pdf>

# CLASSES D'ASSOCIATION (EXEMPLE)



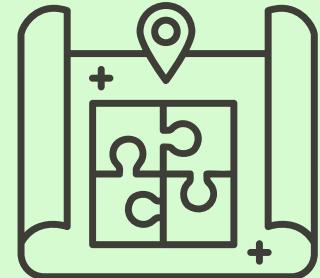
# \*SOLUTION EXERCICE

MDD SUR GOOGLE

CLASSROOM

voir: seance-03-exercice-reserverLivre

# EXERCICE MDD (HÔTEL)



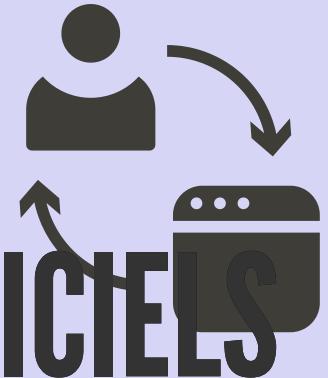
1. Un client appelle à l'hôtel pour placer une réservation.
2. Le commis démarre une nouvelle réservation.
3. Le commis saisit la date d'arrivée; la date de départ; le nom de la catégorie de chambre; la quantité de chambres.
4. Le système inscrit les informations à la réservation.

Les étapes 3 et 4 sont répétées tant que le client n'indique pas qu'il a terminé

5. Le commis termine la réservation.
6. Le système affiche toutes les informations entrées.
7. Le commis valide les informations auprès du client et confirme la réservation à l'aide du nom et du numéro de téléphone du client.
8. Le système enregistre la réservation et affiche le numéro de confirmation.
9. Le commis communique le numéro de confirmation au client.

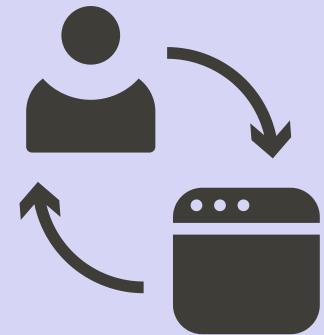
# LOG210 SÉANCE #03

## ANALYSE ET CONCEPTION DE LOGICIELS



1. Administration
2. Équipe: Confiance
3. MDD: Classes, Attributs, Associations
4. DSS: Exercice, résumé ← S20203
5. Contrats d'opération
6. RDCU: Créateur et Expert GRASP

# DSS

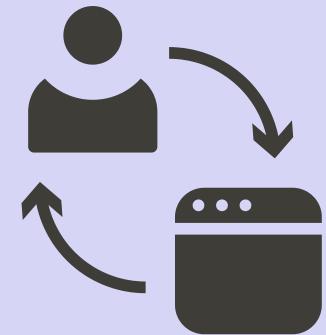


Retour sur l'exercice Google

Voir: Seance-03-solution-reserverLivreBibliotheque

# POINTS IMPORTANTS

Voir le PDF des notes de cours



# LOG210 SÉANCE #03

## ANALYSE ET CONCEPTION DE LOGICIELS



Créé par Sébastien Ral

1. Administration
2. Équipe: Confiance
3. MDD: Classes, Attributs, Associations
4. DSS: Exercice, résumé
5. Contrats d'opération  S20203
6. RDCU: Créeur et Expert GRASP

# CONTRATS



# D'OPÉRATION

Created by Mohammed Ra

- Qu'est-ce qu'un contrat d'opération?
- Pourquoi les contrats d'opération?

# \*QU'EST-CE QU'UN CONTRAT D'OPÉRATION?



Created by Mohammed Ra

Un document décrivant ce qui est arrivé après l'exécution d'une opération système.

# ÉLÉMENTS D'UN CONTRAT D'OPÉRATION



- Signature de l'opération système:  
`saisirArticle(codeArticle : CodeArticle, quantité : int)`
- Postconditions (3 formes, vocabulaire du MDD)
  - création (ou suppression) d'instances;
  - modification des valeurs des attributs;
  - formation (ou rupture) d'associations.

Created by Mohammed Ra

# \*CONTRAT D'OPÉRATION:

Opération: créerNouvelleVente( )



Références croisées: Cas d'utilisation : Traiter Vente

Created by: Mohammed Ra

Postconditions:

- une instance v de Vente a été créée
- v a été associée au Registre
- des attributs de v ont été initialisés

# POURQUOI LES CONTRATS



- Facilite la conception OO (RDCU)
  - Liste d'épicerie (quoi faire)
  - RDCU s'inspire du MDD (réduire décalage des représentations)
- Donne des conditions pour tests
- Aide à valider le MDD

Created by Mohammed Ra

# EXERCICE CONTRAT<sup>S</sup>



# (GOOGLE CLASSROOM)

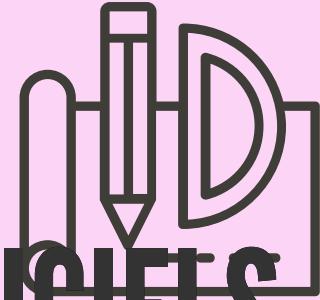
Created by Mohammed Ra

Voir seance-03-exercice-reserverLivre



# LOG210 SÉANCE #03

## ANALYSE ET CONCEPTION DE LOGICIELS

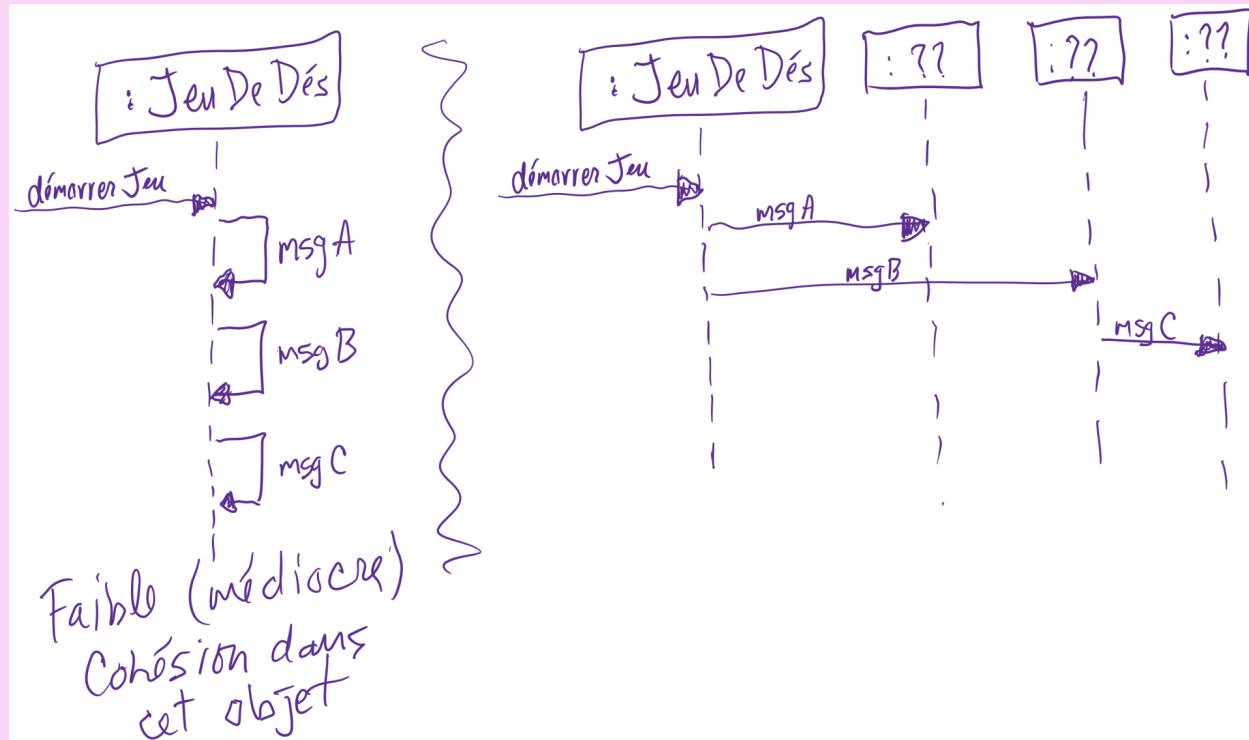


1. Administration
2. Équipe: Confiance
3. MDD: Classes, Attributs, Associations
4. DSS: Exercice, résumé
5. Contrats d'opération
6. RDCU: Créeur et Expert GRASP  S20203

# RDCU



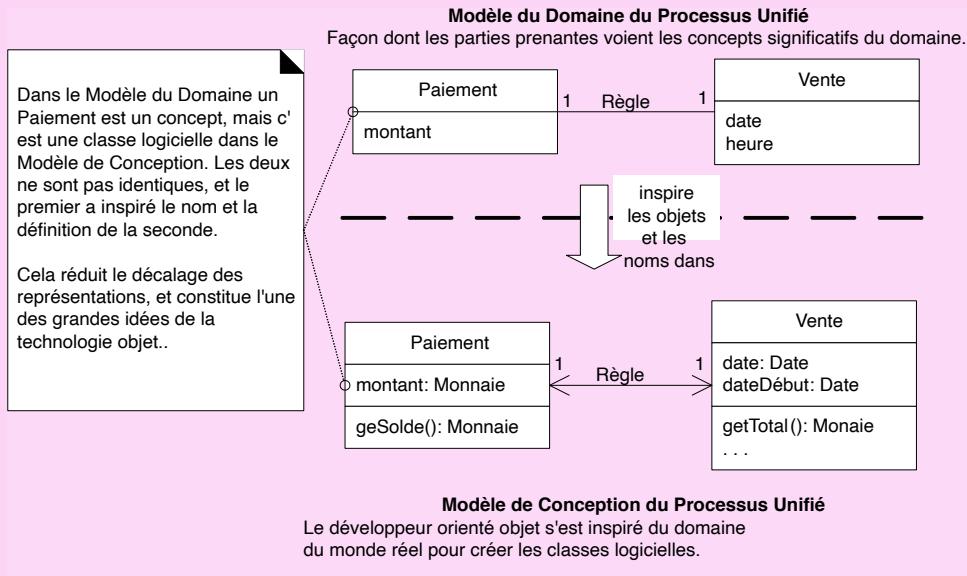
Prendre les bonnes décisions pour une solution facile à comprendre et modulaire...



# DÉCALAGE DES REPRÉSENTATIONS

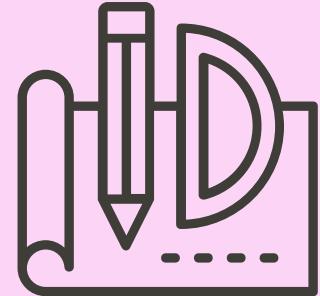


Facile? Les classes logicielles devraient ressembler à des classes conceptuelles.



Qui fait quoi? Qui a quelle responsabilité?

# RDCU



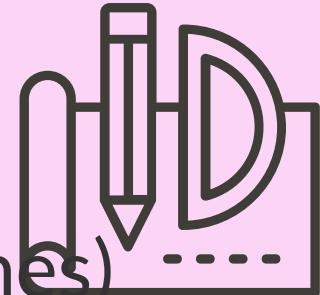
Approche: conception orientée-responsabilités

## GRASP

General Responsibility Assignment Software Patterns

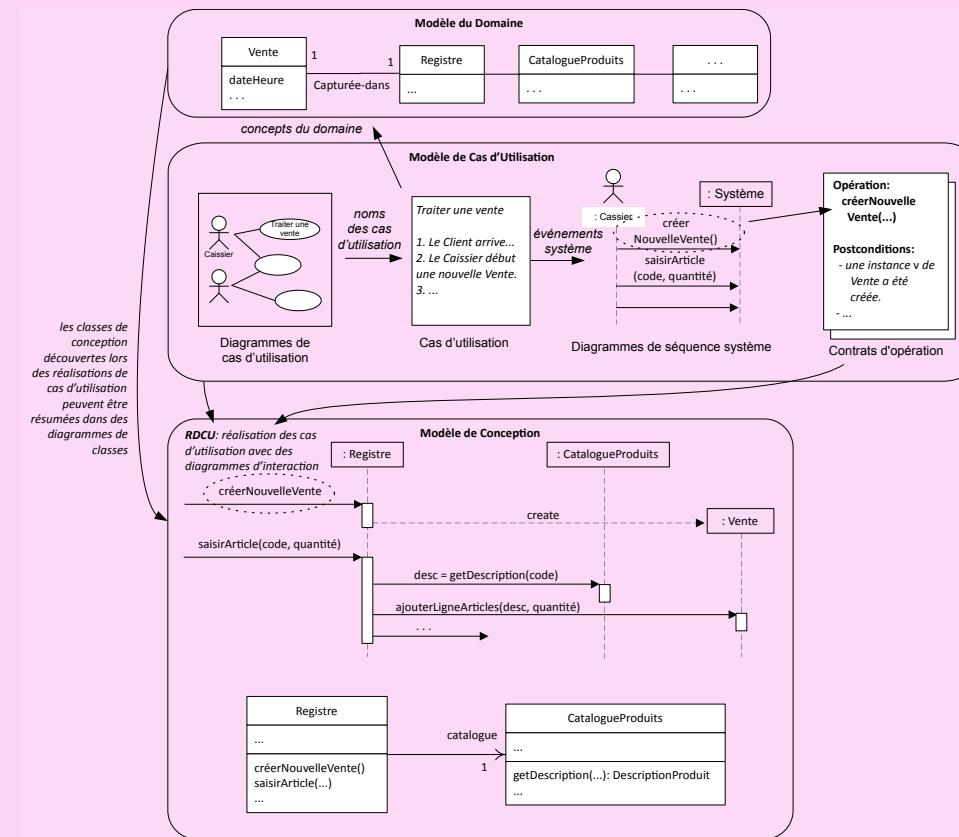
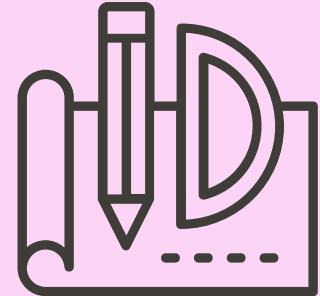
Pour décider où mettre les méthodes...

# GRASP

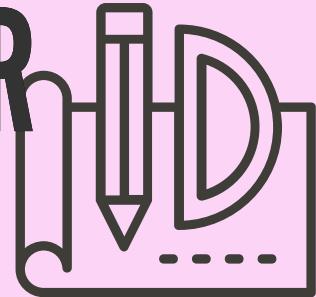


- Contrôleur (séparation des couches)
- Créateur
- Expert en information
- Faible couplage
- Forte cohésion
- Polymorphisme
- Indirection
- Protection de variation
- Fabrication pure

# \*RDCU (SURVOL)

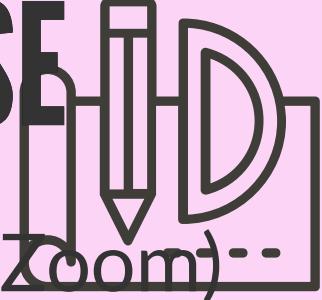


# RDCU: SCÉNARIO DÉMARRER

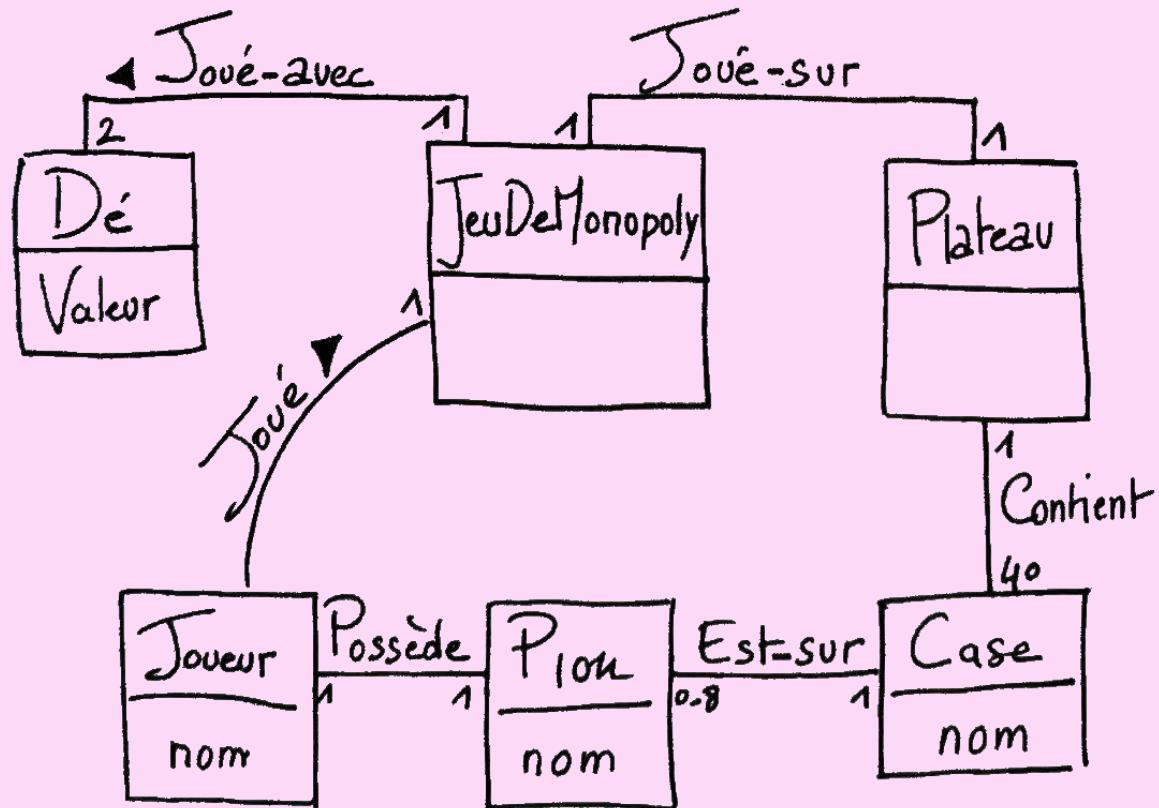


- C'est l'initialisation du système.
- C'est implicite mais essentiel!
- On doit instancier les objets faisant partie de l'application.

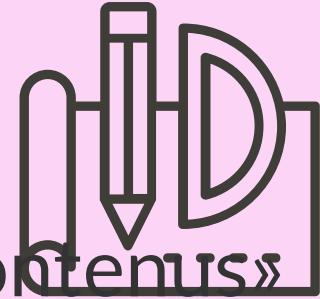
# INSTANCIER LES OBJETS CASE



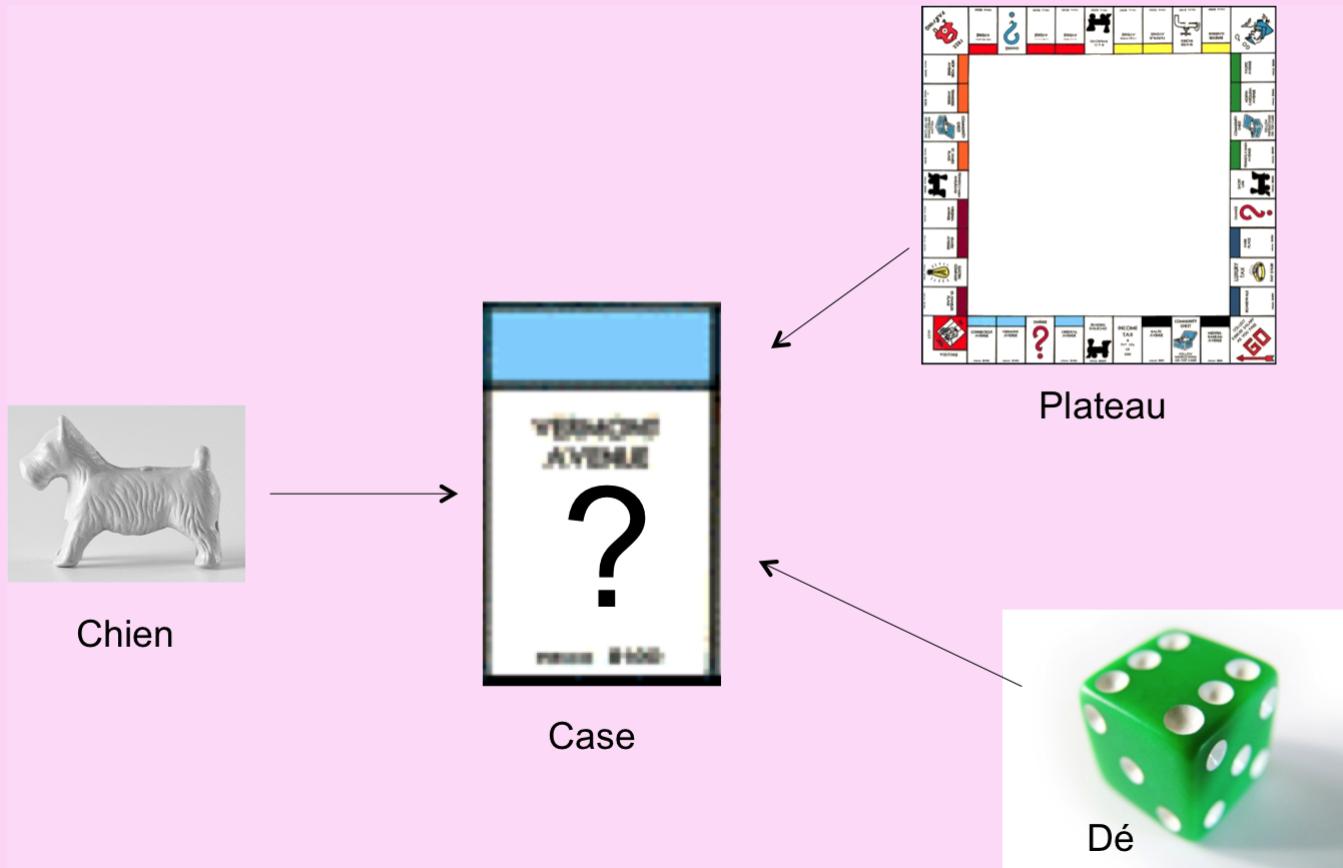
- Quelle classe s'en occupe? (sondage Zoom) --



# CRÉATEUR (GRASP)



- Les «conteneurs» créent les objets «contenus»



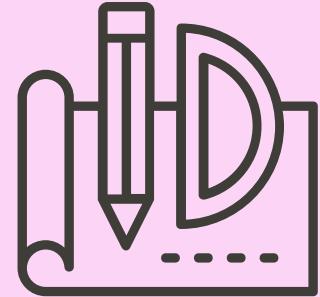
# CRÉATEUR (GRASP)



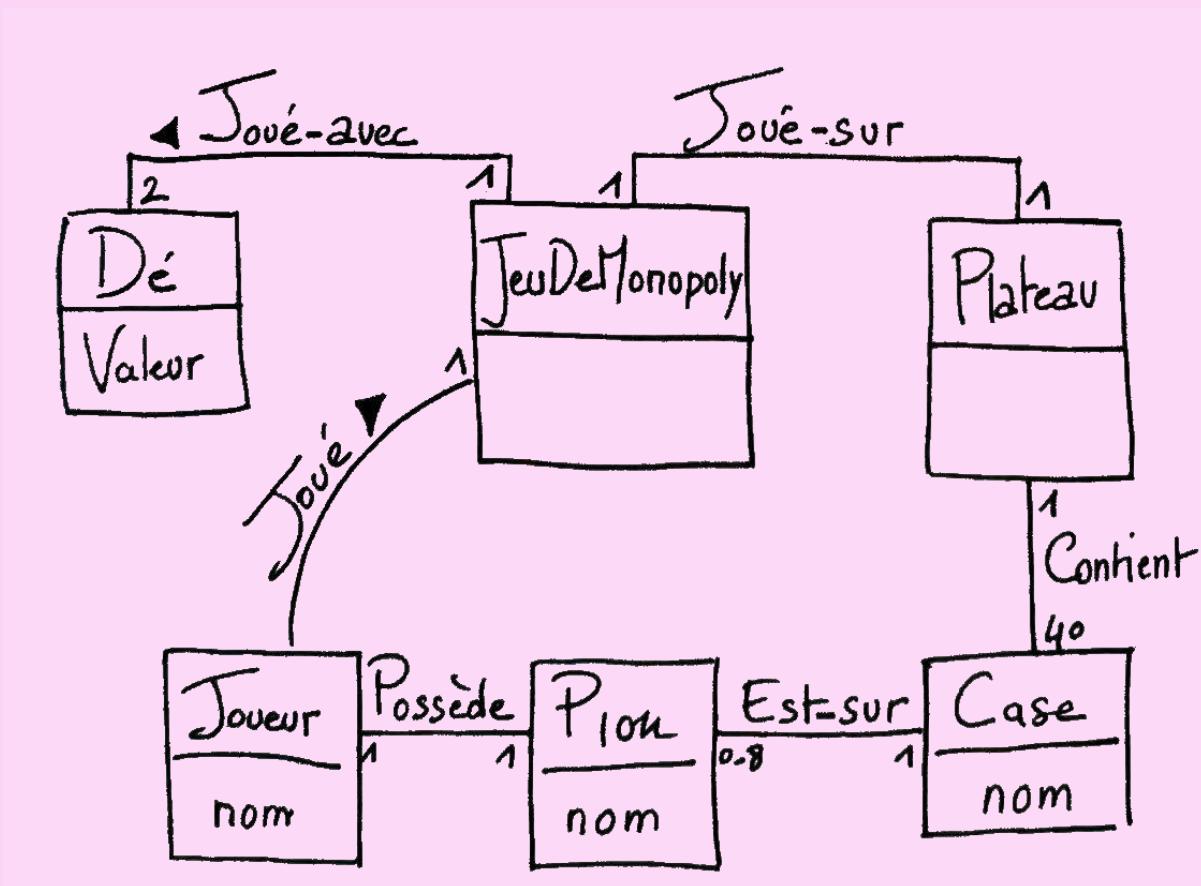
- **Problème:** Qui crée? (postcondition d'un contrat)
- **Solution:** Affecter à la classe B la responsabilité de créer les objets d'une classe A si...
  - B possède les données d'initialisation des objets A
  - B contient ou agrège des objets A
  - B utilise étroitement des objets A
  - B enregistre des objets A

On s'inspire du MDD. On réutilise les liens existents.

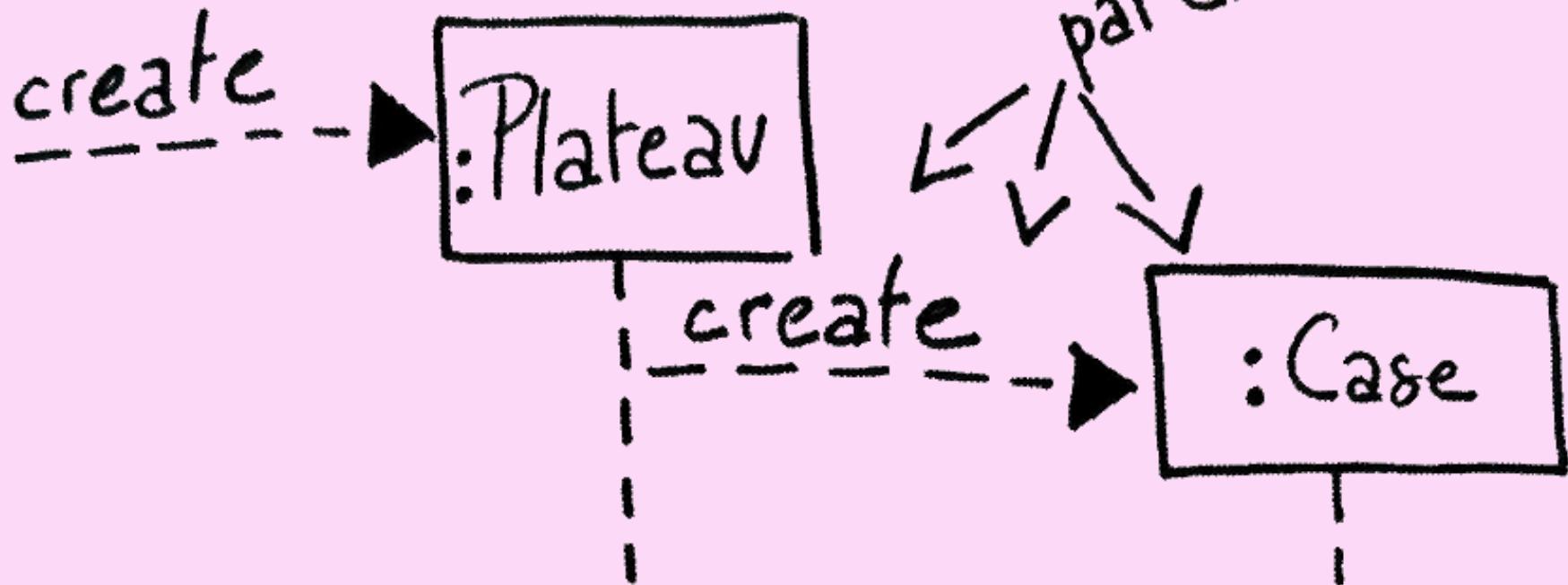
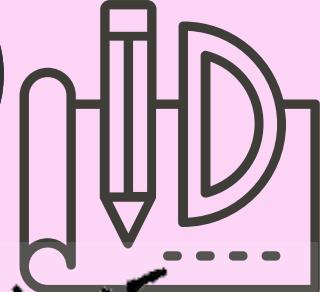
# \*CRÉATEUR



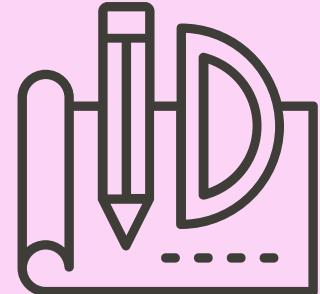
- Qui crée les cases (Square)?



# \*CRÉATEUR (ANNOTATION)



# EXPERT EN INFORMATION



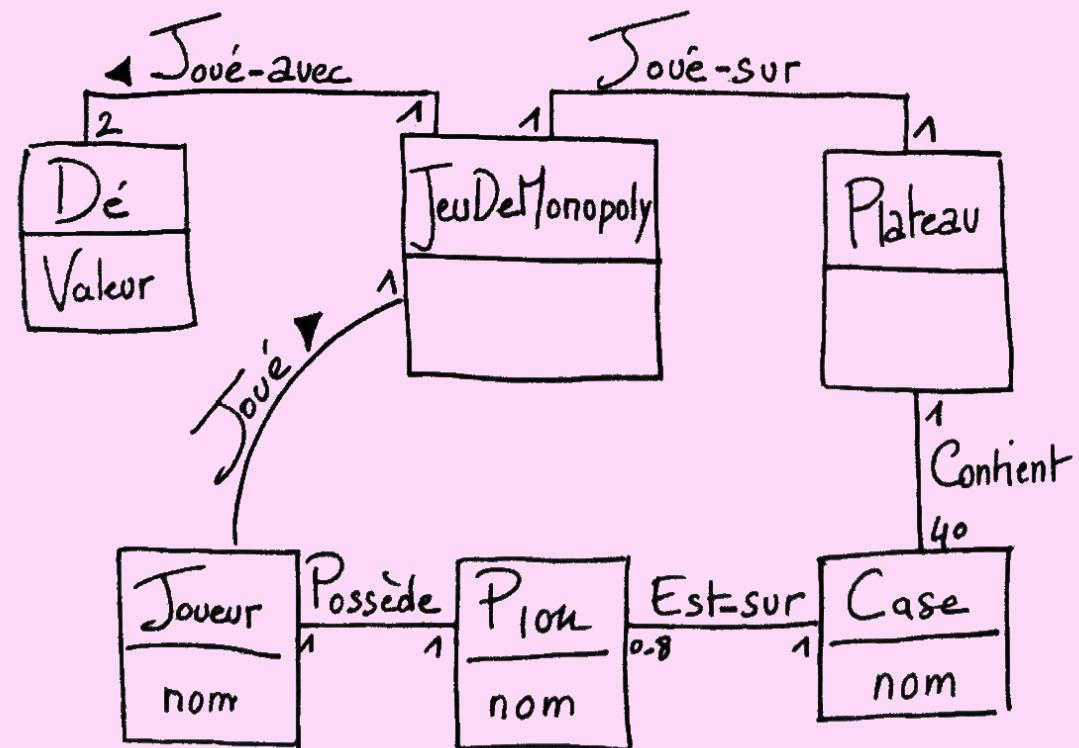
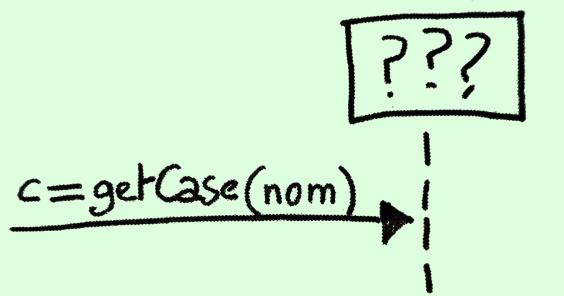
- **Problème:** Quel est le principe général d'affectation des responsabilités aux objets?
- **Solution:** Affecter la responsabilité à la classe qui possède les informations nécessaires pour s'en acquitter

*En termes de paramètres, associations*

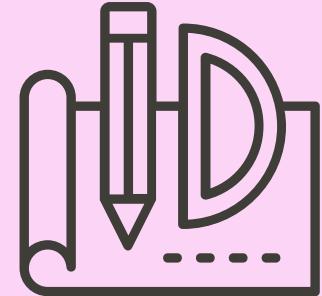
# EXPERT (GRASP)



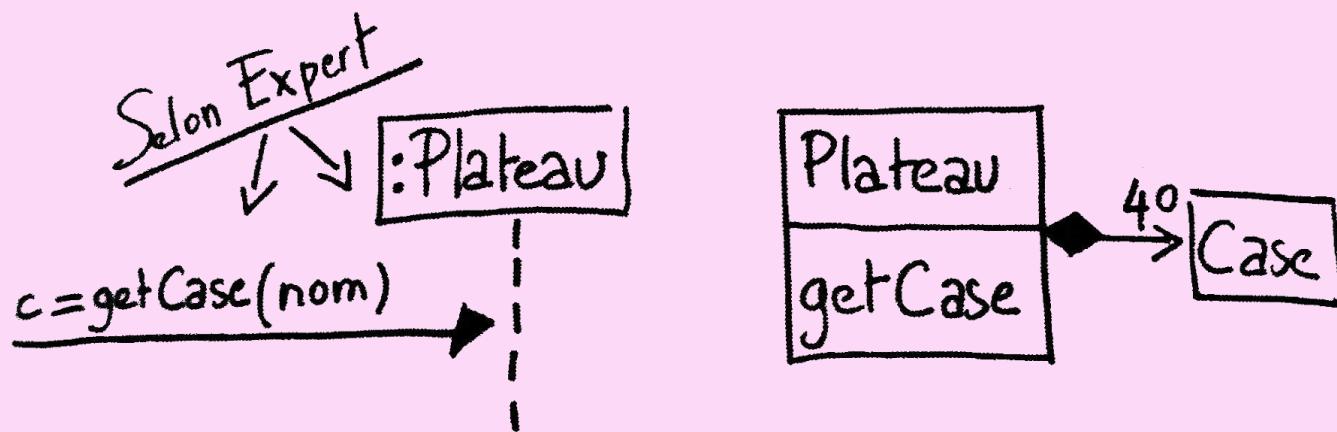
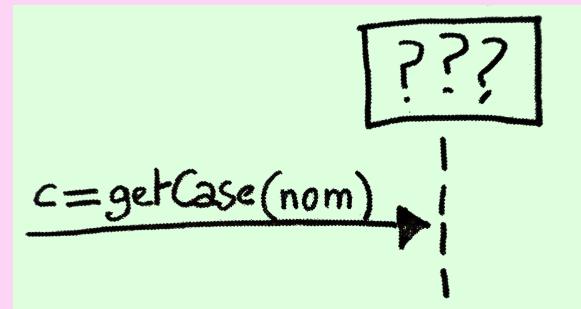
- (Sondage) Où mettre la méthode `getCase(nom)`?



# EXPERT



- Application du patron Expert



# EXERCICE RDCU



Created by Mohammed Ra

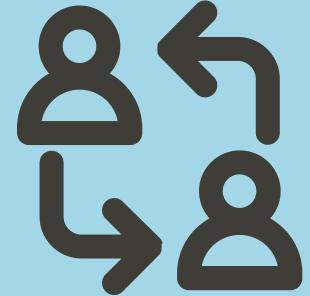
# (GOOGLE CLASSROOM)

Voir seance-03-exercice-reserverLivre



# SÉANCE #03

## RÉTROACTION: PAGE D'UNE MINUTE



Created by Prithvi  
from the Noun Project

1. Quels sont les deux [trois, quatre, cinq] plus importants [utiles, significatives, surprenantes, dérangeantes] choses que vous avez apprises au cours de cette session?
2. Quelle (s) question (s) reste (s) en tête dans votre esprit?
3. Y a-t-il quelque chose que tu n'as pas compris?

<https://1drv.ms/u/s!An6-F73ulxAOhVyiCB46jTeINVLS>

