


LOG210 SÉANCE #04

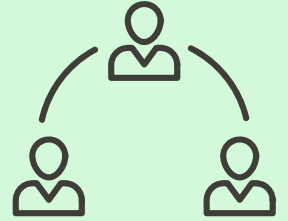
ANALYSE ET CONCEPTION DE LOGICIELS

1. **Administration** 
2. **Equipe** - Travail d'équipe - Humilité
3. **MDD** - MDD - Composition
4. **Contrats** - Contrats - list/map
5. **RDCU** - GRASP Couplage et Cohésion
6. **Exercice**

ADMINISTRIVIA

1. Invitation Lab 1+2+3 (GitHub Classroom)
 - ⚠ lien différent pour chaque groupe!
 - ne pas partager
 - problèmes de création d'équipe GitHub classroom (à signaler aux chargés sur Discord)
2. Il y aura un MDD pour le système SGB que vous devez utiliser pour votre MDD

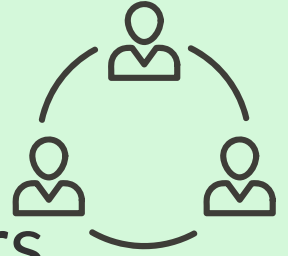
LOG210 SÉANCE #04



ANALYSE ET CONCEPTION DE LOGICIELS

1. **Administration**
2. **Equipe** - Travail d'équipe - Humilité ← S20203
3. **MDD** - MDD - Composition
4. **Contrats** - Contrats - list/map
5. **RDCU** - GRASP Couplage et Cohésion
6. **Exercice**

HUMILITÉ



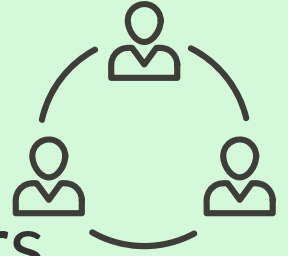
- Je ne suis pas le centre de l'univers.
- Je ne suis ni omniscient ni infailible

HUMILITÉ



- Je ne suis pas le centre de l'univers.
- Je ne suis ni omniscient ni infailible
- Je suis ouvert à m'améliorer

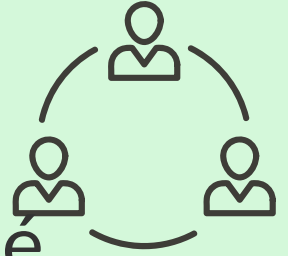
HUMILITÉ



- Je ne suis pas le centre de l'univers.
- Je ne suis ni omniscient ni infailible
- Je suis ouvert à m'améliorer

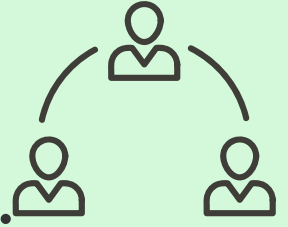
Moi < Équipe

HUMILITÉ EN PRATIQUE...



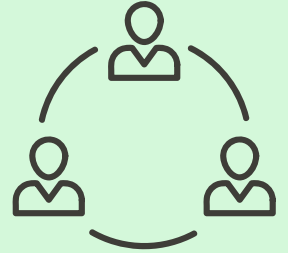
- Si quelqu'un travaille sans assez d'humilité
 - «Veux-tu descendre de ton piédestal?»
- Même si tu es le plus fort en JavaScript dans l'équipe, ne le mets pas en évidence constamment. C'est agaçant.
- Dans certaines cultures, l'humilité est très importante (p. ex. le confucianisme).

HUMILITÉ EN PRATIQUE...



- Apprendre à donner et à accepter les critiques **CONSTRUCTIVES**
- Les donner avec respect, p.ex. « J'ai de la misère à comprendre le flot de contrôle ici dans ton code » plutôt que « Ton code est mal écrit. »
- Les accepter avec humilité, p.ex. « Je comprends ton point de vue. Je vais refactoriser ce code en fin de semaine. »
- Savoir que son estime de soi n'équivaut pas à sa qualité de code.

HUMILITÉ EN PRATIQUE...




- Apprendre à être patient
- Chacun a sa personnalité et donc sa façon de déboguer, de concevoir, d'écrire du code
- Rester susceptible à l'influence des autres...
- Plus on est ouvert à être influencé, plus on *peut* influencer
- Plus on est vulnérable, plus on a l'air fort
- Si on veut être entendu, on doit d'abord écouter

LOG210 SÉANCE #04



ANALYSE ET CONCEPTION DE LOGICIELS

1. **Administration**
2. **Equipe** - Travail d'équipe - Humilité
3. **MDD** - MDD - Composition  **S20203**
4. **Contrats** - Contrats - list/map
5. **RDCU** - GRASP Couplage et Cohésion
6. **Exercice**

MDD



Correction de l'exercice

MDD?



- Attributs dérivées?

MDD?



- Attributs dérivées?
- Généralisation/spécialisation?

MDD?



- Attributs dérivées?
- Généralisation/spécialisation?
 - Règle « à 100% »?

MDD?



- Attributs dérivées?
- Généralisation/spécialisation?
 - Règle « à 100% »?
 - Règle « est un »?

MDD?



- Attributs dérivées?
- Généralisation/spécialisation?
 - Règle « à 100% »?
 - Règle « est un »?
 - Création d'une sous-classe conceptuelle?

MDD?



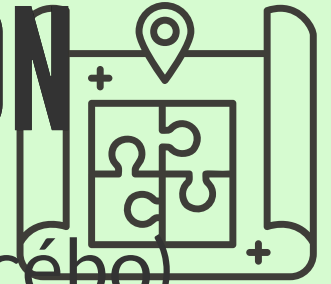
- Attributs dérivées?
- Généralisation/spécialisation?
 - Règle « à 100% »?
 - Règle « est un »?
 - Création d'une sous-classe conceptuelle?
 - Création d'une super-classe conceptuelle?

MDD?

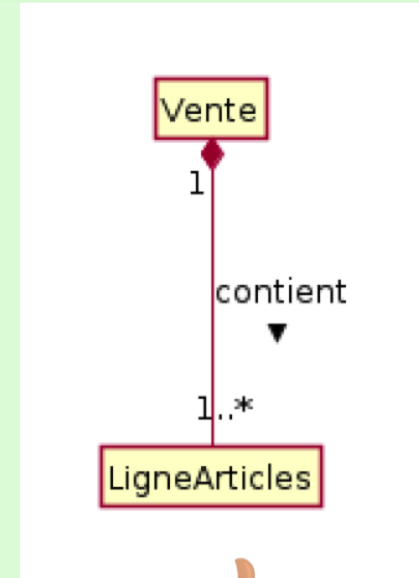
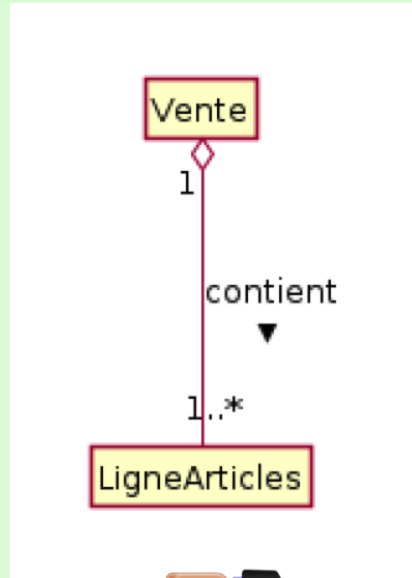
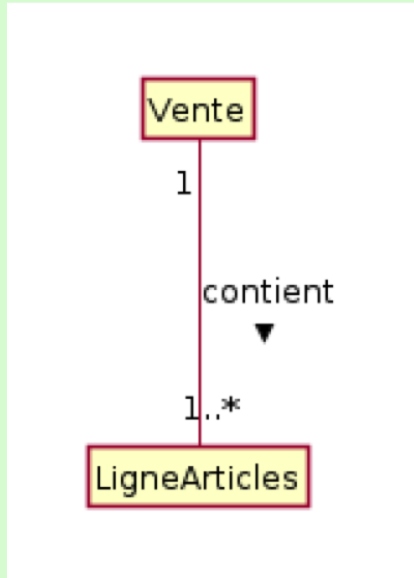


- Attributs dérivées?
- Généralisation/spécialisation?
 - Règle « à 100% »?
 - Règle « est un »?
 - Création d'une sous-classe conceptuelle?
 - Création d'une super-classe conceptuelle?
- Classe d'association

AGRÉGATION ET COMPOSITION



- N'utilisez pas l'agrégation (c'est un placebo)
- Utilisez la composition lorsque c'est approprié.



COMPOSITION?



COMPOSITION?



Durée de vie du composant est limitée à celle du composite;

COMPOSITION?



Durée de vie du composant est limitée à celle du composite;

- **dépendance de création-suppression** de la partie avec le tout.

COMPOSITION?



Durée de vie du composant est limitée à celle du composite;

- **dépendance de création-suppression** de la partie avec le tout.
- **Assemblage logique ou physique évident** entre le tout et les parties

COMPOSITION?



Durée de vie du composant est limitée à celle du composite;

- **dépendance de création-suppression** de la partie avec le tout.
- **Assemblage logique ou physique évident** entre le tout et les parties
- **Propriétés** du composite (ex. son emplacement s'étendent aux composants)

COMPOSITION?



Durée de vie du composant est limitée à celle du composite;

- **dépendance de création-suppression** de la partie avec le tout.
- **Assemblage logique ou physique évident** entre le tout et les parties
- **Propriétés** du composite (ex. son emplacement s'étendent aux composants)
- **Opérations** appliquées au composite (destruction, déplacement et enregistrement) se propagent aux composants

COMPOSITION



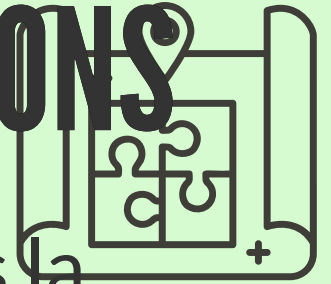
AVANTAGES

- Clarifie des contraintes du domaine
- Aide à identifier le **patron créateur**
- Opérations sont souvent propagées aux composants

INCONVÉNIENT

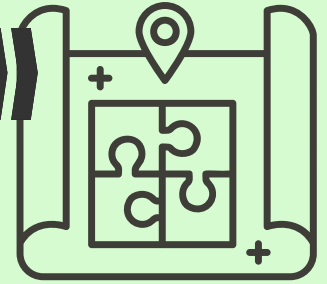
- «Chicanerie» - laisser tomber la composition en cas de doute ou de conflit dans l'équipe (voir **Bikeshedding**)

MODÉLISATION DES ASSOCIATIONS



- La création d'une instance ne signifie pas la création d'une association
- Association un à plusieurs (⚠ dans la solution)
 - Avec un attribut
 - Avec une map (tableau associatif)
 - Avec une liste

CLASSE DE « DESCRIPTION »

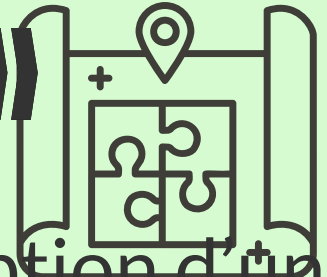


Article
description prix numéro de série codeArticle

Déconseillé

- Article représente un article physique
- Attributs prix, numéro de série et code
- Chaque fois qu'un article est vendu, l'objet Article correspondant est « supprimé »

CLASSE DE « DESCRIPTION »



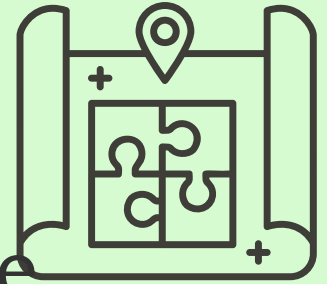
- Il est nécessaire de disposer de la description d'un produit ou d'un service, indépendamment de l'existence actuelle de ces derniers
- La suppression d'instances d'une entité qu'elle décrit entraîne la perte d'information qui doit être mémorisée
- Elle réduit la redondance ou la duplication des informations

CLASSES DE DESCRIPTION

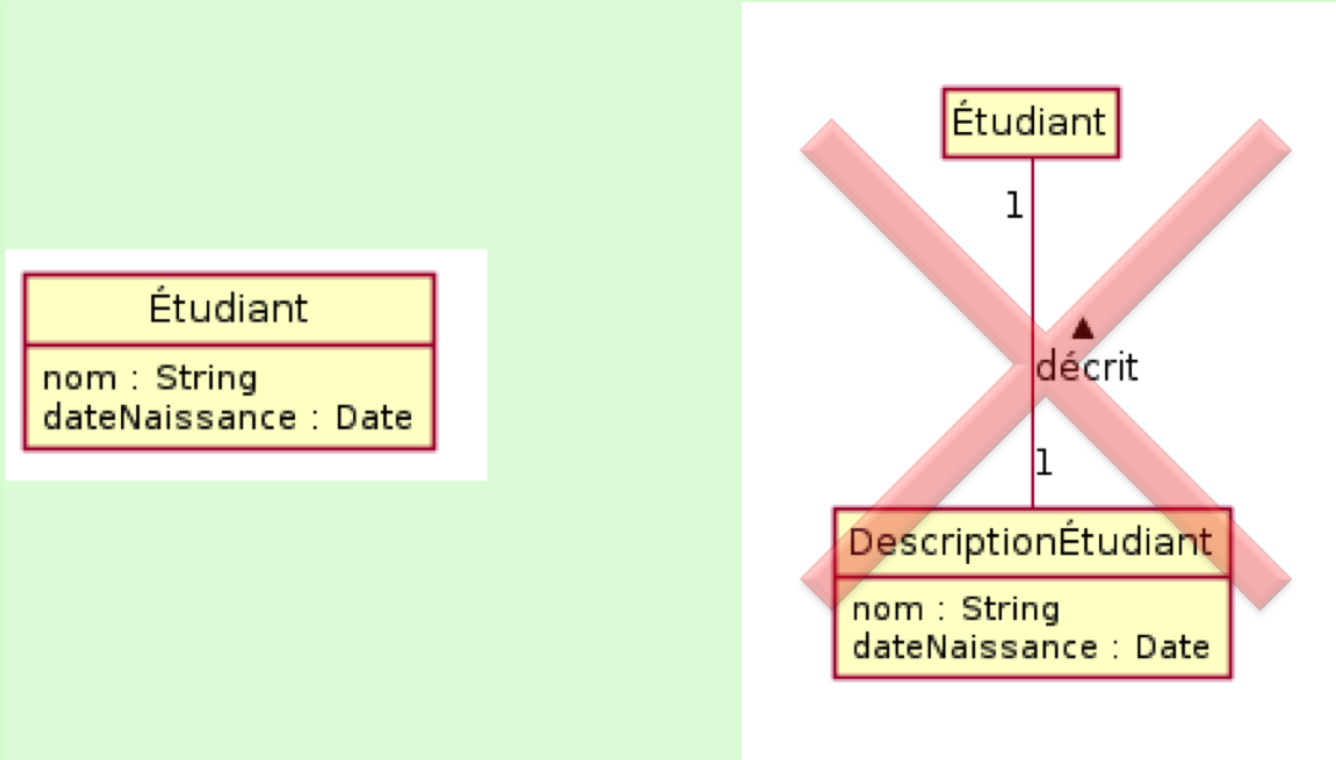


- Souvent, les objets d'une classe de description sont stockés dans un Catalogue
- Lire en détail la section 9.13 du livre pour d'autres exemples

CLASSES DE DESCRIPTION




- Attention : ne pas appliquer les classes de description n'importe comment:



LOG210 SÉANCE #04



ANALYSE ET CONCEPTION DE LOGICIELS

1. **Administration**
2. **Equipe** - Travail d'équipe - Humilité
3. **MDD** - MDD - Composition
4. **Contrats** - Contrats - list/map  **S20203**
5. **RDCU** - GRASP Couplage et Cohésion
6. **Exercice**

CONTRATS



Created by Mohammed Ra

Quel sont les trois type de postconditions?

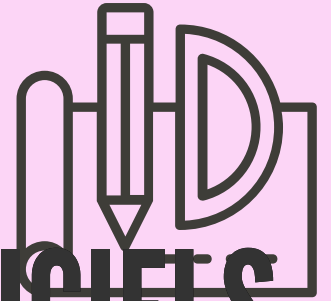
CONTRATS




Created by Mohammed Rai

- Postcondition:
 - « ... sur la base de correspondance avec *clé*. »
 - implique une multiplicité de 1 à plusieurs
 - implique l'utilisation d'une Map (tableau associatif)

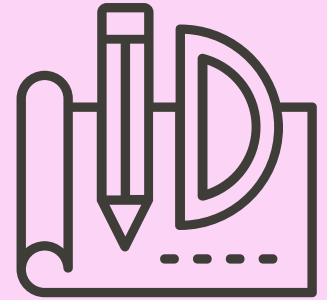
LOG210 SÉANCE #04



ANALYSE ET CONCEPTION DE LOGICIELS

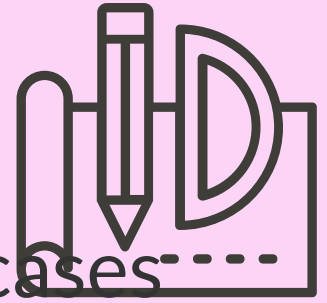
1. **Administration**
2. **Equipe** - Travail d'équipe - Humilité
3. **MDD** - MDD - Composition
4. **Contrats** - Contrats - list/map
5. **RDCU** - GRASP Couplage et Cohésion  **S20203**
6. **Exercice**

GRASP

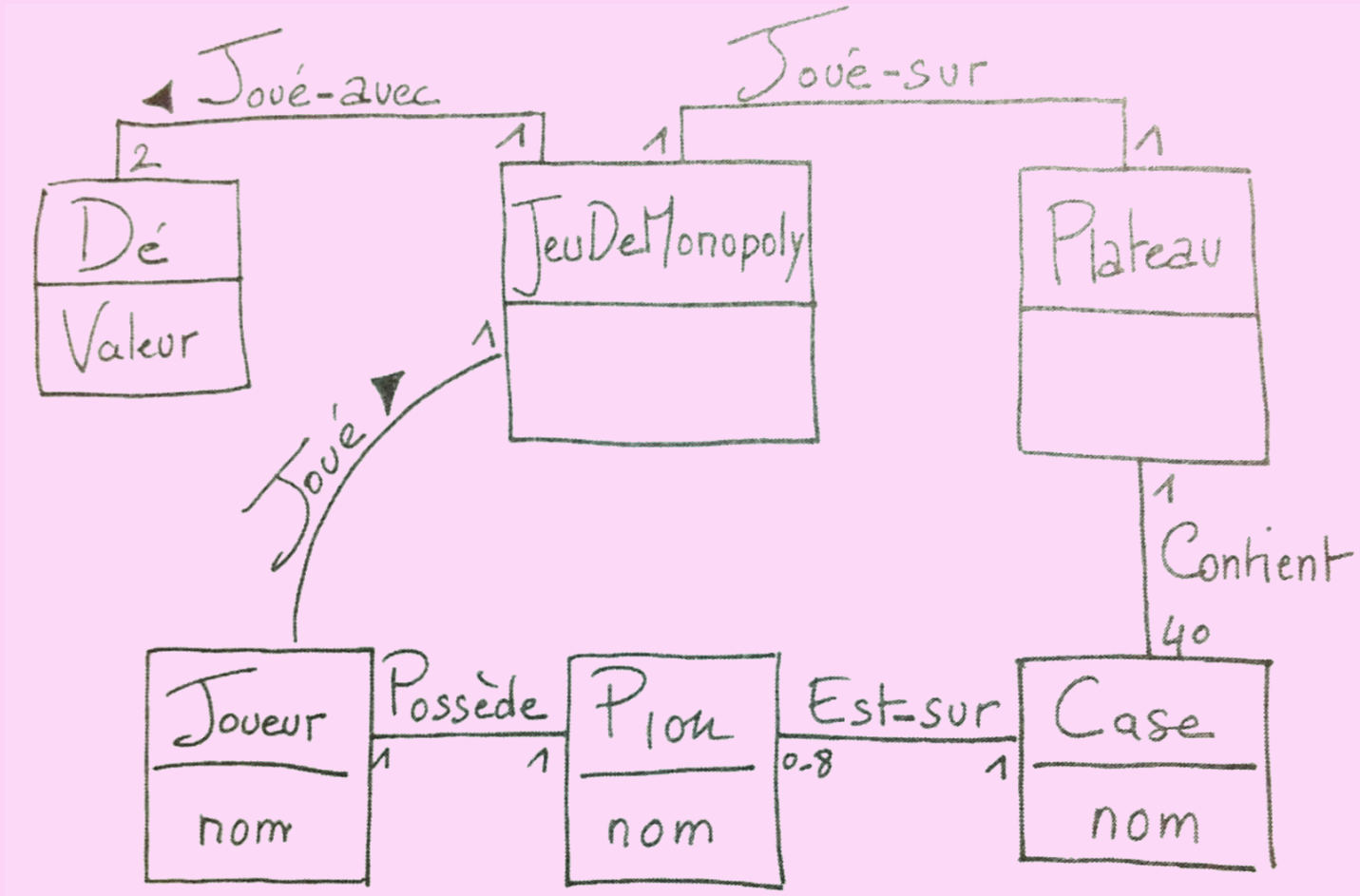


- Contrôleur
- Créateur
- Expert en information
- **Faible couplage**
- **Forte cohésion**
- Polymorphisme
- Indirection
- Protection de variation
- Fabrication pure

FAIBLE COUPLAGE



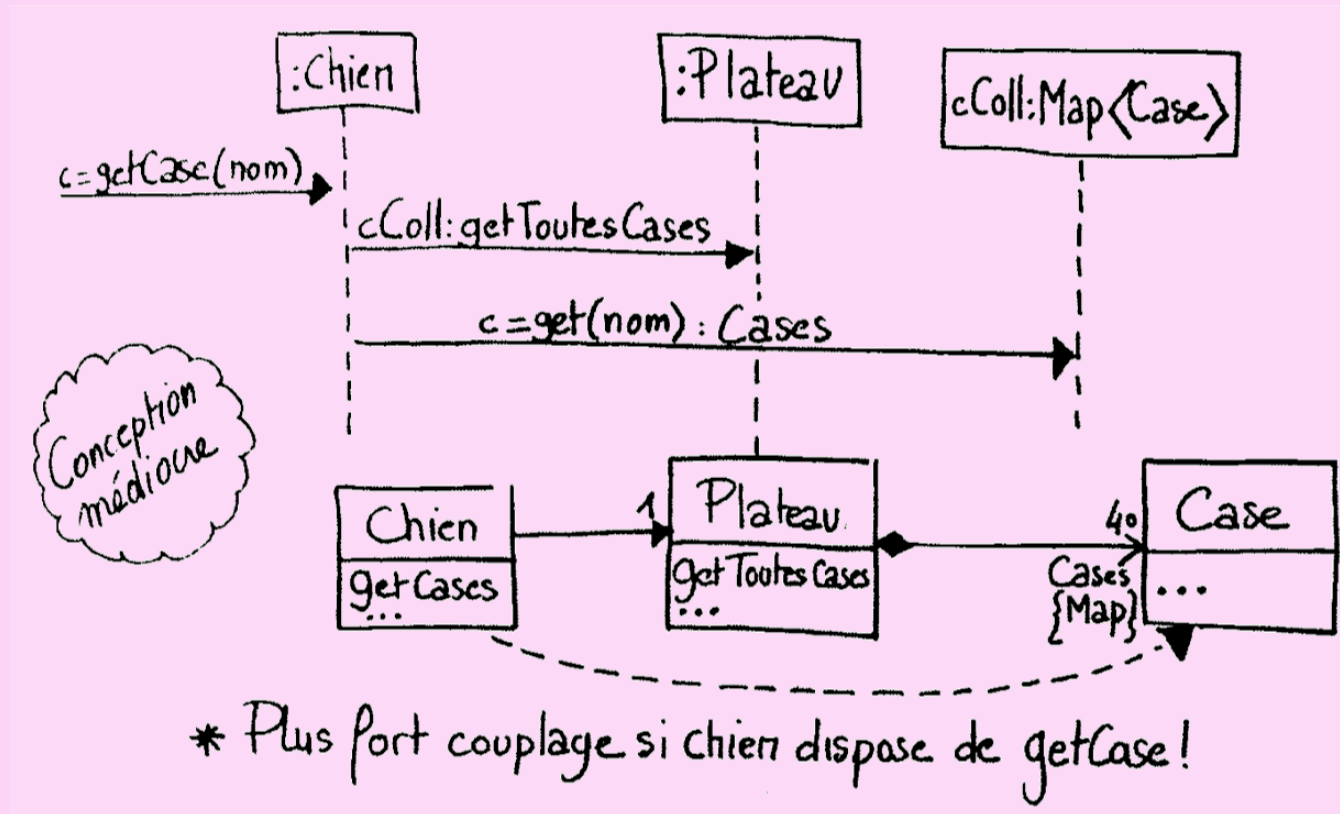
Question : Qui peut fournir la liste des cases...



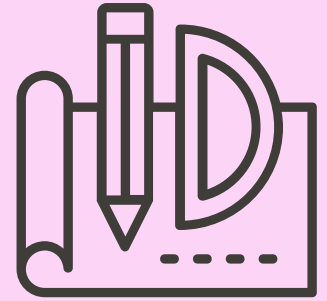
FAIBLE COUPLAGE



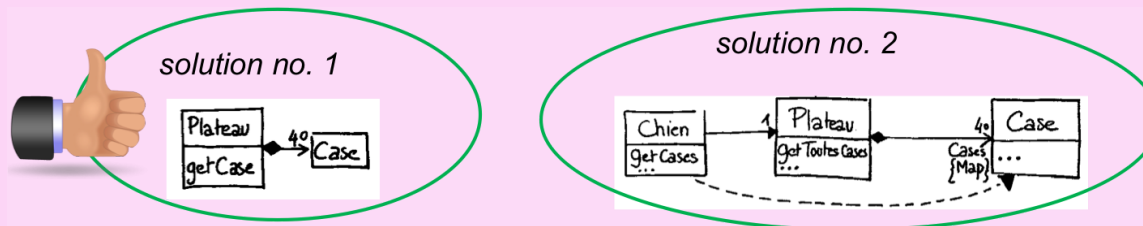
- Si la classe Pion (chien) avait la responsabilité



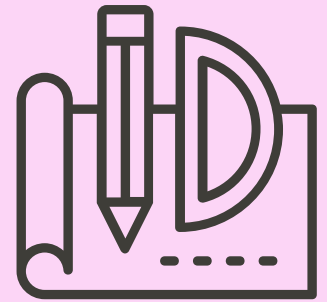
FAIBLE COUPLAGE



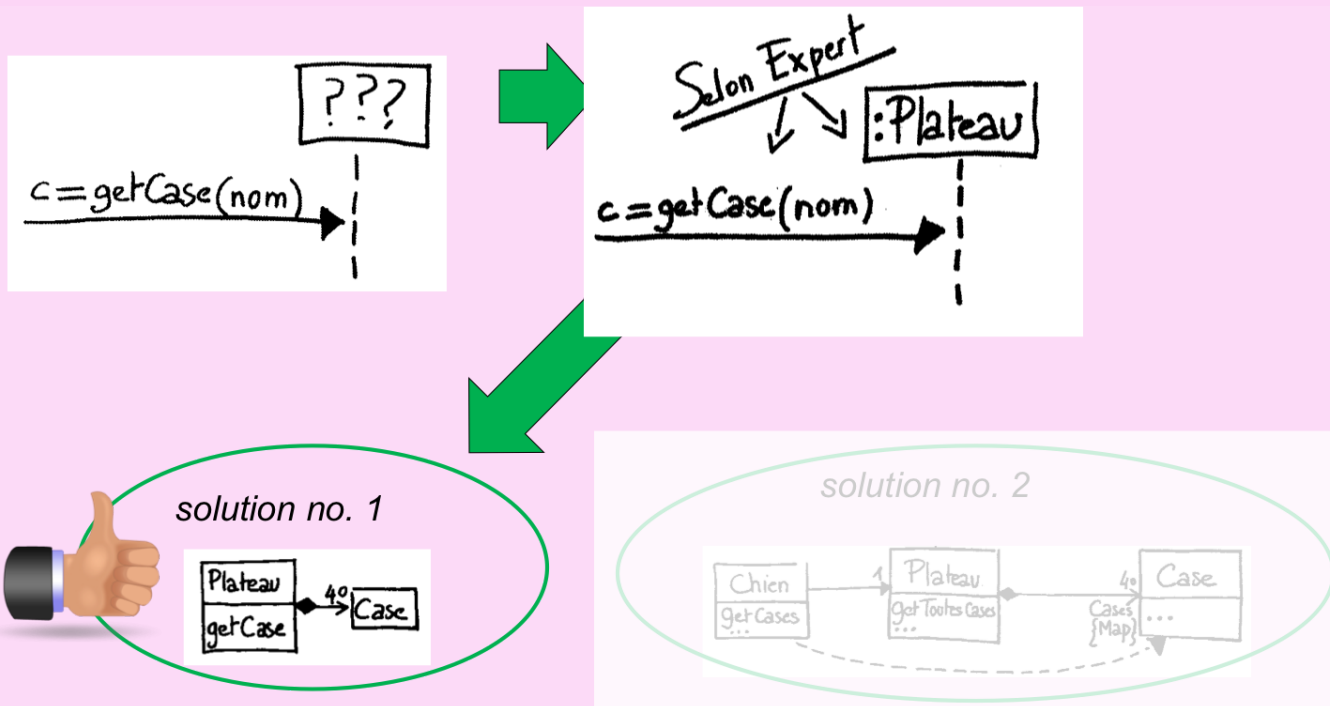
- Problème :
 - Comment réduire l'impact des modifications?
- Solution :
 - Assigner les responsabilités de sorte à **éviter tout couplage inutile.**
 - Appliquer ce principe pour **évaluer plusieurs solutions possibles.**



FAIBLE COUPLAGE

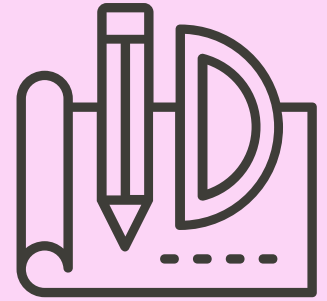


- Expert soutient Faible Couplage
- Expert évite de faire du couplage inutile, puisqu'un expert n'aura pas à demander à quelqu'un d'autre...



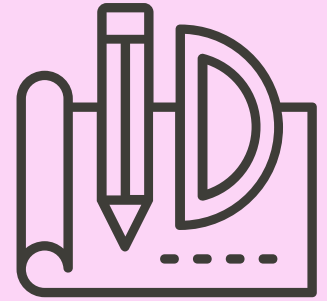
FAIBLE COUPLAGE

MOTIVATION

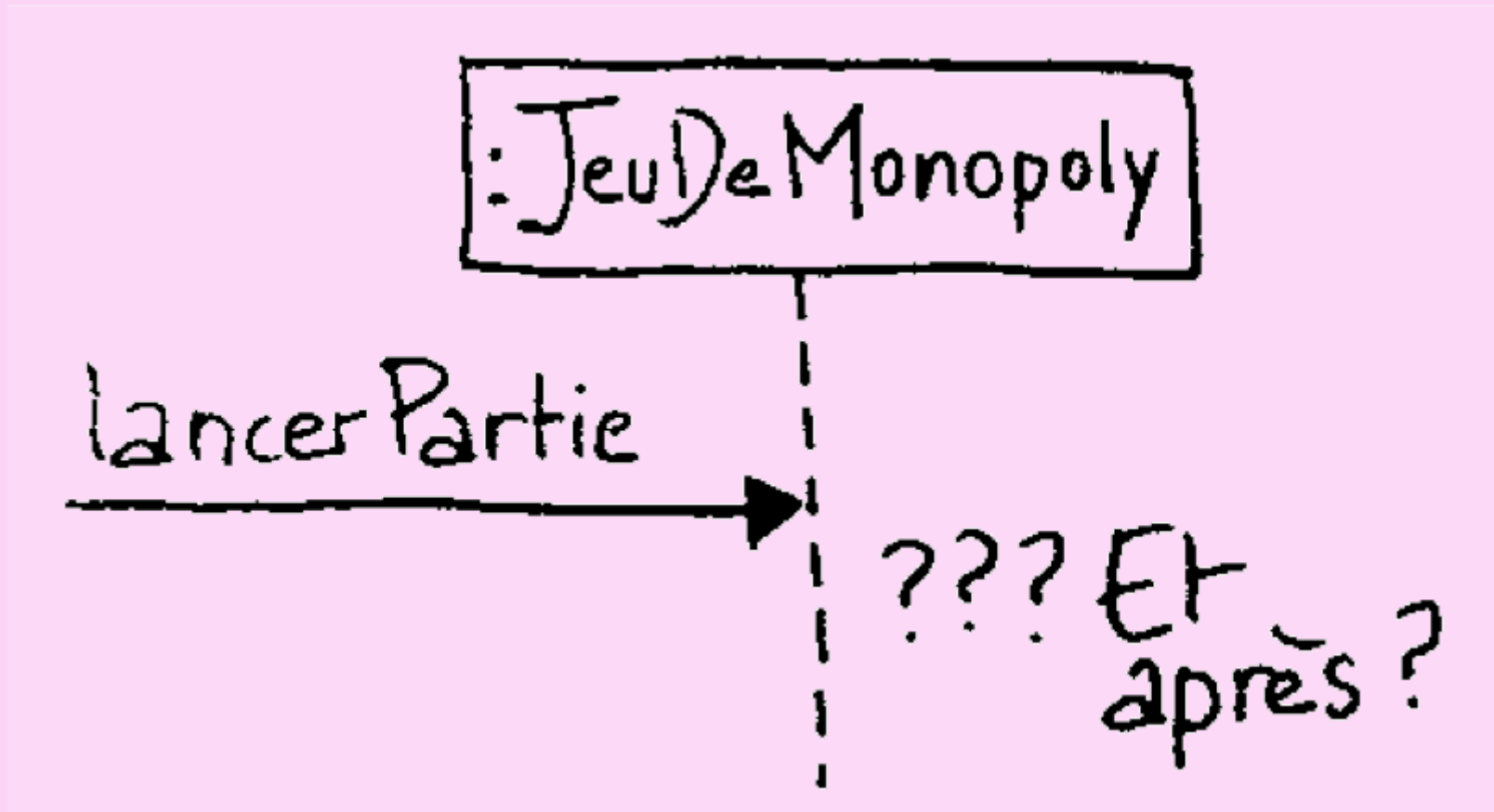


- Classes fortement couplées sont
 - difficiles à maintenir
 - difficiles à comprendre seules
 - difficiles à réutiliser

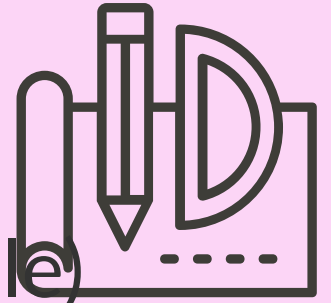
FORTE COHÉSION



- Poursuivons la conception



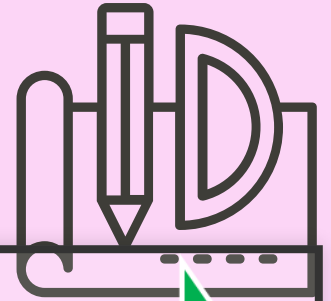
COHÉSION



- Mesure de combien une classe (ou module) supporte une seule responsabilité



COHÉSION



Faible

Beaucoup de fonctions

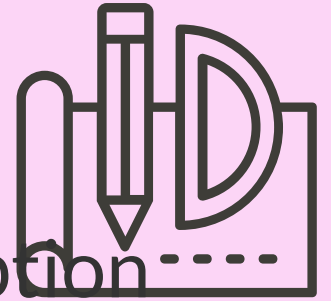
Responsabilités écartées

Forte

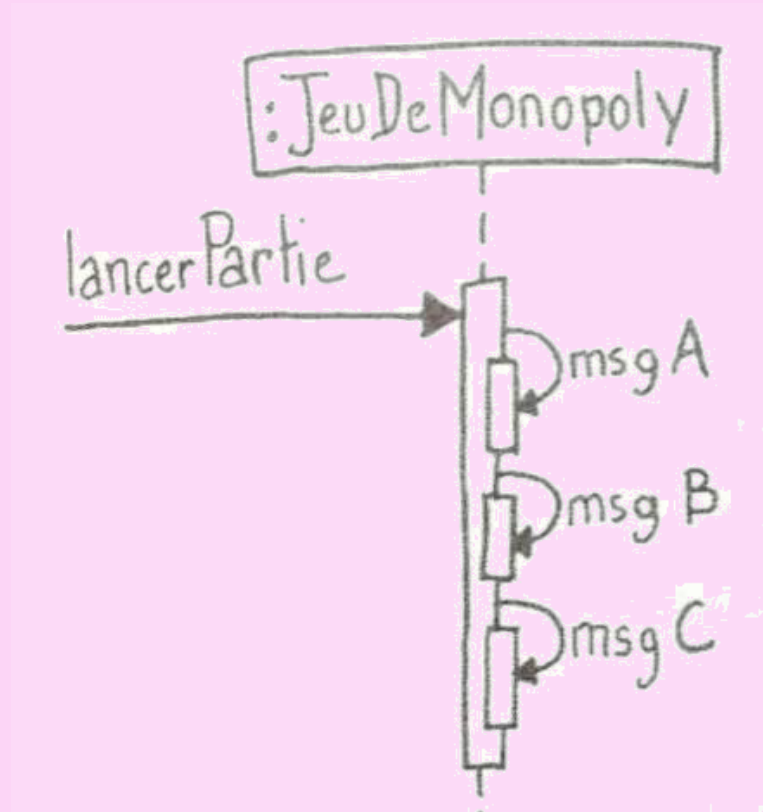
Peu de fonctions

Fonctions apparentées

FORTE COHÉSION



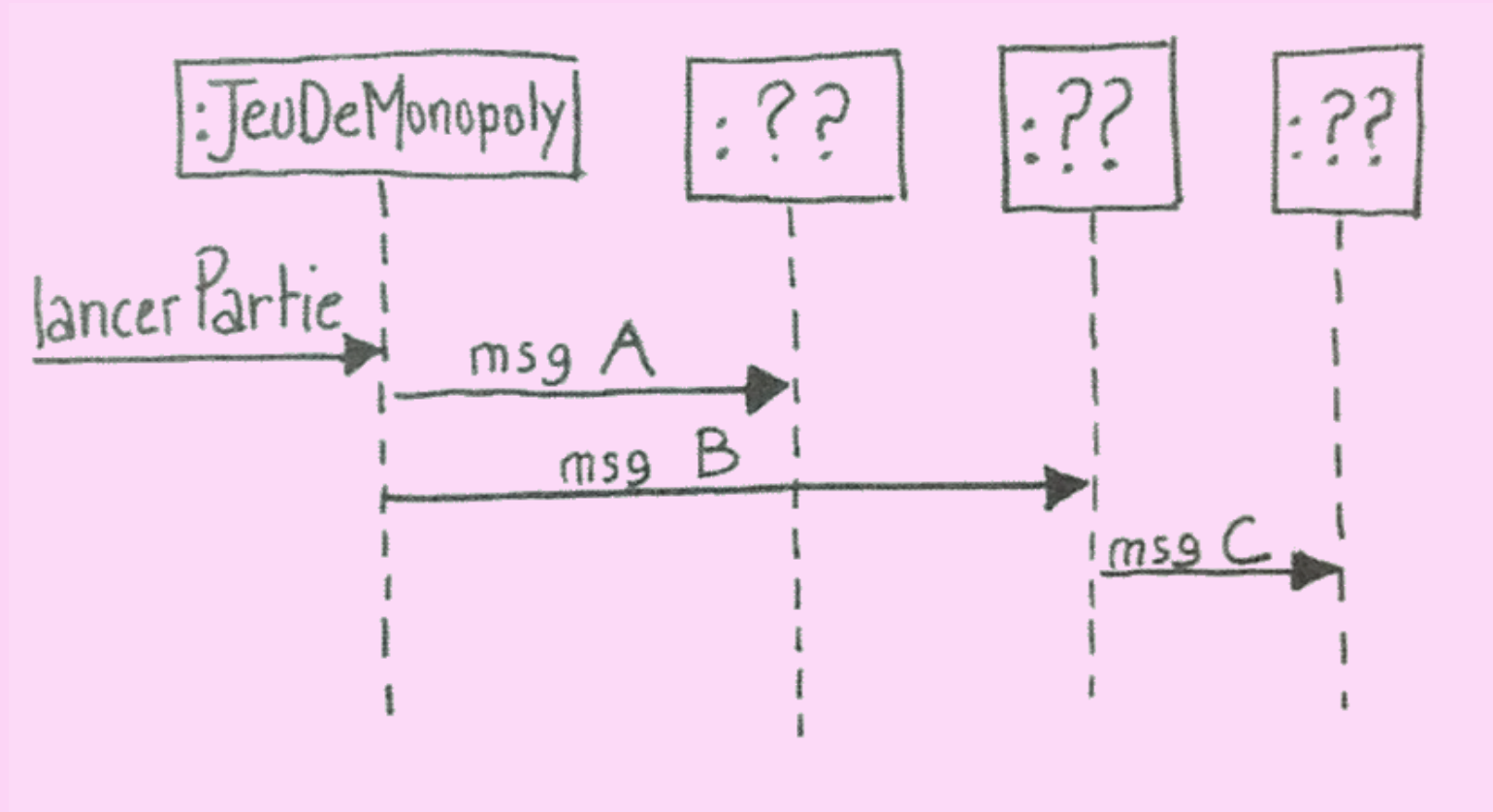
- Plusieurs choix opposés de conception



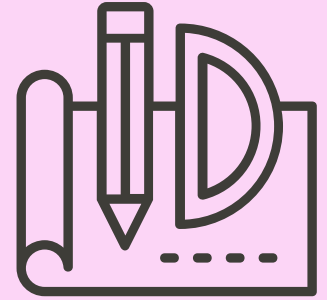
FORTE COHÉSION



- JeuDeMonopoly délègue les responsabilités



FORTE COHÉSION

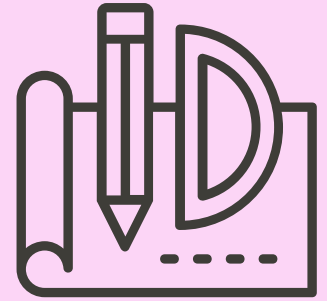


1. Problème :

- Comment s'assurer que les objets restent compréhensibles et faciles à gérer, et, bénéfice second, qu'ils contribuent au Faible Couplage?

2. Solution :

FORTE COHÉSION



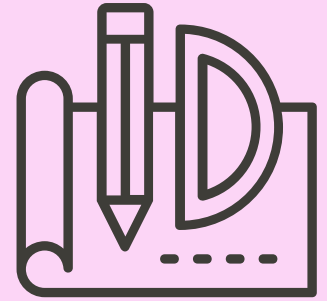
1. Problème :

- Comment s'assurer que les objets restent compréhensibles et faciles à gérer, et, bénéfice second, qu'ils contribuent au Faible Couplage?

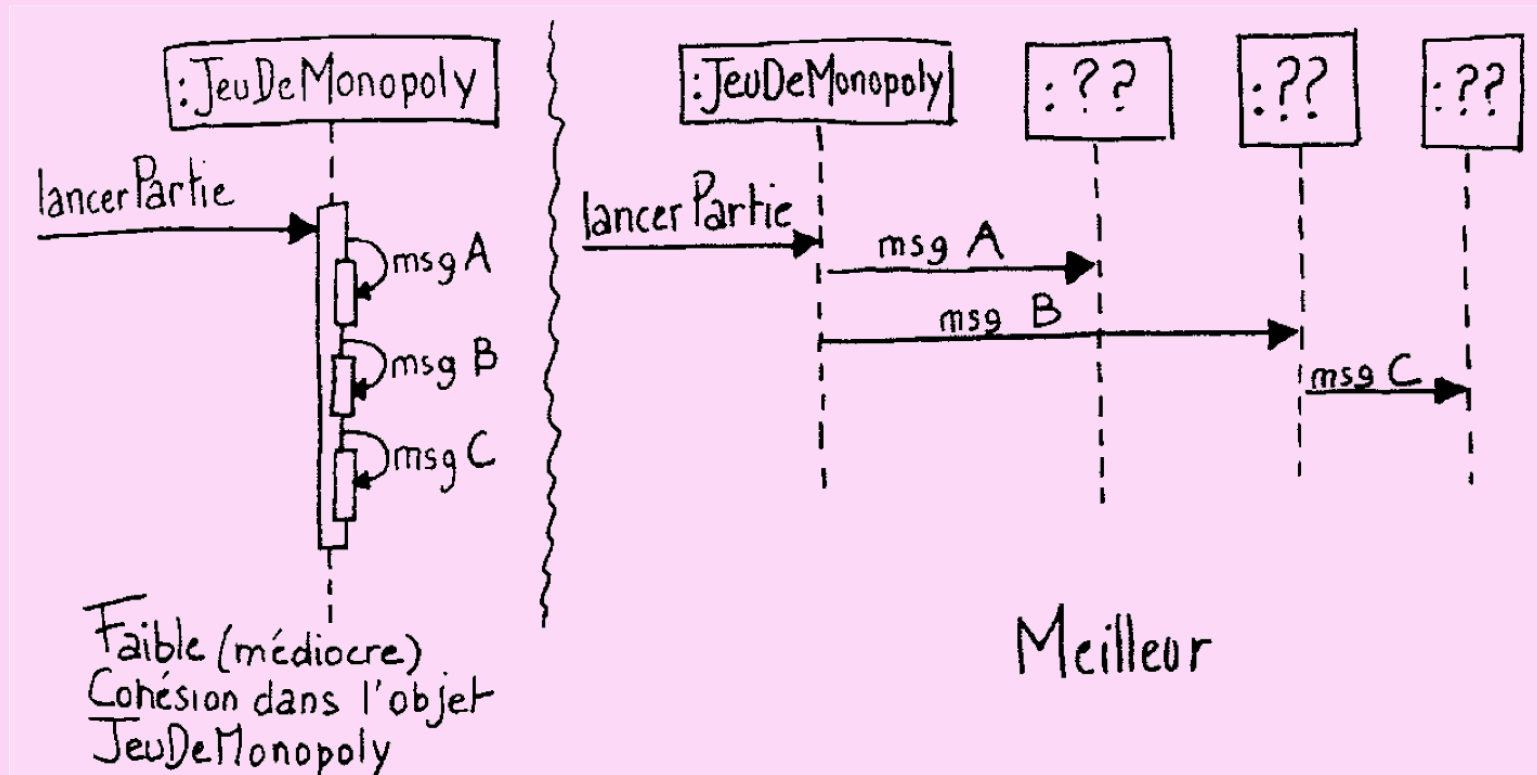
2. Solution :

- Assigner les responsabilités de sorte que la cohésion demeure élevée. Appliquer ce principe pour évaluer les solutions possibles.

FORTE COHÉSION



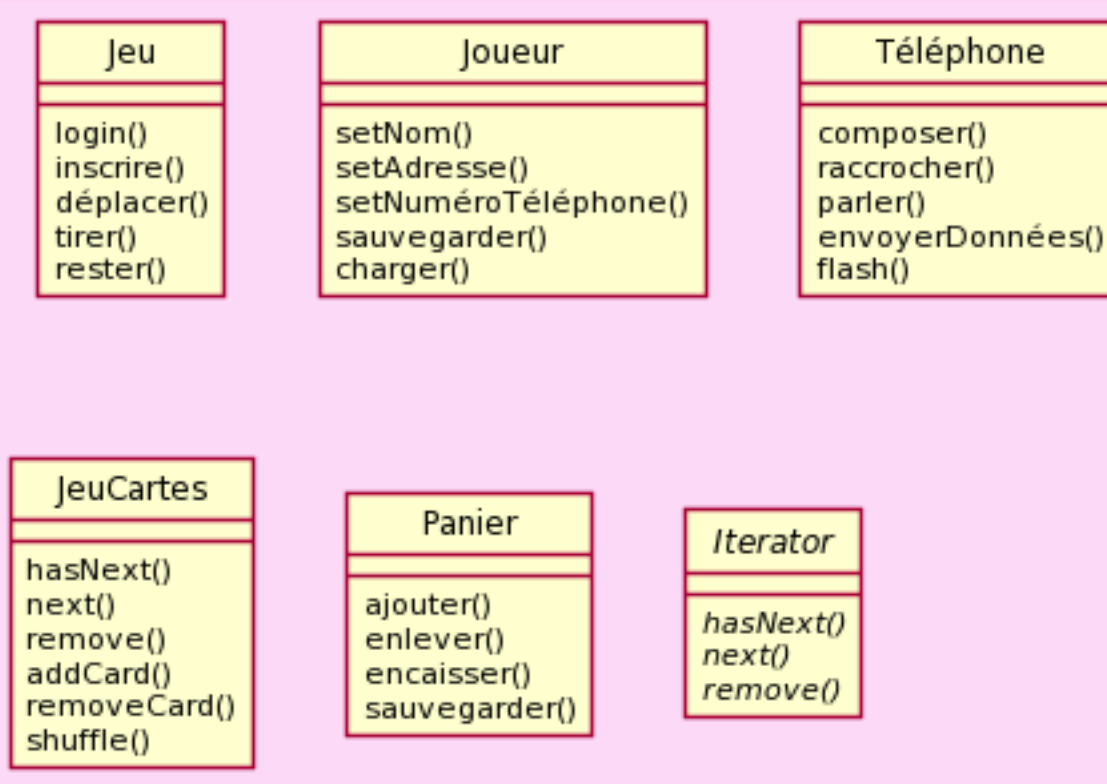
- Comparaison des choix



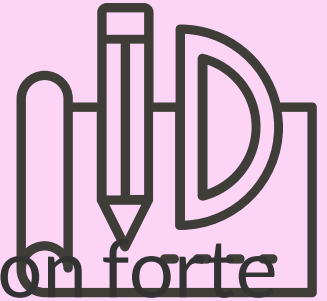
COHÉSION: EXERCICE



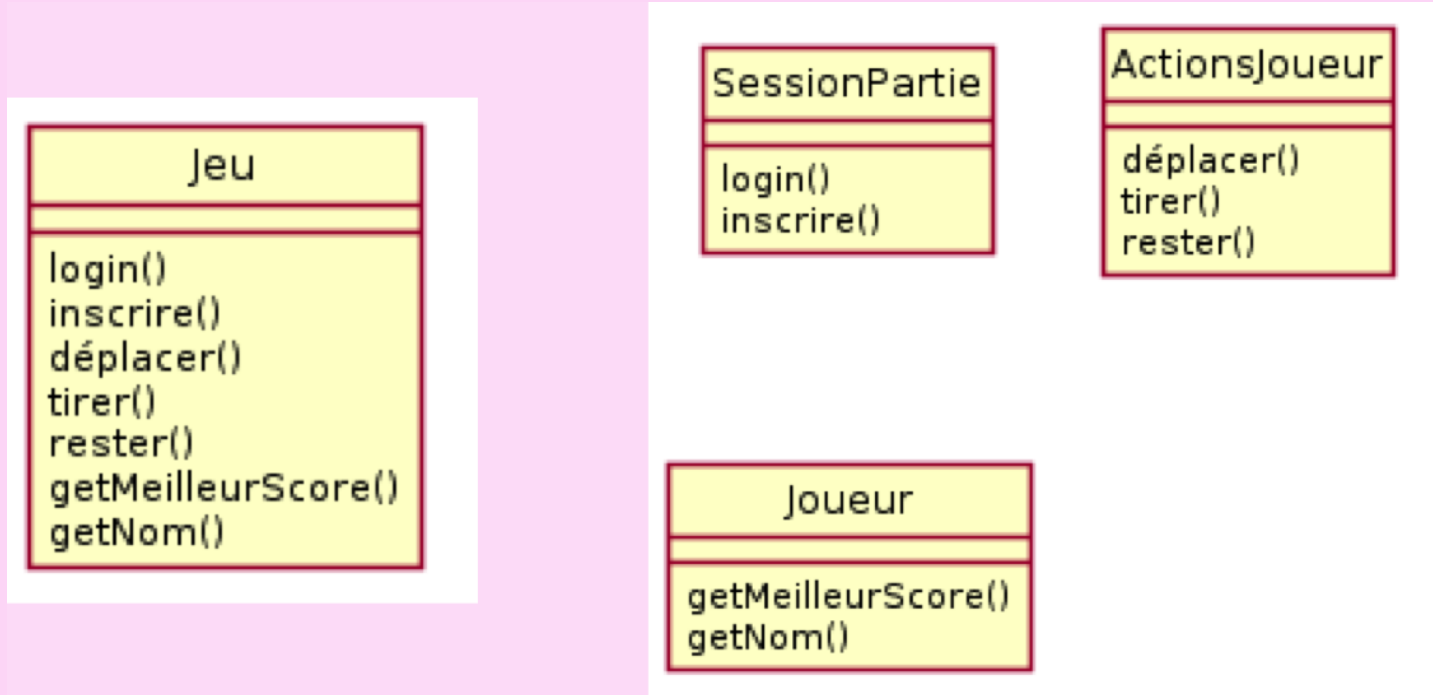
- Lesquelles des classes suivantes ont de multiples responsabilités?



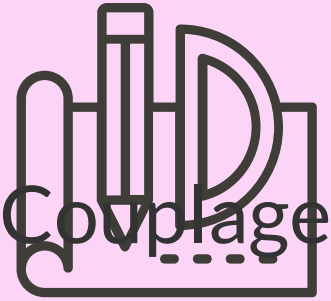
COHÉSION: EXERCICE



- Déterminez si ces classes ont une cohésion forte ou faible:



VALIDATION DE LA COMPRÉHENSION




Les patrons GRASP Forte Cohésion et Faible Couplage servent principalement à

1. concevoir des classes avec beaucoup de responsabilités
2. choisir les classes dans la couche applicative pour traiter une opération système
3. évaluer plusieurs possibilités lors de la réalisation d'une dynamique de conception
4. choisir une bonne classe pour créer des instances d'une autre classe

LOG210 SÉANCE #04



ANALYSE ET CONCEPTION DE LOGICIELS

1. **Administration**
2. **Equipe** - Travail d'équipe - Humilité
3. **MDD** - MDD - Composition
4. **Contrats** - Contrats - list/map
5. **RDCU** - GRASP Couplage et Cohésion
6. **Exercice** 

EXERCICE



- En équipe de 4 personnes, réaliser les modèles suivant:
 - MDD(avec catégorie de classe et d'association),
 - DSS,
 - Contrats,
 - RDCU
 - DCL

ÉTUDE DE CAS



SYSTÈME D'ÉCHANGE DE LIVRES UNIVERSITAIRES

Le Bureau du Développement Durable (BDD) de l'Université a mis en place un système d'échange de livres aux fins de développement durable et pour réduire les coûts pour les étudiants (les clients du système). La version initiale est rudimentaire et ne permet que deux fonctionnalités :

CU01 - AJOUTER UN LIVRE À ÉCHANGER



Acteur principal : Client (étudiant)

Préconditions :

Le Client est identifié (par son nom d'utilisateur) et authentifié par son mot de passe.

Scénario principal (succès)

1. Le Client démarre un nouvel ajout de livre.
2. Le Client entre le code ISBN du livre, ainsi que le code de sa condition.
3. Le Système enregistre le livre et présente sa description (titre, nombre de pages, auteurs, maison d'édition, no d'édition). Le Client répète les étapes 2 à 3 jusqu'à ce qu'il ait saisi tous les livres à échanger.
4. Le Système présente la liste de livres que possède le Client.

Cas alternatifs

3a. Le système affiche un message d'erreur puisque le livre n'existe pas.



Acteur principal : Client (étudiant)

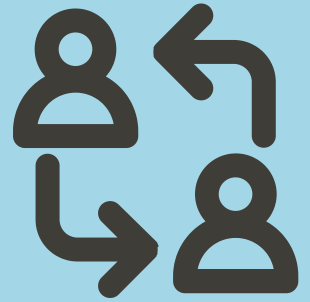
Préconditions : Le Client est identifié (par son nom d'utilisateur) et authentifié par son mot de passe.

Scénario principal (succès)

1. Le Client démarre une nouvelle proposition d'échange de livres.
2. Le Système présente une liste d'autres Clients dans le système ainsi que le(s) livre(s) qu'ils ont à échanger.
3. Le Client sélectionne un autre Client (le Client Proposé) à qui il veut proposer un échange.
4. Le Système présente la liste de livres que possède le Client Proposé et une liste de livres que possède le Client.
5. Le Client ajoute à la proposition d'échange un livre. Si c'est un de ses livres, alors c'est à offrir dans la proposition. Sinon c'est un livre du Client Proposé et c'est à recevoir dans la proposition. Le Client répète l'étape 5. jusqu'à ce qu'il soit satisfait de la proposition.
6. Le Système présente le nombre total de livres à offrir et à recevoir et demande au Client de confirmer la proposition.
7. Le Client confirme et le Système enregistre sa proposition d'échange avec la date et l'heure.
8. Le Système envoie un courriel au Client Proposé pour l'informer de la proposition d'échange.

SÉANCE #04

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>

