Les patterns de création

- Pour créer des objets : *new*
  - *new* permet l'instanciation de classes concrètes uniquement
    - *new* induit un couplage fort entre la classe « cliente » et la classe qui définit l'objet créé
      - Donc, en cas d'évolution de la classe qui définit l'objet créé, le code de la classe cliente doit être repris
  - Créer l'instance, ce n'est pas le « métier » de la classe cliente !
  - La création ne doit pas toujours être libre et publique

  ☞Séparer ce qui peut évoluer de ce qui ne change pas et (bien) distribuer les responsabilités, donc externaliser la code de création

- Principal objectif des patterns de création
  - Réduire le couplage entre classe « cliente » (c.-à-d. demandeuse de la création) et classe qui définit l'objet créé
  - Séparer les responsabilités (= préoccupations, métiers)

Une bonne pratique

- Fabrique simple
  - La « responsabilité » (savoir-faire) de créer est externalisée
    - Le code est extrait de la classe cliente
    - Il est placé dans une classe dédiée : la « fabrique simple »
      - Méthode de classe
      - Ou méthode d'un objet « fabrique » (qui opère par délégation)
  - Avantages
    - Réduction du couplage
    - Point de maintenance unique (« Don't Repeat Yourself »)
    - Séparation des responsabilités
  - Ce n'est pas un design pattern du GoF, mais plutôt une bonne pratique (GRASP)