Le modèle « Fabrique abstraite » (1/4)

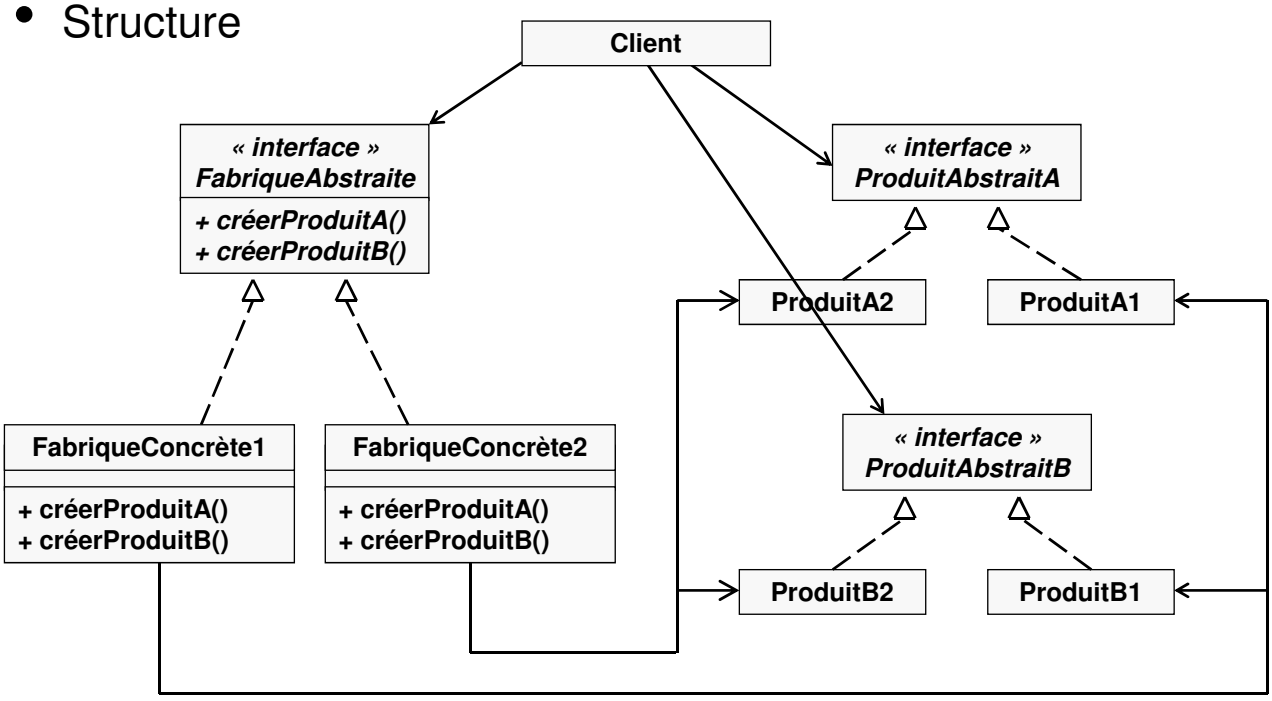
- Fabrique abstraite
  - Catégorie « créateur », de niveau objet
- Intention
  - Permet la création d'objets regroupés en familles sans avoir à spécifier (connaître) leurs classes concrètes
  - Au moyen d'un objet « fabrique » (comme la fabrique simple)
- Motivation
  - Par exemple, quand on choisit un certain style graphique pour une IHM, on veut créer des objets graphiques (fenêtres, boutons…) conformes à ce style là (famille)
  - On veut éviter de coder « en dur » dans la classe « cliente » la création d'objets (new) en faisant référence à leurs classes concrètes
    - Rendre la classe « cliente » indépendante de la façon dont les objets sont créés, afin de faciliter l'évolution

Le modèle « Fabrique abstraite » (2/4)

- Participants
  - *FabriqueAbstraite* est une interface qui spécifie les méthodes de création des différents objets
  - *FabriqueConcrète1*, *FabriqueConcrète2*… sont les classes concrètes qui implantent *FabriqueAbstraite* pour chaque famille
  - *ProduitAbstraitA*, *ProduitAbstraitB*… sont des interfaces ou des classes abstraites qui définissent les objets de la famille A, B…
  - *ProduitConcretA1*, *ProduitConcretA2* implante (ou hérite de) *ProduitAbstraitA* pour chaque famille de produits
  - *Client* est la classe qui utilise l'interface de *FabriqueAbstraite*
- Collaborations
  - La classe *Client* utilise une instance d'une des fabriques concrètes pour créer les produits (via l'interface de *FabriqueAbstraite*)

Le modèle « Fabrique abstraite » (3/4)

- Structure



Le modèle « Fabrique abstraite » (4/4)

- Conséquences
  - L'interface de « Fabrique abstraite » change dès qu'on introduit un nouveau produit ☹ (mais l'objectif n'est pas là…)
  - Mais pas quand on ajoute une nouvelle famille ☺
- Modèles apparentés
  - « Factory Method » permet d'implémenter les méthodes des fabriques de création des produits
    - Dans notre exemple, FabriqueConcrète1 (idem pour FabriqueConcrète2) serait la classe « Créateur » et créerProduitA() (idem pour créerProduitB()) serait la méthode générique qui fait appel à une méthode de fabrication spécifique au produit A de la famille 1.