Le modèle « Factory Method » (1/8)

- Nom
  - Alias « Fabrication »
  - Catégorie « créateur », de niveau classe
- Intention
  - Définir une interface pour la « fabrication » d'objets sans connaître le type réel (classe) de ces objets
    - Permettre à une application de fabriquer des objets dont le type réel peut varier
  - Reporter la création effective dans les sous-classes
    - Utilisation de l'héritage

Le modèle « Factory Method » (2/8)

- Motivation
  - Ne pas faire appel à *new*
  - Par exemple, dans une bibliothèque graphique pour pouvoir créer des formes géométriques qui seront définies par l'utilisateur de la bibliothèque (le concepteur de la bibliothèque ignore la classe concrète des objets à créer)

Le modèle « Factory Method » (3/8)

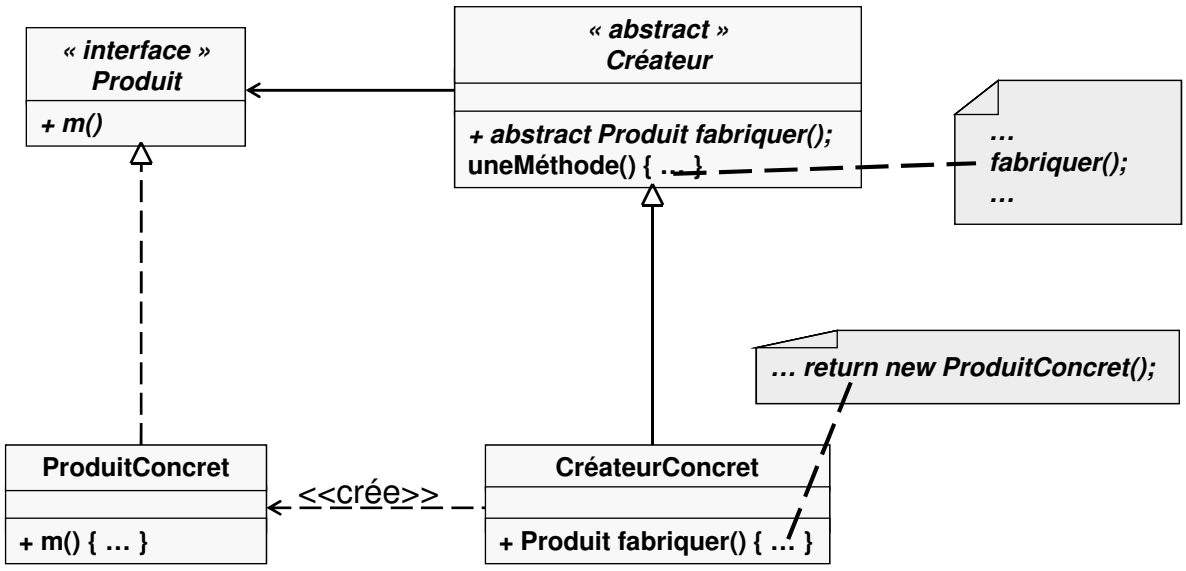
- Indication d'utilisation
  - Une classe ne connait pas la classe (concrète) des objets à créer
  - Une classe attend de ses sous-classes qu'elles spécifient les objets qu'elles créent
  - Le pattern « Factory Method » introduit une classe abstraite qui offre une méthode abstraite de fabrication
    - Comme patron de méthode
    - Implantée dans une sous-classe à qui est délégué le choix du type de l'objet à créer
    - Retourne un produit
      - Une classe abstraite (un super-type) représente le type du produit

Le modèle « Factory Method » (4/8)

- Participants
  - *Produit* : définit l'interface des objets à créer
  - *ProduitConcret* : implémente l'interface *Produit*
  - *Créateur* : déclare l'interface de fabrication qui retourne un *Produit*
  - *CréateurConcret* : définit (ou redéfinit) la fabrication pour retourner une instance de *ProduitConcret*
- Collaboration
  - C'est la sous-classe *CréateurConcret* qui réalise la fabrication d'un *ProduitConcret* (vu comme un *Produit*)

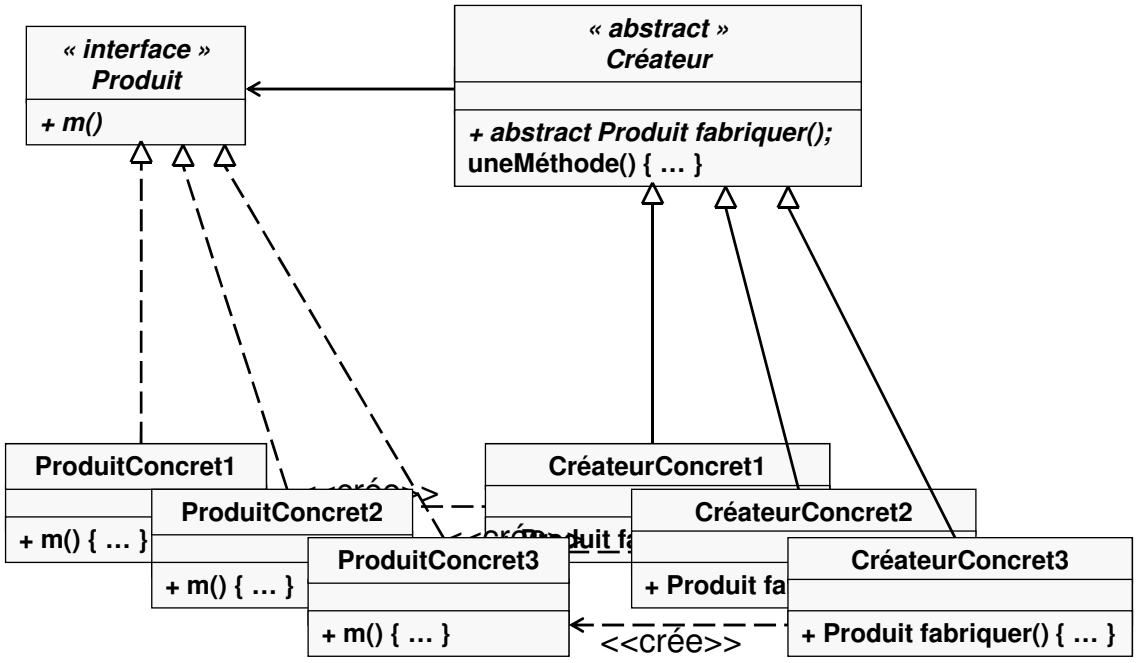
Le modèle « Factory Method » (5/8)

- Structure



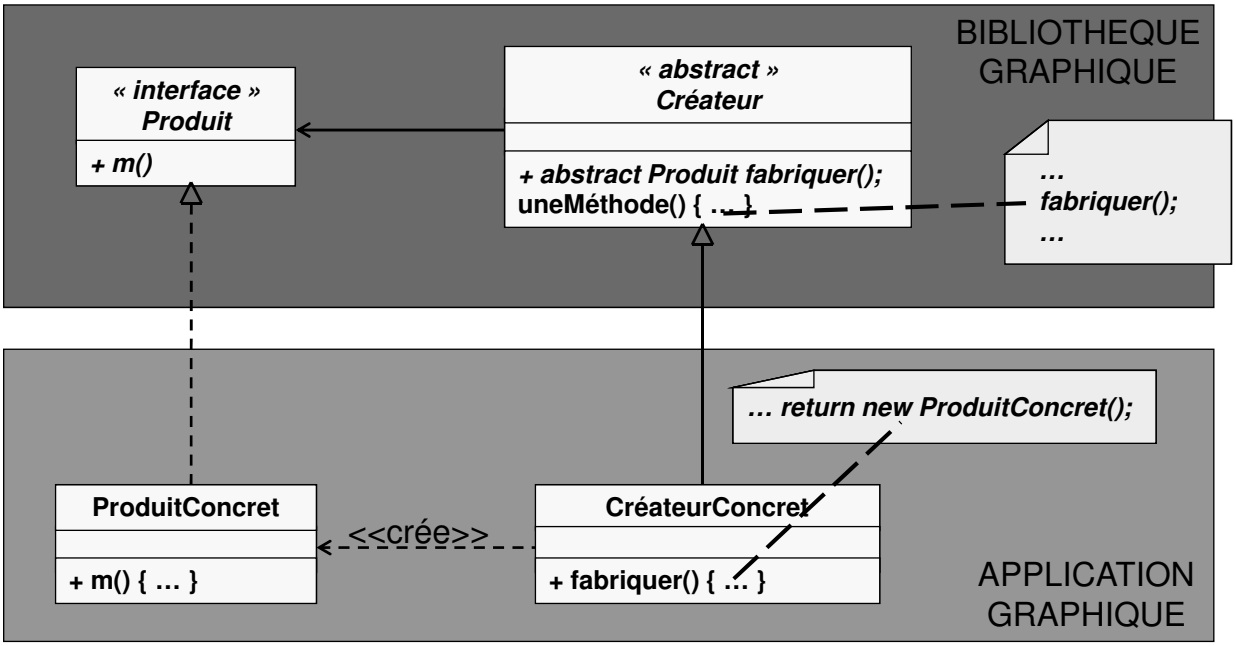
Le modèle « Factory Method » (6/8)

- Exemple



Le modèle « Factory Method » (7/8)

- Utilisation remarquable



Le modèle « Factory Method » (8/8)

- Implémentation
  - *Créateur* peut aussi définir une implémentation par défaut de *fabriquer()*
    - *Créateur* peut donc être une classe concrète
  - La requête de fabrication peut être paramétrée afin de permettre la création de plusieurs (nombreux) types de produits concrets