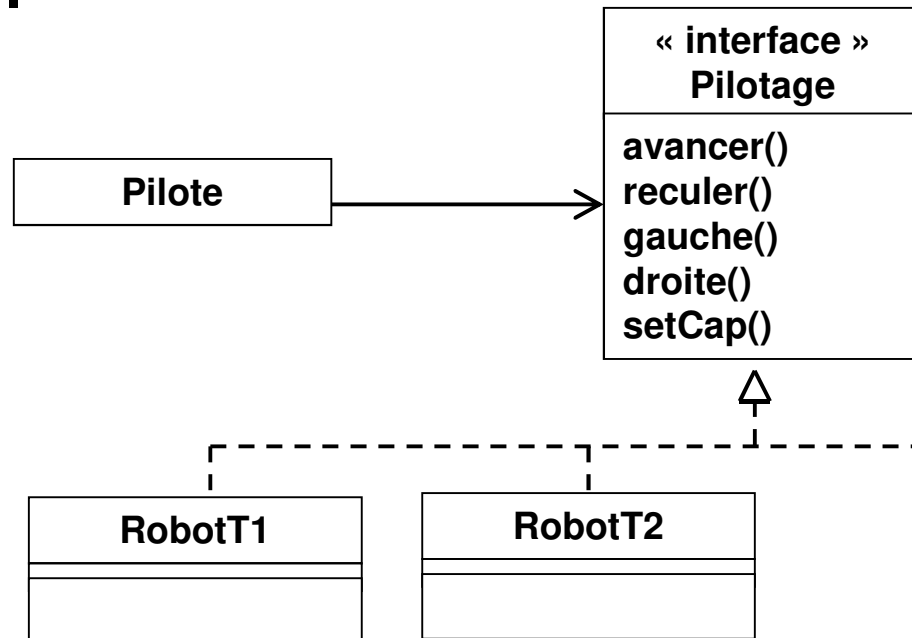


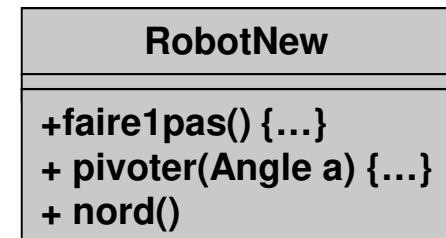
## *Un autre problème...*

- On considère une application existante
  - Des robots terrestres de différents types (provenant p. ex. de différents fabricants)
  - Qui peuvent être commandés (pilotés) via leur interface de pilotage
    - avancer et reculer d'un nombre de pas donnés
    - tourner de 90° à droite ou à gauche
    - changer de cap
  - Des pilotes de robots utilisent les commandes des robots
- Comment étendre l'application ? (évolution)
  - Pour intégrer un nouveau type de robot qui offre une interface de commande différente
    - avancer d'un pas
    - pivoter d'un angle donné
    - s'orienter vers le nord

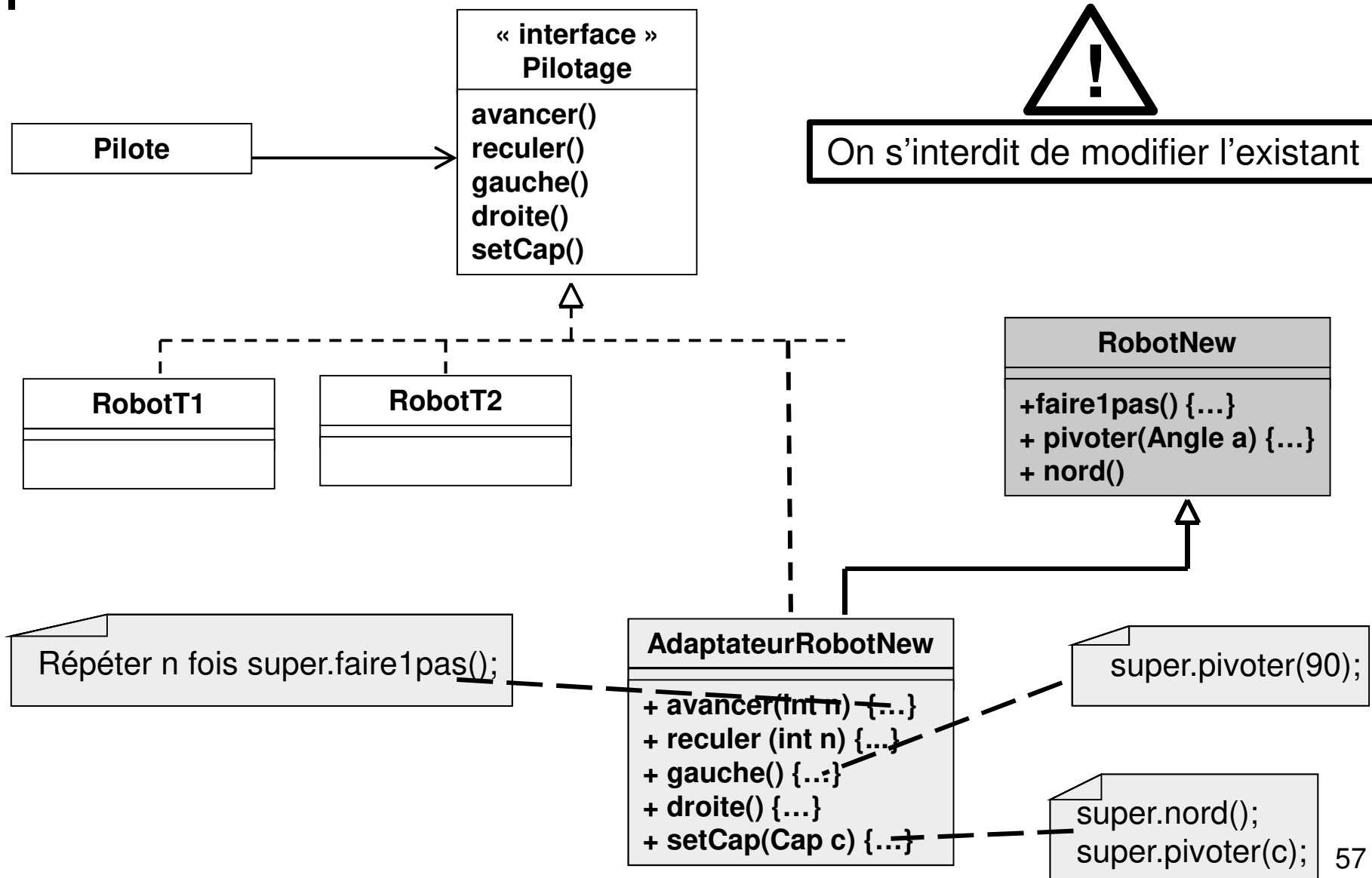
## *Un autre problème...*



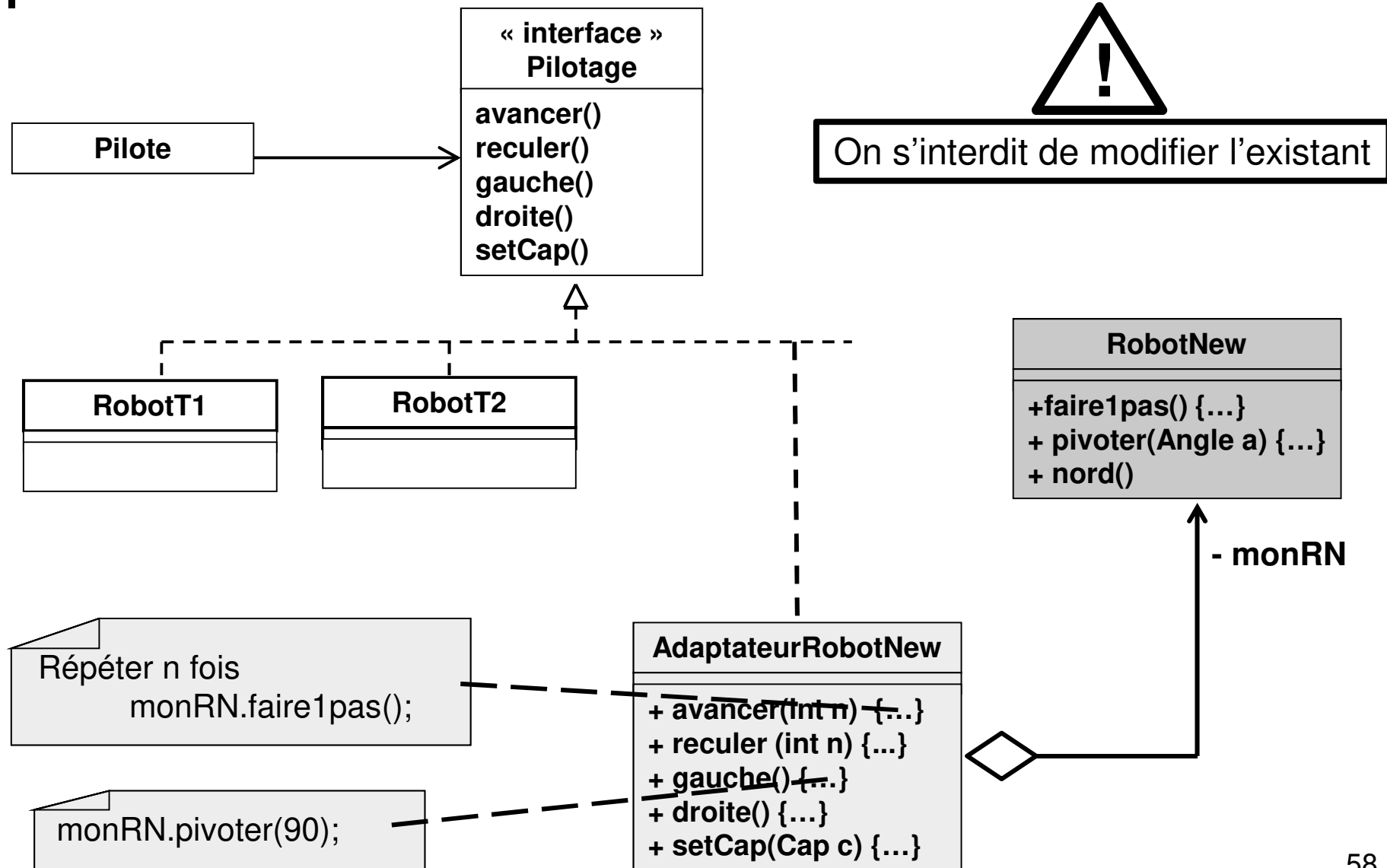
On s'interdit de modifier l'existant



# Un autre problème... (1<sup>re</sup> solution)



## Un autre problème... (2<sup>ème</sup> solution)



## *Le(s) modèle(s) « Adaptateur »*



- Adaptateur
- Intention
  - Convertit l'interface d'une classe en une autre conforme aux besoins d'un client (c'est de l'adaptation de l'existant)
  - Permet l'interaction entre objets, instances de classes dont les interfaces sont différentes
- Motivation
  - La réutilisation de classe d'une boîte à outils peut être empêchée parce que l'interface ne correspond pas au besoin « métier » ou au système existant côté client (« ils ne parlent pas la même langue »)
  - On ne veut pas modifier le code des classes impliquées

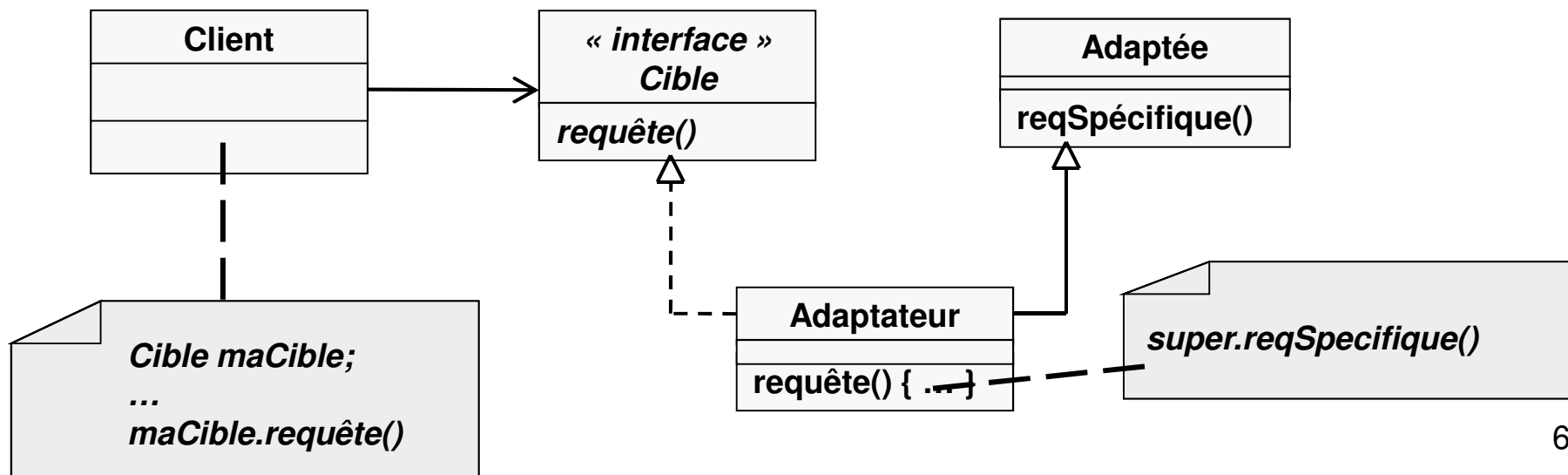


# Le(s) modèle(s) « *Adaptateur* »

- Participants
  - *Cible* : interface spécifique au domaine du client
  - *Client* : classe qui collabore avec une autre qui implémente *Cible*
  - *Adaptée* : classe ou interface existante réutilisée qui diffère de *Cible* et doit être adaptée
  - *Adaptateur* : classe qui adapte l'interface de l'adapté au client, donc implante l'interface *Cible* et
    - Soit étend la classe *Adaptée* (cas d'un adaptateur « de classe »)
    - Soit est associée à la classe *Adaptée* (cas d'un adaptateur « d'objet »)
      - Délégation
- Collaboration
  - Les clients invoquent les opérations d'une instance d'*Adaptateur* qui appelle les opérations de la classe *Adaptée*

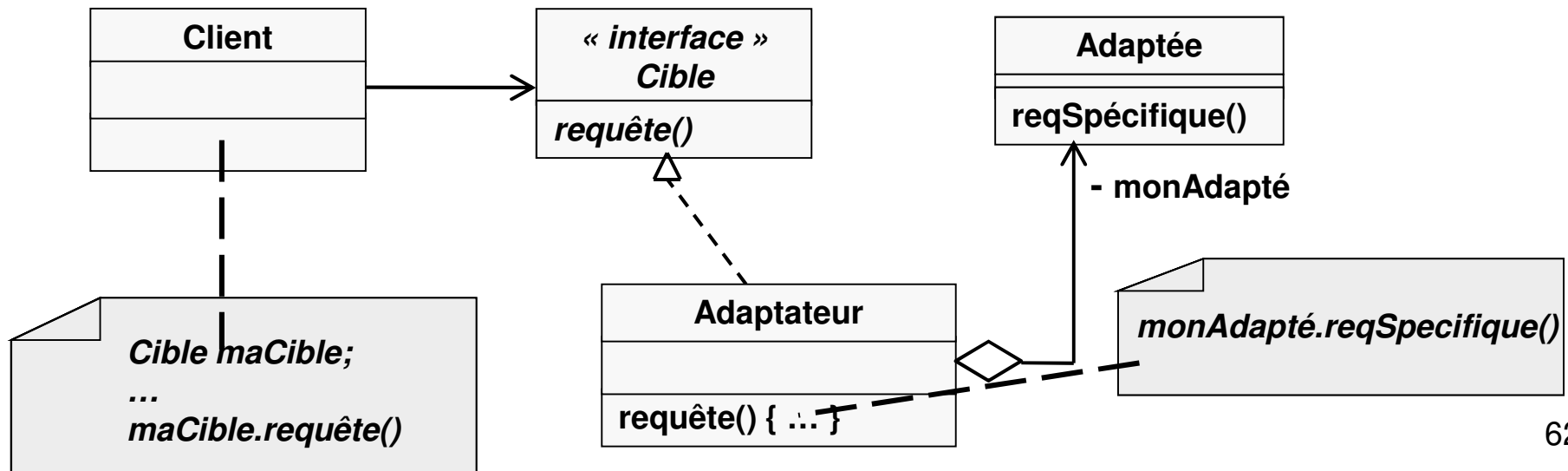
# *Le modèle « Adaptateur » de niveau classe*

- Collaboration (suite)
  - Le client invoque directement l'objet adapté via une méthode ajoutée pour la conformité des interfaces
    - L'interface fournie par l'adaptateur étend l'interface fournie par l'adapté avec l'interface requise par le client
- Structure



# Le modèle « Adaptateur » de niveau objet

- Collaboration (suite)
  - Un objet adaptateur sert d'intermédiaire entre l'objet client et l'objet adapté
    - Son interface est celle requise par le client
    - Il traduit la requête client en une requête pour l'adapté, invoque l'adapté, récupère le résultat, le transforme éventuellement et le retransmet au client
  - Les objets clients et adaptés s'ignorent mutuellement
- Structure





## *Le(s) modèle(s) « Adaptateur »*

- Conséquences

- Le rôle de l'adaptateur peut être plus ou moins complexe selon l'écart entre le client et l'adapté
- Un adaptateur de niveau « classe »
  - Peut redéfinir certaines méthodes de la classe « adaptée »
  - N'introduit pas d'objet intermédiaire (donc plus efficace en temps d'exéc.)
- Un adaptateur de niveau « objet »
  - Peut gérer plusieurs objets adaptés
  - Peut adapter des objets issus de sous-classe de *Adaptée*
  - Peut intercepter les requêtes ou les exceptions et les traiter lui-même...
  - Introduit un objet intermédiaire donc une indirection
  - Fonctionne en Java si Cible est une classe
    - On remplace le lien d'implantation par un lien d'héritage (pour l'adaptateur de niveau « classe » : problème d'héritage multiple)

- Modèles apparentés

- Pont, Décorateur, Procuration (Proxy), Façade