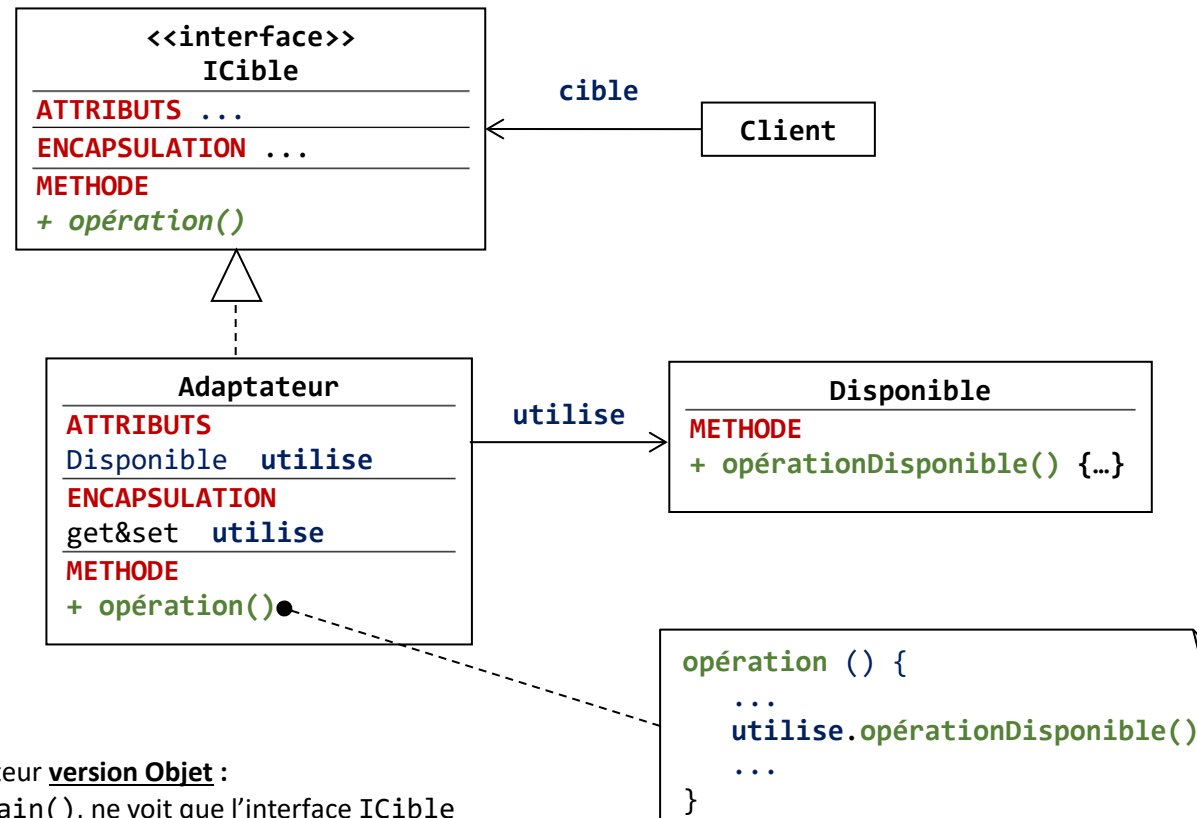


TD-TP : Le Design pattern « Adaptateur »

Le patron de conception Adaptateur permet de convertir l'interface d'une classe Disponible en une autre interface que le Client attend.

On peut dire que cela permet à un Client d'utiliser des ressources existantes alors qu'il est lié à une autre interface.



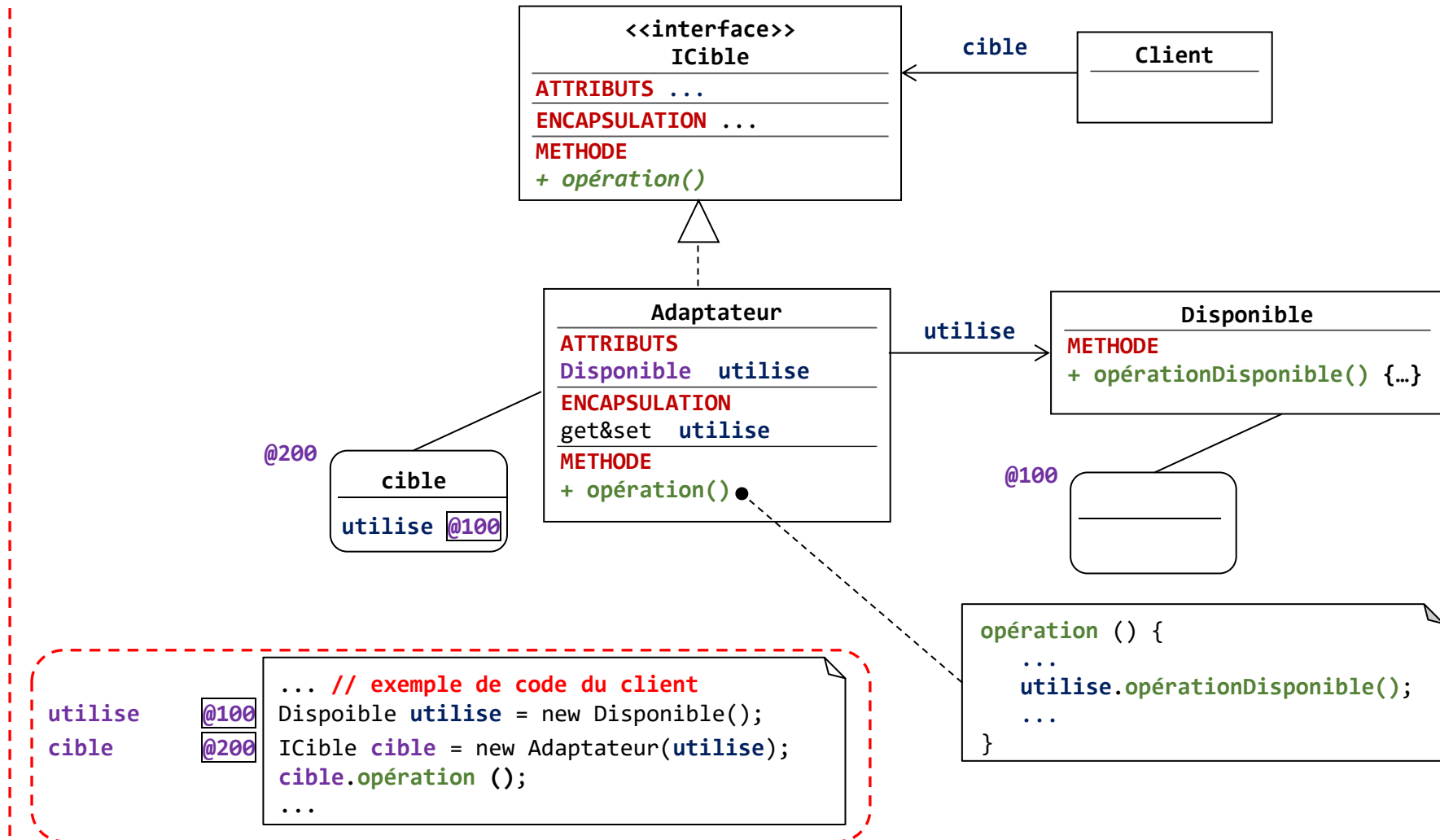
Principe du design Pattern Adaptateur **version Objet** :

- Le Client, par exemple un `main()`, ne voit que l'interface `ICible`
- L'Adaptateur implémente l'interface `ICible`
- L'Adaptateur est lié une classe `Disponible`
- Toute `opération()` de l'interface `ICible` est réalisée en sollicitant les méthodes d'une (ou plusieurs) classe `Disponible`

Design pattern Adaptateur – version Objet

Travail à faire

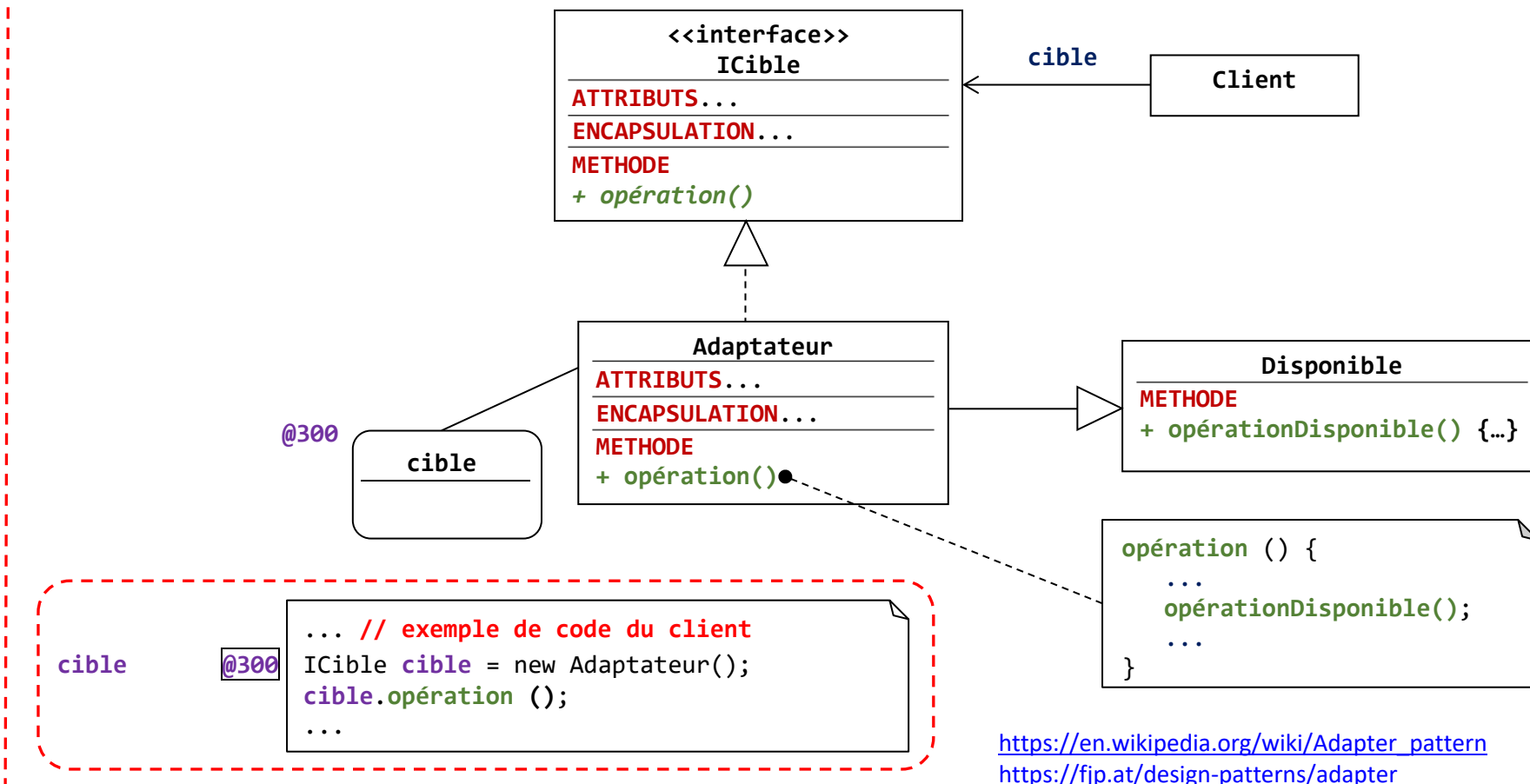
1. Quel serait le code d'un client qui mettrait en œuvre le patron de conception Adaptateur tel qu'il est schématisé dans le schéma UML.
2. Enrichir le schéma UML pour qu'apparaissent les objets créés par votre code.



Le design Adaptateur version Objet ci-dessus est similaire à sa version Classe que nous allons maintenant étudier. La différence réside dans la relation qui lie la classe Adaptateur et la classe Disponible. Dans la version Objet la relation est une relation de composition, alors que dans la version Classe la classe Adaptateur hérite de la classe Disponible.

Travail à faire

- Donner la représentation UML de la version Classe du design pattern Adaptateur décrite ci-dessus
- Donner le code d'un client qui mettrait en œuvre le patron de conception Adaptateur version Classe.
- Enrichir le schéma UML de la version Classe pour qu'apparaissent les objets créés par votre code.



https://en.wikipedia.org/wiki/Adapter_pattern
<https://fjp.at/design-patterns/adapter>