

Curso: Sistemas de Informação	
Disciplina: Padrões de projeto	Período: 8
Professor: Luiz Gustavo Dias	Tipo: Design Patterns
Objetivo: Adapter	

DESIGN PATTERNS: ADAPTER

Converter a interface de uma classe em outra interface, esperada pelos clientes. O Adapter permite que classes com interfaces incompatíveis trabalhem em conjunto – o que, de outra forma, seria impossível, por causa de suas interfaces incompatíveis.

Este padrão é muito utilizado quando se tem algum código de terceiros, ou código legado, na aplicação e deseja-se que não haja dependência de tal código.

Sobre o padrão

- É um padrão da categoria estrutural;
- Procede com a adaptação de códigos na aplicação, tal qual qualquer adaptador da vida real para o seu devido propósito;
- É muito usado para definir limites dentro de camadas de aplicação ou da aplicação para fora, como um framework de terceiro, por exemplo;
- Também pode ser usado para adaptar interfaces de código legado para um novo código.

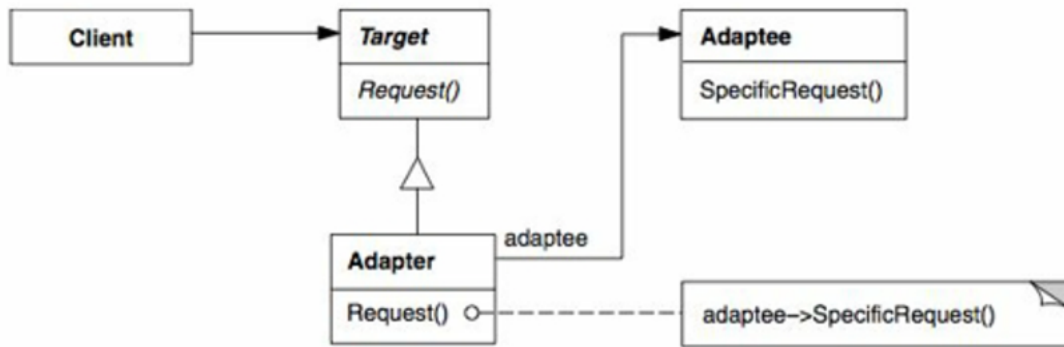
Aplicabilidade

Utilize o padrão de projeto Adapter quando:

- Você não quiser que o seu código dependa diretamente de código de terceiros ou legado;
- Você quiser usar uma classe existente, mas sua interface não corresponder à interface de que necessita;
- Você quiser criar uma classe reutilizável que coopere com classes não-relacionadas ou não-previstas, ou seja, classes que não necessariamente tenham interfaces compatíveis;

- Você precisar usar várias subclasses existentes, porém, for impraticável adaptar essas interfaces criando subclasses para cada uma. Um adaptador de objetos pode adaptar a interface da classe-mãe.

Estrutura



Target: interface que o código cliente precisa. Ela define a interface específica do domínio que Client utiliza.

Adaptee: o código que precisa ser adaptado. Ele adapta a interface do Adaptee à interface de Target.

Adapter: o adaptador em si. Ele colabora com objetos compatíveis com a interface Target.

Client: código cliente qualquer. Ele manipula objetos através de Target.

Consequências

Positivas:

1. Desacopla o código da aplicação de códigos de terceiros;
2. Aplica o SRP ao separar a conversão de interfaces da lógica da aplicação;
3. Aplica o OCP ao permitir introduzir novos adaptadores para o código existente.

Negativas:

1. Aumenta a complexidade da aplicação.