

Frameworks

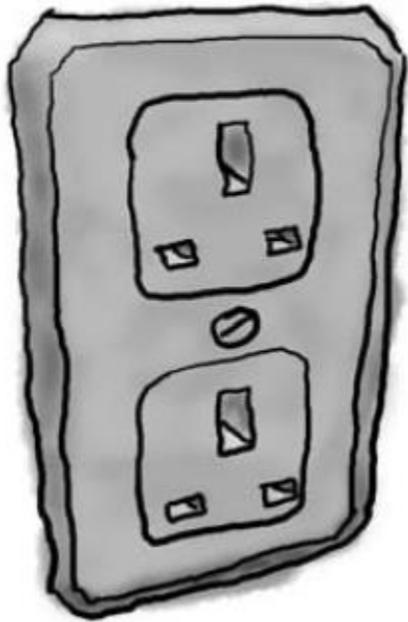
Padrão de projeto: Adapter

Padrão de projeto: Adapter

- Classificação: estrutural
- 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.

Adaptadores

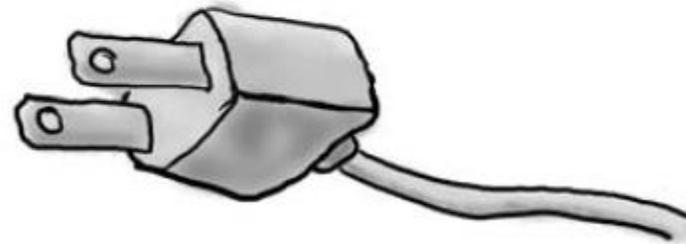
European Wall Outlet



AC Power Adapter



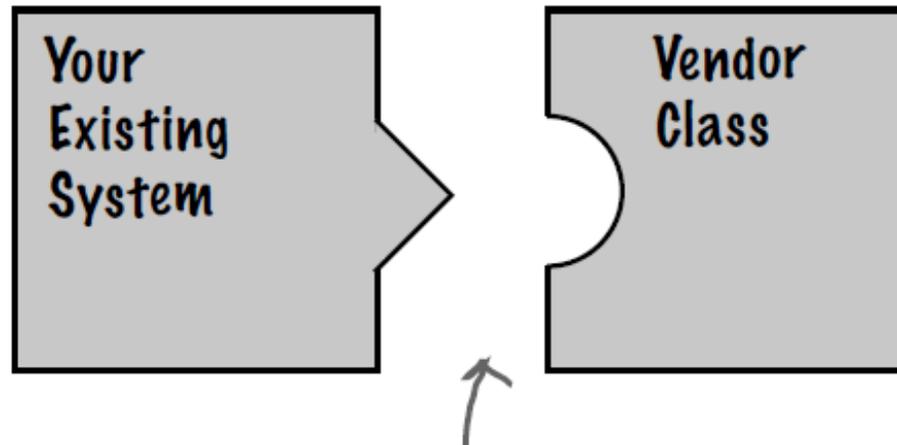
Standard AC Plug



The US laptop expects another interface.

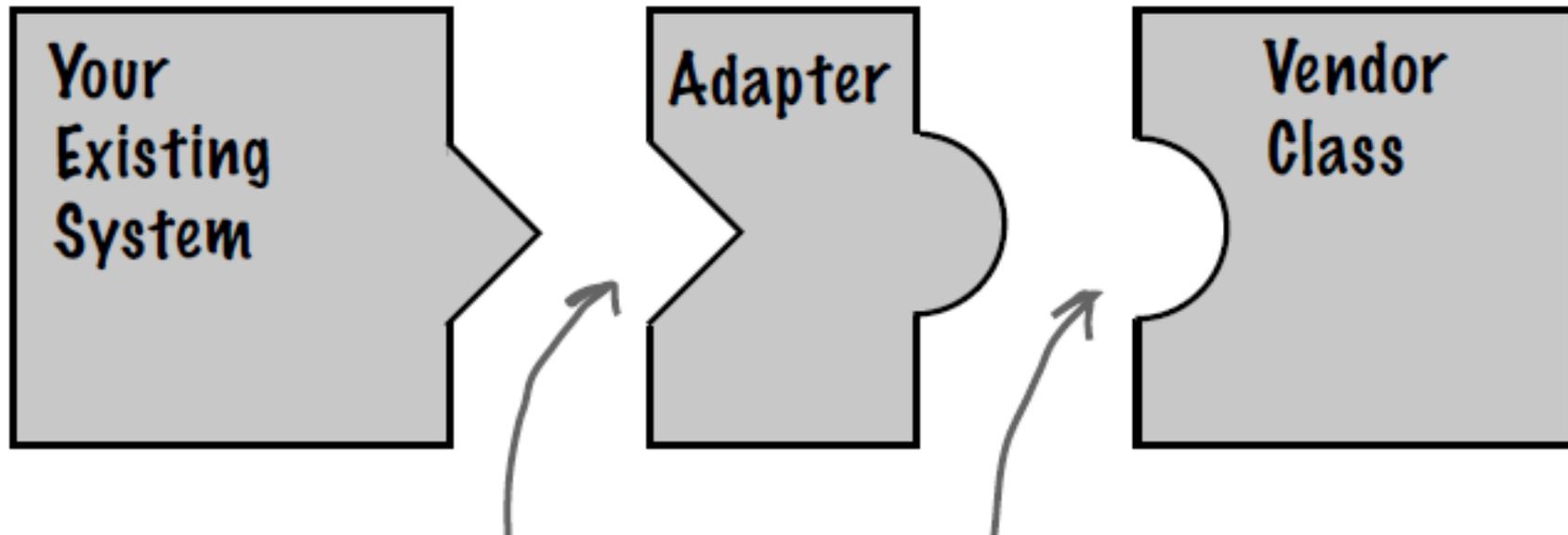
Cenário

- No mundo OO os adaptors desempenham o mesmo papel. Eles recebem uma interface e adaptam ela para o formato esperado pelo cliente.
- Exemplo: um sistema usa uma biblioteca, desenvolvida por terceiros. O sistema é construído para utilizar a interface desta biblioteca. O que aconteceria se essa biblioteca for substituída por outra?



Cenário

- Poderíamos modificar o código da aplicação de modo que ao invés de usar diretamente a interface da biblioteca, o sistema chame a interface de um adapter.
- Assim, escrevemos apenas o código de uma classe que adapta a interface de qualquer biblioteca para a nossa aplicação.



Exercício

- Proponha um conjunto de classes que represente o cenário: Sua empresa está desenvolvendo uma nova IDE chamada SuperBeans que oferece recurso de trabalhar com plug-ins.
- Para poder anexar um plugin na IDE, todo plugin deve implementar uma interface que define os seguintes métodos:
 - `int obterVersao()`
 - `void atualizar()`
 - `void instalar()`
 - `void remover(boolean)`
- Implemente o `pluginSQL` que utiliza essa interface.
- Uma parceria da sua empresa com a Oracle, exige que o sistema seja adaptável o suficiente para trabalhar com plug-ins do Eclipse. A interface dos plug-ins do Eclipse segue o seguinte contrato:
 - `string getVersion()`
 - `boolean update()`
 - `boolean install()`
 - `boolean remove(String feedback)`
- Neste momento você precisa adaptar apenas o plugin de modelagem. Crie um adapter que permite instalar plug-ins do tipo Eclipse na plataforma SuperBeans.