**Padrões de Projetos - Adapter VS Bridge**

1. **Definição**

Em Engenharia de Software é uma solução geral para um problema que ocorre com freqüência dentro de um determinado contexto no projeto de software.

* 1. **Padrões de Projetos**

"*O padrão é uma descrição do problema e essência de sua solução, onde pode ser reutilizada em diversos casos. O padrão não é uma especificação detalhada, pode-se pensar com uma descrição de conhecimento e experiência acumulados.*"(SOMMERVILLE, 2013).

Os padrões de projetos surgiram para solucionar problemas recorrentes no desenvolvimento de software sendo a solução comum.

* 1. **Características dos padrões de projeto**

Um padrão deve ter as seguintes características:

* + 1. **Nome** do padrão: permite definir o vocabulário a ser utilizado pelos projetistas;
    2. **Problema** a ser resolvido: relato de maneira clara de qual(is) problema(s) deve(m) ser(em) aplicado(s);
    3. **Solução** dada pelo padrão: gabarito para a resolução do problema.
  1. **Classificação dos padrões**

De acordo com o GoF (Gang of Four) são classificados em:

* + 1. **Padrões de Criação:**  auxiliam na criação dos objetos, tornando-os independentes de como são criados, compostos e representados;
    2. **Padrões Estruturais:** auxiliam na forma como as classes e objetos são compostos para formar estruturas maiores;
    3. **Padrões comportamentais:** auxiliam na forma como algoritmos e atribuições de responsabilidades entre os objetos e na sua comunicação.

1. **Adapter**

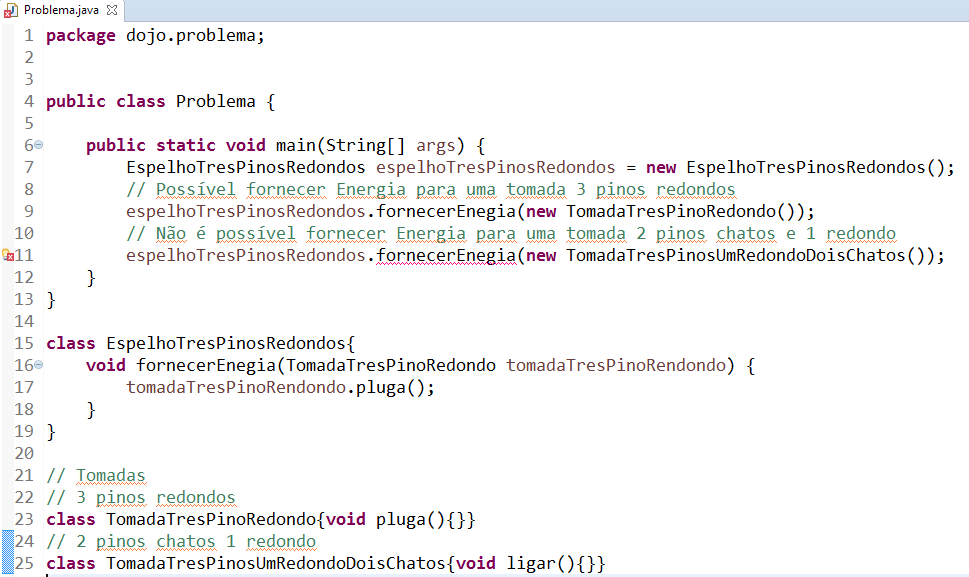
Padrão estrutural que converte a interface de uma classe em outra interface que os clientes esperam.

* 1. **Problema**

Classes que não poderiam trabalhar juntas devido a interfaces incompatíveis, teremos o auxílio do Adapter.



**Figura 01** - Problema das tomadas 3 pinos redondos para a tomada de 2 pinos chatos e 1 redondo.



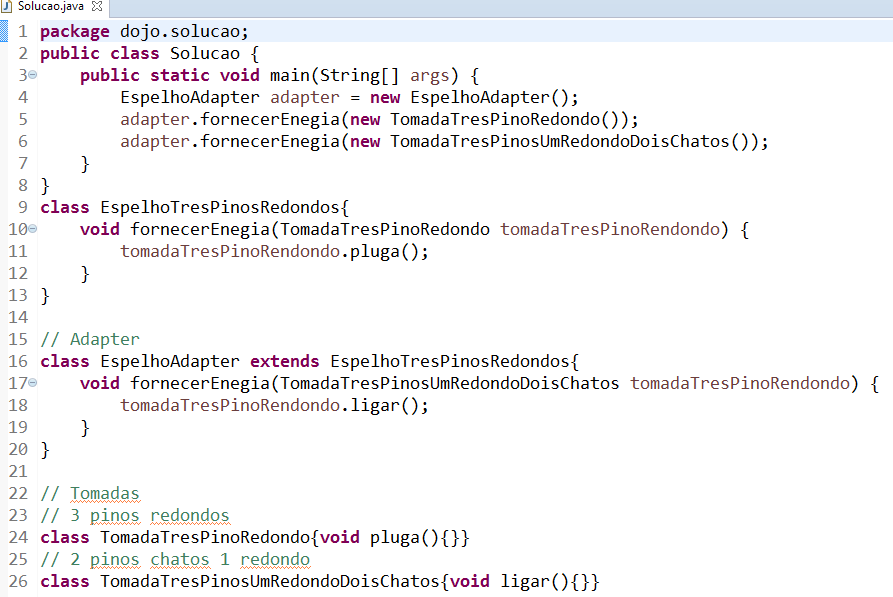
**Figura 02** - Exemplo da implementação do problema das tomadas 3 pinos redondos para a tomada 2 pinos chatos e 1 redondo.

* 1. **Solução**

Cria-se uma classe intermediária que servirá como interface entre as chamadas do código do cliente e o código alvo.



**Figura 03** - Adicionando um adaptador entre a tomada 3 pinos redondos e a tomada de 2 pinos chatos e 1 pino redondo.



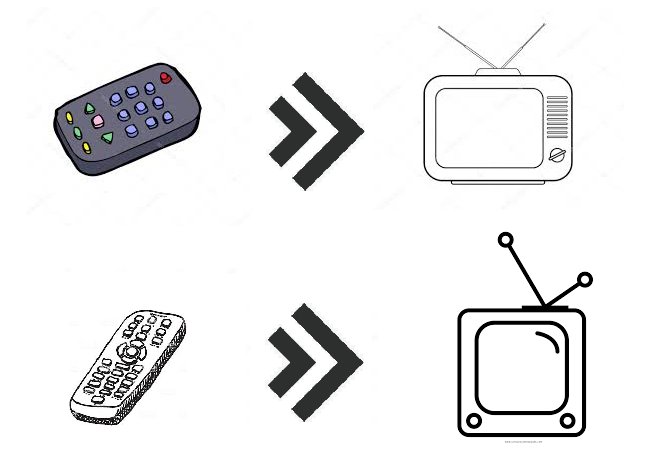
**Figura 04** - Exemplo da implementação da solução com a utilização da classe Adapter.

1. **Bridge**

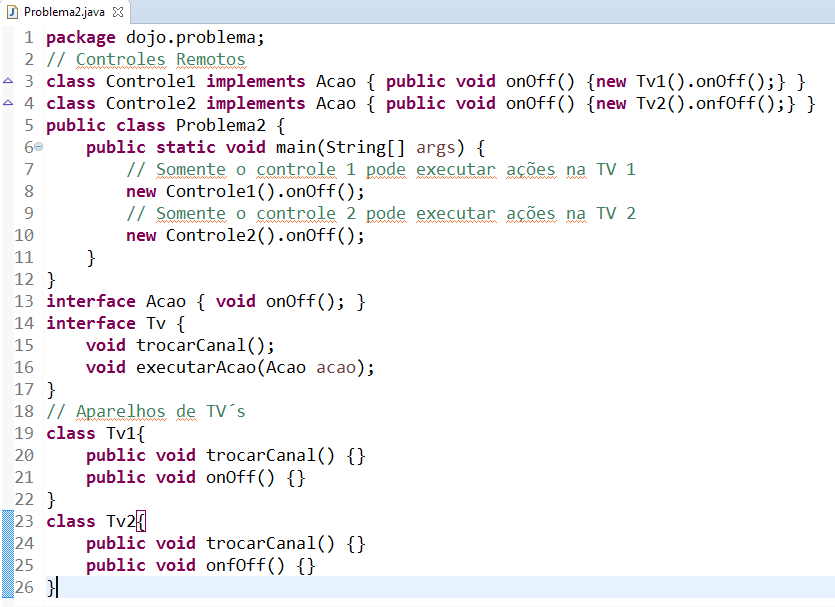
Padrão estrutural que desacopla uma abstração de sua implementação.

* 1. **Problema**

Como desacoplar uma abstração a partir da implementação, onde ambas possam variar.



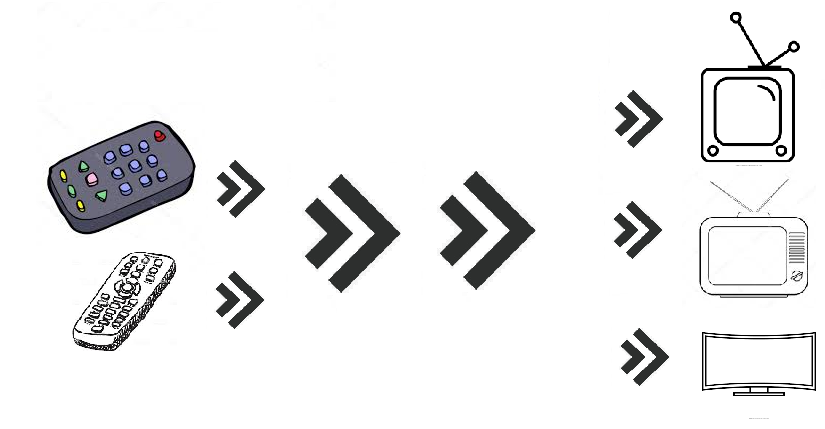
**Figura 05** - Apenas o controle remoto da mesma marca do aparelho de TV consegue 'conversar' com o mesmo.



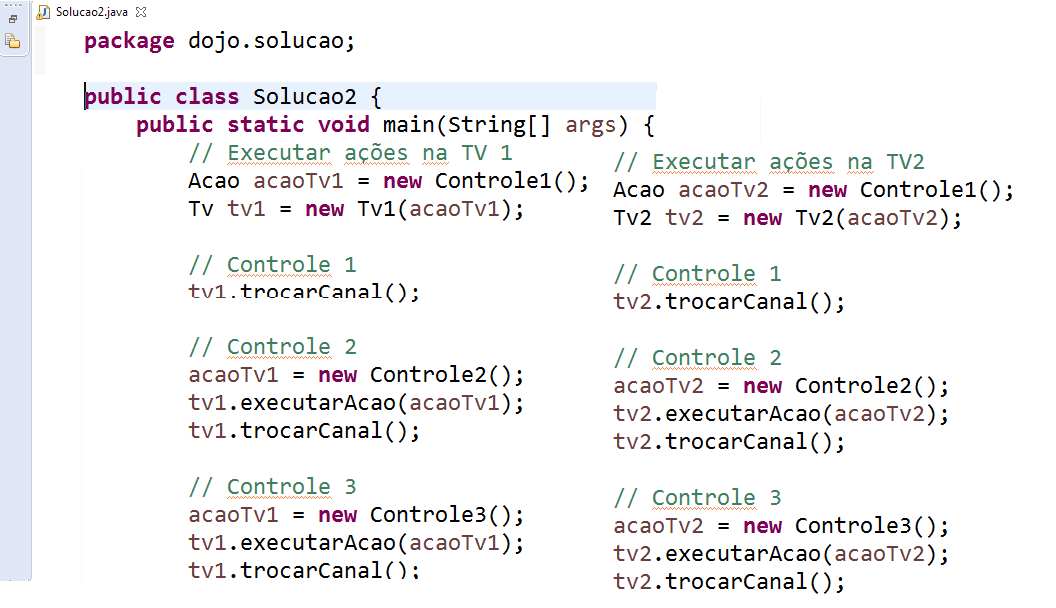
**Figura 06** - Exemplo da implementação do problema onde somente o controle da próprio aparelho consegue ligar-lo/desligá-lo.

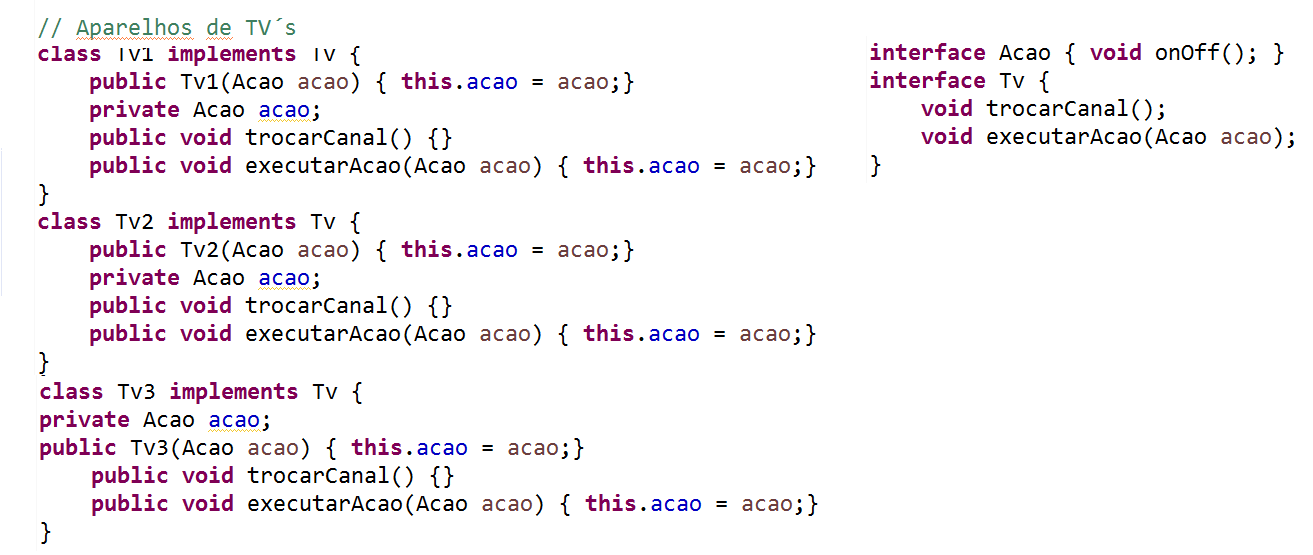
* 1. **Solução**

Remover os detalhes da implementação a partir da abstração.



**Figura 07** - Alterando a implementação para que seja possível qualquer controle remoto 'conversar' com qualquer aparelho de TV e vice-versa. Foi possível até mesmo adicionar um novo aparelho.





**Figura 08** - Exemplo da implementação da solução com a utilização da classe Bridge.

1. **Principais diferenças entre o Adapter e Bridge**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Adapter** | **X** | **Bridge** |
| **Classificação** | **Estrutural** |  | **Estrutural** |
| **Problema** | Classes que não poderiam trabalhar juntas devido a interfaces incompatíveis, teremos o auxílio do Adapter. |  | Como desacoplar uma abstração a partir da implementação, onde ambas possam variar. |
| **Solução** | Cria-se uma classe intermediária que servirá como interface entre as chamadas do código do cliente e o código alvo. |  | Remover os detalhes da implementação a partir da abstração. |

**Referências**

SOMMERVILLE, Ian. **Engenharia de Software**. São Paulo: Addison Wesley, 2003..

**Bridge Design Pattern**, 2018. Disponível em: https://www.geeksforgeeks.org/bridge-design-pattern. Acesso em 09/09/2018.

**Gang of Four**, 2013. Disponível em: http://wiki.c2.com/?GangOfFour. Acesso em 09/09/2018.

**Adapter** **Design Pattern**, 2018. Disponível em: https://sourcemaking.com/design\_patterns/adapter. Acesso em 09/09/2018.

**Bridge Design Pattern**, 2018. Disponível em: https://sourcemaking.com/design\_patterns/bridge. Acesso em 09/09/2018.