

## README

- PROJETO ORIGINAL (Projeto Sistema de Academia (1) / Projeto Movimentação de veículos (2))
  - Execução: Basta Salvar o projeto e executar normalmente na IDE. Projeto realizado no Netbeans.
- PROJETOS REFATORADOS
  - Execução: Basta Salvar os projetos e executar normalmente na IDE. Projeto realizado no Netbeans.

### -Projeto Sistema de Academia (1):

- Padrões aplicados: Command, Iterator;

#### -Command:

Motivação: Encapsulas uma solicitação para fazer algo em um objeto específico, ou seja, criar objetos de comando.

Solução: Retirou do código a presença de diversos switch cases, compactando as ações em blocos de ações, facilitando futuras manutenções e checagens no código.

#### -Iterator:

Motivação: Encapsular uma iteração, evitando acessos diretos no método, sem expor a sua implementação subjacente.

Solução: Resumiu o Código das interações realizadas no projeto, deixando visualmente mais didático e de fácil entendimento, assim como, evitou acessos frequentes aos locais das variáveis em cada iteração, esta quando antes feita no próprio método

- Classes e métodos afetados:

#### -Command:

Classes: Command, Invoker, Invoker\_four, Invoker\_one, Invoker\_three, Invoker\_two, SisGym, four\_Command, four\_Command\_next\_four, four\_Command\_next\_one, four\_Command\_next\_three, four\_Command\_next\_two, one\_Command, one\_Command\_next\_five, one\_Command\_next\_four, one\_Command\_next\_one, one\_Command\_next\_six, one\_Command\_next\_three, one\_Command\_next\_two, three\_Command, three\_Command\_next\_one, three\_Command\_next\_three, three\_Command\_next\_two, two\_Command,

two\_Command\_next\_four, two\_Command\_next\_one,  
two\_Command\_next\_three,  
two\_Command\_next\_two, Plano, Invoker\_Plano,  
Command\_Plano, Command\_Plano\_one,  
Command\_Plano\_two, Command\_Plano\_three,  
Command\_Plano\_four, Command\_Plano\_five,  
Command\_Plano2, Invoker\_Plano2,  
Command\_Plano2\_one, Command\_Plano2\_two,  
Command\_Plano2\_three, Command\_Plano2\_four,  
Command\_Plano2\_five.

Métodos: execute, invoke,

-Iterator:

Classes: Academia, Iterator, Iterator\_ArrayList\_client,  
Iterator\_ArrayList\_equip, Iterator\_ArrayList\_func,  
four\_Command\_next\_three,  
four\_Command\_next\_two,  
three\_Command\_next\_three,  
two\_Command\_next\_three,  
two\_Command\_next\_two.

Métodos: removerEquipamento, removerFuncionario,  
setReceberCliente, setDispesaFuncionario,  
pagarFuncionario, clientePagar, getclientesDevendo,  
first, isDone, next, hasNext, execute.

- Funcionalidades afetadas:

Apenas foram inseridos tratamentos de entrada com try e catch, o resto está idêntico.

## **-Projeto Movimentação de veículos (2):**

- Padrões aplicados: Strategy;

-Strategy:

Motivação: Clarificar o código ao diminuir ou remover lógica condicional, simplificando e compactando informações referentes a um objeto.

Solução: Retirou vários switch cases que dificultariam futuras manutenções no código, facilitando a organização e visualização de alguns problemas.

- Classes e Métodos afetados:

-Strategy:

Classes: Bicicleta, Caminhao, Carro, Moto, Contador, Mundo, NewEnum, Strategy\_Interface, Veiculo.

Métodos: Bicicleta, Caminhao, Carro, Moto, contar, incrementaVeiculo, Mundo, destruir, Incrementa, adiciona, Veiculo.

- Funcionalidades afetadas:  
Nenhuma funcionalidade foi afetada, foi apenas implementado o strategy, o terceiro padrão que faltava ser implementado entre os projetos.