

Lab12: *Facade*, Delegação e *State*

INSTRUÇÕES INICIAIS:

Nota: Os exercícios devem ser executados na ordem apresentada, pois o nível de dificuldade é crescente.

1. Obtenha o arquivo **lab12.zip** (disponível no ensino aberto);
2. Descompacte esse arquivo no seu diretório de trabalho (workspace do Eclipse);
3. Crie um novo projeto Java no Eclipse. Clicar em File > New > Java Project, manter o lugar padrão e escrever como nome do projeto exatamente o nome da pasta descompactada.
4. Ao final do lab, compacte a **pasta raiz do projeto**, em um arquivo com extensão **.zip**, e suba no ensino aberto no seu portfólio*, disponibilizando-o para os formadores. Suba apenas um arquivo. **Outros formatos não serão avaliados, implicando em nota 0 (zero) na atividade.**
5. Suba um arquivo de texto com a resposta para todas as questões conceituais **dentro** da pasta compactada. Serão aceitos os formatos **.txt** e **pdf**. **Coloque seu nome e RA neste arquivo.**

***obs:** criar um item no portfólio, com nome *Lab12*, com **somente** o arquivo *lab12.zip*. Não criem pastas no portfólio.

As questões de 1 a 3 devem ser feitas em sala e entregues até às 19h. As questões 4 e 5 podem ser entregues até às 23h59 de quinta-feira. Para isso, **edite** o item atual e substitua o zip com a **atividade completa**.

Segue abaixo alguns links para os padrões de projeto:

<http://www.vincehuston.org/dp/state.html>
<http://www.vincehuston.org/dp/facade.html>

Pacote *br.unicamp.ic.mc302.motor*:

1. Abra os arquivos *Principal.java*, *Bateria.java*, *VelaDeIgnicao.java* e *Injetor.java* e veja o comportamento deles. Complete o método *main* da classe *Principal* de forma a criar um motor e colocá-lo em funcionamento, ou seja, cumprir o seguinte comportamento:
 1. Deve-se criar um objeto de cada classe componente de motor;
 2. Deve-se verificar se a bateria está com carga;
 3. Se tem carga, coloca-se o motor em funcionamento - enquanto existir combustível no injetor:
 1. Injeta combustível
 2. Aciona a ignição

Compile e verifique o funcionamento do programa.
 2. O funcionamento do motor é complexo, e deveria ser acessado pelas classes usuárias através de uma interface simplificada. Implemente o padrão de projeto ***Facade***, criando uma classe, *Motor.java*, que encapsule o funcionamento do motor, e que tenha como meio de acesso um único método, *ligar()*. Explique as alterações obtidas com a implementação do padrão ***Facade***.
-

Pacote *br.unicamp.ic.mc302.veiculos*:

3. Complete a classe *Veiculo.java* de forma que ela utilize a interface construída anteriormente.
4. Considere agora que a classe *Veiculo* representa um veículo anfíbio, logo com dois estados possíveis: marítimo e terrestre. Implemente o padrão ***State*** para modelar esse caso. Será necessário criar as classes *EstadoVeiculo* (abstrata), *VeiculoMaritimo* e *VeiculoTerrestre*, além de criar os métodos *andar* e *mudarEstado* em *Veiculo*.
5. Implemente uma classe *Principal*, que possui um método *main* responsável por criar o veículo, fazê-lo andar, modificar o estado e fazê-lo andar novamente. (O método *andar* de cada estado deve imprimir na tela o estado correspondente).