



## Lista – Aula 06 - Padrões de projeto

1. Implemente uma classe que representa dois vetores matemáticos em  $R^3$  e, usando o padrão Bridge, defina 3 operações para objetos desta classe: adição, produto escalar, e produto vetorial. Exemplifique no método main.
2. Escreva uma classe derivada de `FilterInputStream` que seja capaz de decorar o funcionamento de um `BufferedReader` de maneira que quando o método `readLine` for evocado, a linha retornada mostre um contador de quantos caracteres já foram lidos desde que o stream foi criado, além da linha que foi lida. Exemplifique.
3. Pesquise o padrão Flyweight, explique sua finalidade e funcionamento – coloque sua resposta em um comentário de sua classe onde o método main foi definido. Acesse o exemplo <http://www.oodeesign.com/flyweight-pattern-wargame-example-java-sourcecode.html>, implemente-o e exemplifique seu funcionamento.
4. Escreva o diagrama UML dos projetos vistos em aula: Bridge, ChainOfResponsibilityTextParser, Facade, FactoryMethod, MediatorBuyer, e Observer.
5. Verifique a implementação Java e implemente seu próprio padrão observador (sem se preocupar com questões de sincronização).

**Para entrega: código do projeto NetBeans em um arquivo zip → entregar via Tidia→Atividades**

### **(ENTREGAR) 6.**

**a)** Use o padrão Bridge para implementar uma classe `MeusVetores` que tem como propriedades dois conjuntos de inteiros positivos representados como arrays, e que recebe funcionalidades novas para serem executadas sobre estes dois conjuntos. Escreva 3 funcionalidades: união, intersecção, e diferença entre conjuntos. Após a execução, as funcionalidades deverão exibir os dados de cada um dos conjuntos e também o resultado do processamento. Exemplifique no método main.

**b)** Altere seu código para que as funcionalidades do item a) sejam criadas por intermédio de um Factory.

**(ENTREGAR) 7.** Implemente uma classe `Divida` e use o padrão Decorator para efetuar os seguintes cálculos sobre a dívida representada por objetos desta classe: juros, juros com desconto, e juros com desconto com acréscimo de taxa. Note que a assinatura do Decorator irá requerer, além do objeto sendo decorado, um parâmetro cujo significado muda de decorador para decorador – defina um único parâmetro na classe mãe dos decoradores.



**(ENTREGAR) 8.** Considere um Sistema (classe) que é vulnerável toda vez que um de seus dois métodos (pense em métodos fictícios) recebe como parâmetro uma string “789”. Escreva a classe Sistema.

**a)** Use o padrão proxy para escrever uma classe que filtra chamadas de métodos cujos parâmetros coloquem em risco o sistema – veja o exemplo de ProxySeguranca no Tidia.

**b)** Use o padrão decorator para escrever uma classe FilteredSistema que, caso a string “789” seja detectada, ela seja convertida para “987”.

**(ENTREGAR) 9.** Use o padrão chain of responsibility para processar números, sendo que as classes da cadeia deverão dividir o processamento da seguinte maneira: classe 1, números múltiplos de 2; classe 2, números múltiplos de 3; classe 3, demais números. Implemente o cálculo do quadrado de um número dado, e o cálculo da raiz quadrada de um número dado. Exemplifique no método main.

**(ENTREGAR) 10.** Use o padrão Observer para escrever uma classe tal que toda vez que ela recebe uma nova string, observadores distintos realizam o seguinte processamento: transformar a string inteira em maiúscula, concatenar o ano atual ao fim da string, e concatenar o tamanho da string ao fim da própria string.