

## **Java - Padrões de Projetos Estruturais**

**Questão 1:** Sobre o padrão de projetos Adapter responda:

- Para que serve o padrão de projetos Adapter?
- Quando devemos usar o padrão de projetos Adapter
- Cite as vantagens e desvantagens do padrão de projetos Adapter
- Explique o código apresentado na Aula Prática - 6 Padrões Estruturais - Adapter
  - Explique a interface utilizada
  - Explique todas as classes implementadas
  - Qual a classe mais importante que define um Adapter? Porque?

**Questão 2:** Sobre o padrão de projetos Bridge responda:

- Para que serve o padrão de projetos Bridge?
- Quando devemos usar o padrão de projetos Bridge
- Cite as vantagens e desvantagens do padrão de projetos Bridge
- Explique o código apresentado na Aula Prática - 7 Padrões Estruturais - Bridge
  - Explique a interface utilizada
  - Explique a classe abstrata utilizada
  - Explique todas as classes implementadas
  - Qual a classe mais importante que define um Bridge? Porque?

**Questão 3:** Sobre o padrão de projetos Composite responda:

- Para que serve o padrão de projetos Composite?
- Quando devemos usar o padrão de projetos Composite
- Cite as vantagens e desvantagens do padrão de projetos Composite
- Explique o código apresentado na Aula Prática - 8 Padrões Estruturais - Composite
  - Explique todas as classes implementadas
  - Qual a classe mais importante que define um Composite? Porque?

**Questão 4:** Sobre o padrão de projetos Decorator responda:

- Para que serve o padrão de projetos Decorator?
- Quando devemos usar o padrão de projetos Decorator
- Cite as vantagens e desvantagens do padrão de projetos Decorator
- Explique o código apresentado na Aula Prática - 9 Padrões Estruturais - Decorator
  - Explique a interface utilizada
  - Explique a classe abstrata utilizada
  - Explique todas as classes implementadas
  - Qual a classe mais importante que define um Decorator? Porque?

**Questão 5:** Implemente os problemas abaixo utilizando seus conhecimentos sobre os padrões de projetos estruturais:

- Implemente uma solução em Java utilizando o padrão de projetos Composite para a situação abaixo:
  - Uma composição de Itens. Itens podem ser compostos de outros itens. Por exemplo: Um carro é composto de vários outros itens (portas, bancos, rodas, etc). Um Computador é composto de vários outros itens (CPU, HD, Monitor, etc).
  - A estrutura de um item deve conter pelo menos, os atributos: nome, descrição, preço
  - A composição deve permitir, incluir novos itens, remover, e trocar itens com outros itens
- Adapte o exemplo acima, para permitir que os itens criados também possam ser decorados através do padrão de projetos Decorator