

AULA PRÁTICA – SEMANA 09

- 1) Utilizando interfaces (Java), você pode especificar comportamentos semelhantes para classes possivelmente não relacionadas. Com base nessa ideia, siga as instruções abaixo:
 - a. Crie duas classes não relacionadas por herança: classe **Produto Eletrônico** e a classe **Serviço**. Dê a cada classe atributos e comportamentos únicos que não estejam relacionados com os atributos e métodos da outra classe. Sugestão:
 - i. **Produto Eletrônico** – atributo: nome do fabricante, peso, etc.
 - ii. **Serviço** – formato (remoto, em loja ou em domicílio), duração (em minutos).
 - b. Escreva uma interface **Loja** com os métodos vender e acionar garantia. Faça cada uma das classes **Produto Eletrônico** e **Serviço** implementar esta interface. Os métodos em cada classe devem simplesmente apresentar mensagens informativas na tela. Ex.: “Vendendo um serviço!”.
 - c. Escreva um aplicativo que crie um objeto de cada uma das duas classes. Crie um objeto **ArrayList<Loja>** e insira as referências dos objetos instanciados nesta coleção. Finalmente, itere pela coleção, chamando polimorficamente o método vender e acionar garantia de cada objeto.
- 2) Modifique o código do exercício 1, tornando **Produto Eletrônico** uma classe abstrata, e implementando duas novas subclasses concretas **Televisao** e **Celular**.
 - a. O aplicativo que cria a coleção de objetos vai continuar funcionando após a modificação na estrutura das classes?
 - b. Modifique o aplicativo para que passe a instanciar diretamente objetos **Televisao** e **Celular**, incluindo-os na coleção.
- 3) Modifique o código do exercício 2, apagando os métodos vender e acionar garantia da classe **Produto Eletrônico**.
 - a. O aplicativo vai continuar funcionando após a remoção dos métodos?
 - b. Modifique o aplicativo para que os métodos vender e acionar garantia sejam implementados nas classes **Televisao** e **Celular**.
 - c. Crie os métodos ligar e desligar na classe **Produto Eletrônico**. Cada método deverá exibir a mensagem de acordo com o objeto

passado como parâmetro. Por exemplo: se o objeto for uma televisão, o método ligar deverá exibir a mensagem “Ligando a televisão”.

- d. No aplicativo, instancie um objeto para cada uma das classes **Televisao**, **Celular** e **Serviço**. Crie uma coleção **LinkedList<Loja>** e insira as referências dos objetos instanciados nesta coleção. Finalmente, itere pela coleção, chamando polimorficamente os métodos vender, acionar garantia, ligar e desligar, de acordo com cada objeto. Dica: use o **instanceof**.