UNIVERSIDADE FEDERAL DO PARANÁ – CENTRO POLITÉCNICO

Curso: Ciência da Computação e Informática Biomédica

Disciplina: CI1062 – Paradigmas de Programação

Professora: Rachel Reis

AULA PRÁTICA – SEMANA 08

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`** – atributos: nome do fabricante, peso, marca, etc.
        ii. **`Serviço`** – atributos: formato (remoto, em loja ou em domicílio), duração (em minutos), etc.
    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`**.

4) Dado os padrões de projeto apresentados em aula (factory method, adapter e observer), identifique qual deles deve ser utilizado nas situações abaixo e, em seguida, crie a **estrutura** do padrão para cada situação com métodos e atributos apropriados (sempre que necessário).

a. Desenvolva uma aplicação para rede sociais que sinalize quando uma nova postagem é exibida no *feed* dos seus usuários. O *feed* é o local onde os usuários interagem e se mantêm atualizados com o conteúdo compartilhado por seus amigos e seguidores.

b. Desenvolva uma aplicação para uma empresa de dispositivos eletrônicos personalizados. Essa empresa produz *smarthpones*, tablets e laptops, cada um com suas especificações. A empresa precisa de um mecanismo que facilite a produção desses dispositivos e garanta que as caraterísticas solicitadas pelos clientes sejam incorporadas.

c. Desenvolva uma aplicação de streaming de música que permita que o usuário tenha acesso a diferentes serviços de música, como Spotify, Apple Music e YouTube Music usando uma única interface. Um streaming de música consiste na transmissão contínua de música pela Internet, sem que seja necessário baixar a música localmente.