**RESUMO DA AULA 08 - CLASSES ABSTRATAS E INTERFACES**

ALUNO: Denilson José do Bom Jesus Silva de Lima

**1. Classes Abstratas**

Existem problemas na redefinição de métodos (uso de Sobrecarga ou Sobrescrita), como a obrigatoriedade de preservação da semântica do método original, a necessidade de preservação ou aumento da visibilidade e se torna inadequado para alguns casos do mundo real pelo fato de algumas características entre as classes serem comuns, porém com diferenças em relação a funcionalidades. Isso pode ser mitigado com o uso de Classes Abstratas.

Trata-se de um conceito que agrupa categorias que possuem apenas algumas características em comum e com isso resolve os problemas da redefinição de métodos. Melhora a estruturação do software, o reuso e a extensibilidade. Possui métodos, atributos, adicionalmente pode ter "métodos abstratos [que não tem o "corpo" definido - método será sempre escrito pelas classes filhas; indica que todas as classes filhas (concretas) devem reescrever esse método, ou não compilaram] e apesar de ter construtor não pode ser instanciada.

- Métodos Abstratos

Sem corpo definido e a implementação é deixada para as subclasses. Só é possível anotar um método abstrato em uma classe abstrata (Ex:)

public abstract class Conta {

public abstract void debitar (double valor);

}

Subclasses (se não forem abstratas) não abstratas têm obrigatoriedade de implementar todos os métodos abstratos da superclasse ou dará erro. (Ex:)

public void debitar(double valor) { ... }

**2. Interfaces**

Um caso especial de classes abstratas (não é classe), possuindo todos os métodos abstratos, públicos; não *static* e não podem conter atributos, exceto static final; e não possui construtor. Se uma classe (concreta) implementar uma interface, ele deve implementar todos os métodos definidos na interface, caso seja abstrata, tem

liberdade para escolher quais quer implementar ou não. No caso de uma classe Filha (concreta) esta deve implementar todos os métodos da interface que não foram implementados pela classe Pai. É útil para agrupar categorias por funcionalidade e define tipos de forma abstrata indicando a assinatura dos métodos. (Ex:)

public interface IConta {

public void creditar (double valor);

public abstract void debitar (double valor);

}

public class Conta implements IConta {

//pode implementar múltiplas interfaces

//por ser concreta, preciso implementar todos os métodos da interface

}

public abstract class Conta1 implements IConta { ... } //por ser abstrata, posso implementar os métodos da interface que quero.

Para instanciar uma interface é preciso instanciar qualquer classe que implemente essa interface ou que herde de alguma classe que implemente essa interface.

Interfaces são boas para a criação de constantes globais. Servem para publicar para publicar os métodos que um tipo deve prover e introduz conceitos de subtipo e

supertipo, sendo os métodos implementados pelos subtipos.

- Instanceof

Verificar se uma classe é subtipo de uma interface.