



# Classe - Herança

JCAVI Treinamentos - JAVA I

Prof. Marcelo da Silva



# Herança

Segundo Deitel, a herança é uma forma de reutilização de software na qual uma nova classe é criada, absorvendo membros de uma classe existente e aprimorada com capacidade novas ou modificadas.

A classe existente chamamos de Superclasse e a classe criada chamamos de subclasse. Uma subclasse pode tornar-se Superclasse para outras classes geradas a partir dela.

A subclasse exhibe os comportamentos da SuperClasse e pode adicionar outros comportamentos a si mesma.



# Herança

Em Herança estamos trabalhando primariamente com o relacionamento entre classes, definido como “É UM”, isto é uma subclasse é um objeto da SuperClasse também.

Um relacionamento de Herança cria uma estrutura hierarquica do tipo árvore.

Um dos problemas referentes a herança é que uma subclasse pode herdar métodos que ela não necessitaria ter, tornando-se necessário as vezes sobrescrever os métodos na sbuclasse.



## Modificadores de acesso: Private

Os membros private só são acessíveis dentro de sua própria classe e não são herdados por suas subclasses.



# Modificadores de acesso: Protected

Utilizar acesso *protected* oferece um nível intermediário entre o *public* e o *private*.

Os membros *protected* de uma superclasse podem ser acessíveis por suas subclasses e por membros de outras classes no mesmo pacote (membros protegidos também tem acesso de pacote).

Os métodos de subclasse podem referir-se a membros *public* e *protected* da super classe usando a palavra *super* precedendo o nome do membro.



# Classes e Construtores

Um construtor define como objetos de uma classe são inicializados.

Os construtores possuem modificadores de acesso public.

Cada classe deve ter pelo menos um construtor padrão, um construtor padrão não tem parametros.

.



# Membros de classe do tipo Static

Uma variável static representa informações de escopo de classe.

Com static todos os objetos gerados a partir de uma classe compartilham os mesmos dados.

Os membros da classe static existem mesmo quando nenhum objeto da classe existe.

Quando um membro da classe for private static, um método private static deve ser fornecido e o método deve ser chamado qualificando seu nome com o nome da classe e um ponto(.).



# Variáveis de instância final

Algumas variáveis precisam ser modificadas e outras não, aí que entra a utilização de *final*.

ex.: **private final int INCREMENT;**