

Aula 5 - Atividade EAD - Conceitos de Herança

Augusto Cesar de Aquino Ribas

27 de setembro de 2014

Trabalho para a disciplina de linguagem de programação orientada a objetos do professor Samuel Ferraz

1 Introdução

A herança reduz o tempo de desenvolvimento do programa.

A superclasse direta de uma subclasse(especificada pela palavra-chave `extends` na primeira linha de uma declaração de classe) é a superclasse a partir da qual a subclasse herda. A superclasse indireta de uma subclasse está dois ou mais níveis acima da hierarquia de classe dessa subclasse.

Em herança única, uma classe é derivada de uma superclasse direta. Na herança múltipla, uma classe é derivada de mais de uma superclasse direta. O java não suporta herança múltipla.

Uma subclasse é mais específica que sua superclasse e representa um grupo menor de objetos.

Cada objeto de uma subclasse também é um objeto da superclasse dessa classe. Entretanto, um objeto de superclasse não é um objeto de subclasse da sua classe.

Um relacionamento *é um* representa herança. Em um relacionamento *é um*, um objeto de uma subclasse também pode ser tratado como um objeto de sua superclasse.

Um relacionamento *tem um* representa a composição. Em um relacionamento *tem um*, um objeto de classe contém referências a objetos de outras classes.

2 Superclasses e subclasses

Os relacionamentos de herança simples formam estruturas hierárquicas do tipo árvore - há uma superclasse em um relacionamento hierárquico com suas subclasses.

3 Membros `protected`

Os membros public de uma superclasse são acessíveis onde quer que o programa tenha uma referência a um objeto dessa superclasse ou para uma de suas subclasses.

Os membros de uma superclasse `private` só podem ser acessados diretamente a partir de dentro da declaração da superclasse.

Os membros `protected` de uma superclasse tem um nível intermediário de proteção entre acesso `public` e `private`. Eles podem ser acessados por membros da superclasse, por membros de suas subclasses e por membros de outras classes no mesmo pacote.

Os membros `private` de uma superclasse permanecem ocultos nas suas subclasses e só podem ser acessados por meio dos métodos `public` ou `protected` herdados da superclasse.

Quando um método de subclasse sobrescrever um método de superclasse, o método de superclasse pode ser acessado a partir da subclasse se o nome de método de superclasse for precedido por `super` e um ponto separados (`.`).

4 Relacionamento entre superclasses e subclasses

Uma subclasse não pode acessar os membros `private` de sua superclasse - permitir isso violaria o encapsulamento da superclasse. Uma subclasse pode, porém, acessar os membros não `private` de sua superclasse.

Uma subclasse pode invocar um construtor da sua superclasse utilizando a palavra-chave `super`, seguida pelo conjunto de parênteses contendo os argumentos do construtor de superclasse. Isso deve aparecer como a primeira instrução no corpo do construtor da subclasse.

Um método de superclasse pode ser sobrescrito em uma subclasse para declarar uma implementação apropriada para a subclasse.

A notação `@Override` indica que um método deve sobrescrever um método de superclasse. Quando o compilador encontrar um método declarado com `@Override`, ele comparará a assinatura do método com as assinaturas de método da superclasse. Se não houver uma correspondência exata, o compilador emite uma mensagem de erro, como `'method does not override or implement a method from a supertype'` (método não sobrescreve ou implementa um método a partir de um supertipo).

O método `toString` não recebe nenhum argumento e retorna uma `String`. O método `toString` da classe `Object` normalmente é sobrescrito por uma classe.

Quando um objeto é enviado para saída utilizando o especificador de formato `%s`, o método `toString` do objeto é chamado implicitamente para obter representação de `String`.

5 Construtores em subclasses

A primeira tarefa de qualquer construtor de subclasse é chamar o construtor de sua superclasse direta, explícita ou implicitamente, para assegurar que as variáveis de instância herdadas da superclasse são inicializadas.

6 Engenharia de software com herança

Declarar variáveis de instância `private`, ao fornecer métodos não `private` para manipular e realizar a validação, ajuda a impor uma boa engenharia de software.

7 Classe Object

Classe Object é a superclasse usada na linguagem Java, todos as classes são por padrão extensões dessa classe.

8 Referência

Paul Deitel e Harvey Deitel 2010 Java. Como Programar 8ª edição Pearson Prentice Hall 1143pp.