Linguagem de Programação II

Prof. Marc Antonio Vieira de Queiroz

Ciência da Computação - Sistemas de Informação - UNIFIL
marc.queiroz@unifil.br

23/05/2013

Roteiro da aula I

1. 8 Reutilização de classes

Sobrecarga de métodos I

Uma das características mais interessantes de linguagens de programação orientadas a objetos é a capacidade de facilitar a reutilização de código - o aproveitamento de classes e seus métodos que já estejam escritos e que já tenham o seu funcionamento testado e comprovado. Reutilização de código diminui a necessidade de escrever novos métodos e classes, economizando o trabalho do programador e diminuindo a possibilidade de erros.

Solução I

Existem dois mecanismos básicos de reuso de classes em Java: delegação (ou composição) e herança. Com delegação, usamos uma instância da classe base como campo na nova classe, e com herança criamos a classe nova como uma extensão direta da classe base.

8.2 Delegação ou Composição I

O primeiro mecanismo de reaproveitamento de classes em Java é conhecido como delegação ou composição. Podemos criar novas classes que estendem uma outra classe base se incluirmos uma instância da classe base como um dos campos da nova classe, que será então composta de campos específicos e de uma instância de uma classe base. Para que os métodos da classe base possam ser executados, escreveremos métodos correspondentes na classe nova que chamam os da classe base, desta forma delegando a execução dos métodos.

Exemplo listagem 8.1 I

Listagem 8.1: A classe DataHora, que reusa as classes Data e Hora através de delegação.

Outros exemplos I

Um outro exemplo de reaproveitamento de classes existentes usando o mecanismo de delegação pode ser visto na classe RegistroAcademicoDeGraduacao, mostrada na listagem 8.2. A classe DemoRegistroAcademicoDeGraduacao, mostrada na listagem 8.3, demonstra o uso de instâncias da classe RegistroAcademicoDeGraduacao.

Delegação e modificadores de acesso

Campos e métodos de classes podem ter modificadores de acesso que definem quais destes campos e métodos poderão ser acessados a partir de outras classes. Classes que contém instâncias de outras classes sofrem as restrições impostas pela classe contida da forma esperada, mais restrições impostas pela própria classe. Por exemplo, suponhamos que a classe Data tem seus campos públicos, e uma instância da classe Data é usada via delegação na classe Evento, e esta instância é declarada como private. De dentro da classe Evento, todos os campos da classe Data podem ser acessados e modificados, mas nenhuma classe que use a classe Evento poderá acessar diretamente estes campos.

Exemplos

Os efeitos das modificações de acesso com delegação são exemplificados pelas classes Pessoa0, Funcionario0 e DemoFuncionario0, mostradas a seguir.

Delegação e construtores

As regras para uso de construtores com classes que contém instâncias de outras classes é bem simples: se os construtores das classes usadas através da delegação deverem obrigatoriamente ser chamados, eles podem ser chamados em qualquer ponto da classe que os contém. Por exemplo, se for obrigatória a execução do construtor de uma instância da classe Data em uma classe Evento, este construtor pode ser chamado em qualquer método da classe Evento, mas idealmente deve ser chamado a partir do construtor da própria classe Evento.