Universidade Federal do Sul e Sudeste do Pará

Programação Orientada a Objetos

Prof. Dr. Marcela Alves

Tarefa 4 – Herança

Elson da Silva Sousa[1]

Questão 2) Reflita e Responda com suas palavras as seguintes perguntas:

1. O que é herança em Programação Orientada a Objetos (POO) e qual seu principal objetivo?

R: Herança é quando a classe pode aproveitar os atributos e métodos de outra. O objetivo é reaproveitar código e evitar repetições.

2. Qual é a diferença entre uma superclasse e uma subclasse? Dê um exemplo de cada.

R: A superclasse é a classe-mãe, mais genérica. A subclasse é a classe-filha, mais especifica.

Exemplo:

Superclasse: Animal.

Subclasse: Cachorro, Papagaio, gato...

3. Dada a seguinte hierarquia UML: Pessoa → Estudante, o que significa afirmar que "todo Estudante é uma Pessoa, mas nem toda Pessoa é um Estudante"?

R: Significa que a classe Estudante herda de Pessoa, então tem tudo o que uma Pessoa tem. Mas uma Pessoa pode ser algo diferente, como um Professor, por exemplo.

[1] Graduando em Engenharia da Computação pela Universidade Federal do Sul e Sudeste do Pará. Email: elson.sousa@unifesspa.edu.br

4. Quais são as vantagens do uso da herança no desenvolvimento de software orientado a objetos? Por que uma subclasse não consegue acessar diretamente atributos declarados como private na superclasse? Como isso pode ser resolvido?

R: As vantagens são a reutilização do código, organização no sistema e no desenvolvimento e a facilidade na manutenção e expansão. Além de que a subclasse não acessa atributos private da superclasse porque private é restrito a própria classe e isso só pode ser resolvido usando protected.

5. Qual o símbolo e a direção da seta usada para representar herança em diagramas de classes UML?

R: Uma seta com ponta triangular vazia, apontando da subclasse para a superclasse.

6. Para que serve a palavra-chave super em Java? Cite dois contextos diferentes em que ela pode ser usada.

Serve para acessar elementos da superclasse. São usados para chamar construtor da superclasse e acessar um método da superclasse.

7. Um sistema define as classes Professor, ProfHorista e ProfDE. Por que é mais vantajoso centralizar atributos comuns na classe Professor ao invés de declará-los separadamente nas subclasses?

R: Para evitar repetir código. Assim, no exemplo ProfHitoista e ProfAA herdam esses dados sem duplicação.

8. Suponha que você esteja modelando um sistema de transporte. Como você organizaria uma hierarquia de classes para representar Transporte, TransporteTerrestre, TransporteAereo, Carro, Avião e Helicóptero? Qual o papel da herança nessa modelagem?

R: Classe base: Transporte
Subclasses: TransporteTerrestre e TransporteAereo
Sub-subclasses: Carro herda de TransporteTerrestre; Avião e Helicóptero herdam de TransporteAereo.

A herança ajuda a organizar e reaproveitar o que for comum entre os tipos de transporte.

9. O que acontece se, em uma subclasse, você não chamar explicitamente o construtor da superclasse usando super()? Em que situação isso pode gerar erro?

O Java tenta chamar o construtor sem parâmetros da superclasse. Se ele não existir, dá erro.
Isso acontece quando a superclasse tem só construtor com parâmetros.