●Classes do tipo Interface

● Definição

Segundo Horstmann (2008), um tipo interface é semelhante a uma classe, mas há várias diferenças importantes: Todos os métodos em um tipo interface são abstratos, isto é, eles têm um nome, parâmetros e um tipo de retorno, mas não têm uma implementação; todos os métodos em um tipo interface são automaticamente públicos; um tipo interface não tem campos de instância.

Uma interface é formalmente uma classe abstrata, somente com atributos constantes (final) e estáticos (static) e métodos sem corpo. Estes deverão ser implementados pelas classes que irão implementar a interface. É importante observar que os atributos na interface precisam ser inicializados.(MENDES, 2009, pg. 268).

● Utilidade

“Às vezes é possível tornar o código mais geral e reutilizável focando nas operações essenciais que são executadas. Tipos interface são utilizados para expressar essas operações comuns.” (HORSTMANN, 2008).

Uma interface é uma coleção de declarações de métodos sem dados e sem corpo. Isso quer dizer que os métodos de uma interface são sempre vazios (ou seja, são simples assinaturas de métodos). Quando uma classe implementa uma interface, ela deve programar todos os métodos declarados nela. Dessa forma, a interface impõe que a classe que a implementa tenha métodos com assinaturas específicas. (GOODRICH; TAMASSIA, 2013, pg. 81).

● O que são métodos abstratos

“Todos os métodos em um tipo interface são abstratos; isto é, eles têm um nome, parâmetros e um tipo de retorno, mas não têm uma implementação.” (HORSTMANN, 2008).

Um método é composto de uma assinatura, e a sua implementação descreve seu corpo. Há situações em que é possível afirmar que uma classe deve ter um método com determinada assinatura, mas nada deverá ser definido em seu corpo. Para atender a essas necessidades devemos criar métodos classificados como abstratos. A classe que tenha pelo menos um método abstrato torna-se também uma classe abstrata automaticamente e requer que em sua definição seja usada a palavra reservada abstract. É importante observar que o corpo do método (seu comportamento) deverá ser implementado por qualquer classe que venha a herdar da classe abstrata. (MENDES, 2009, pg. 212).

Referências

GOODRICH, Michael T.; TAMASSIA, Roberto. **Estruturas de Dados & Algoritmos**. 5. ed. Porto Alegre: Bookman, 2013.

HORSTMANN, Cay. **Conceitos de computação com Java**. 5. ed. Porto Alegre: Bookman, 2008.

MENDES, Douglas Rocha. **Programação Java com ênfase em orientação a objetos**. São Paulo: Novatec, 2009.