

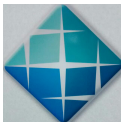
Programação II

Aula 02

Evandro J.R. Silva¹

¹Bacharelado em Ciência da Computação
Estácio Teresina

18/08/2022



Sumário

1 Métodos Genéricos

2 Classes Genéricas

3 Exercícios

4 FIM

Métodos Genéricos

- Imagine um método que não "dependa" de uma classe. Ex.: um método sort que ordena um array de Integers, ou String, ou qualquer tipo que possa ser comparado para ordenação.

Métodos Genéricos

- Imagine um método que não "dependa" de uma classe. Ex.: um método sort que ordena um array de Integers, ou String, ou qualquer tipo que possa ser comparado para ordenação.
- Métodos **sobrecarregados** são frequentemente utilizados para realizar operações *semelhantes* em tipos *diferentes* de dados.
- Exemplo no Eclipse.

Métodos Genéricos

- Imagine um método que não "dependa" de uma classe. Ex.: um método `sort` que ordena um array de `Integers`, ou `String`, ou qualquer tipo que possa ser comparado para ordenação.
- Perceba que temos métodos sobrecarregados, mas não genéricos!
- Então vejamos como fazer um genérico.

Métodos Genéricos

- Operadores relacionais como $>$ ou $<$ não funcionam em tipos genéricos.
- Solução: **interfaces**

Métodos Genéricos

- Operadores relacionais como $>$ ou $<$ não funcionam em tipos genéricos.
- Solução: **interfaces**
- Entendendo o código que vimos

Métodos Genéricos

- Operadores relacionais como `>` ou `<` não funcionam em tipos genéricos.
- Solução: **interfaces**
- Entendendo o código que vimos
 - É possível comparar dois objetos da mesma classe se ela implementar a **interface genérica** `Comparable<T>`.
 - Objetos que implementam essa interface possuem o método `compareTo`.
 - Por padrão, o método `compareTo` **deve** retornar `0` se os objetos forem iguais, um inteiro positivo se o primeiro objeto for maior que o segundo, e um inteiro negativo se o primeiro objeto for menor que o segundo.

Métodos Genéricos

- E os métodos genéricos podem ser sobrecarregados?
- Resposta:

Métodos Genéricos

- E os métodos genéricos podem ser sobrecarregados?
- Resposta: SIM!

Classes Genéricas

- A ideia de classes genéricas é a mesma de métodos genéricos, ou seja, classes que podem servir para construir conjuntos de objetos de diferentes tipos.
- Ex.: Uma estrutura de dados independente do tipo: fila de gente, fila de cachorro, pilha de papel, etc.
- Codiguin ...

Exercícios

- 1 Pesquisar sobre parâmetro **curinga**.
- 2 Escreva um programa que insira, classifique e gere uma saída de um array `Integer` e um array `Float`. Para isso, crie um método genérico sobrecarregado com outros métodos genéricos e não genéricos, de forma que o programa possa receber entre 2 e 5 parâmetros.
- 3 Escreva uma versão genérica simples do método `isEqualTo` que compara seus dois argumentos com o método `equals` e retorna `true` se forem iguais e `false` caso contrário. Utilize esse método genérico em um programa que chama `isEqualTo` com uma variedade de tipos predefinidos, como `Object` ou `Integer`. Qual resultado você obtém ao tentar executar esse programa?
- 4 Escreva uma classe genérica `Pair` que tem dois parâmetros de tipo — `F` e `S` —, cada um representando o tipo do primeiro e segundo elemento do par, respectivamente. Adicione os métodos `get` e `set` ao primeiro e ao segundo elemento do par. [Dica: o cabeçalho da classe deve ser `public class Pair<F,S>`].

Exercícios

- 5 Complete o programa `Pilha`, visto em sala de aula, de forma que seja possível chamar métodos para alguns tipos de dados, como `Double` ou `String`. Esses métodos devem ser responsáveis por empilhar e desempilhar itens, chamando os métodos genéricos já criados.

FIM

- **Tarefa de casa:** estudar o capítulo 20 do livro **Java: Como Programar** do Deitel, décima edição. Se quiser, pode estudar o capítulo 21 também.

Até a próxima!