TOO - Generics

Cassio Seffrin

TOO - Generics

Apartir Java 5
O que é um tipo Genérico
Para que serve?
Exemplo de Array[] simples

TOO - Generics

```
Integer[] intArr = {1, 2, 3, 4, 5};
Double[] douArr = {1.2, 2.1, 3.4, 4.5, 5.6};
String[] strArr = {"Cassio", "Fred", "Juliana"};
```

Como imprimir sem Generics?
Como imprimir com Generics?

TOO - Generics

- Tipos genéricos em Arrays funcionam somente substituindo tipos por referencias como Integer, Float, String. Tipos primitivos (int, char, float) não funcionam
- Os nomes dos parâmetros de tipo por toda declaração do método devem corresponder aqueles declarados na assinatura do método
- O tipo Object pode ser usado invés do conceito generics quando o retorno do método é void.

TOO - Generics

Convenção para determinar nomes para Genéricos

- E – Element (usado para todos elementos do Java Collections Framework, exemplo ArrayList, Set etc.)
- K – Key (usado na coleção Map)
- N – Number
- T – Type
- V – Value (usado na coleção Map)
- S,U,V etc. – T, U, V etc.. quando não sabemos o tipo

TOO - Generics

Um objeto comparável é capaz de se comparar a outro objeto. A própria classe deve implementar a interface java.lang.Comparable para comparar suas instâncias.

Considere uma classe Filme que tem membros como, nota, nome, ano. Imagine que desejamos ordenar uma lista de filmes com base no ano de lançamento.

Podemos implementar a interface Comparable com a classe Filme e substituir o método compareTo () da interface Comparable.

TOO - Generics

Se desejarmos classificar os filmes por classificação e nomes também.

Quando tornamos um elemento de coleção comparável (fazendo com que ele implemente Comparable), temos apenas uma chance de implementar o método compareTo ().

A solução é usar o Comparator.