ESTRITAMENTE **CONFIDENCIAL**

ELABORATA
INFORMÁTICA

LINGUAGEM JAVA

Generics

Descrição

- Os generics foram incluídos na versão 5 do Java como um recurso para forçar a segurança de tipos em listas de objetos e também para permitir a criação de métodos e classes genéricos.

Aplicações

- Segurança de tipos em listas
- Métodos genéricos
- Classes genéricas

Segurança de tipos em listas

- Segurança de tipos em listas de objetos significa garantir que apenas objetos de um tipo de dados sejam armazenados numa lista declarada para tal tipo de dados.

Situação

```
...
List cidades = new ArrayList();
cidades.add("Curitiba");
cidades.add("Maringá");
cidades.add(new Integer(100));
...
```

Aplicando o generics

```
...
List<String> cidades = new ArrayList<String>();
cidades.add("Curitiba");
cidades.add("Maringá");
cidades.add(new Integer(100));
...
```

Métodos Genéricos

- Um cenário para uso de métodos genéricos seria: Criar uma classe para imprimir vetores de números inteiros e números double. Provavelmente a primeira idéia a surgir seria a de criar dois métodos, um para imprimir números inteiros e um outro para imprimir números double.

Métodos Sobrecarregados

```
...
public static void main(String[] args)
{
  int[] fibonacci = {1, 1, 2, 3, 5, 8, 13};
  double[] temperaturas = {22.5, 25.9, 24.6, 21.2, 19.0};

  System.out.println("Sequencia Fibonacci...");
  mostrarElementos(fibonacci);

  System.out.println("Temperaturas...");
  mostrarElementos(temperaturas);
}
...
```

Métodos Sobrecarregados

```
...
private static void mostrarElementos(int[] elementos)
{
  for (int i=0; i < elementos.length; i++)
    System.out.println(elementos[i]);
}

private static void mostrarElementos(double[] elementos)
{
  for (int i=0; i < elementos.length; i++)
    System.out.println(elementos[i]);
}
...
```

Métodos Sobrecarregados

- A classe foi bem desenhada, foi aplicado a sobrecarga de métodos e esta funcional. Mas agora a classe precisa imprimir vetores contendo caracteres. Qual seria a solução? Criar um terceiro método sobrecarregado para imprimir o vetor de caracteres. É uma possível solução, mas não a ideal.

Solução?

- A solução ideal seria criar apenas um método que utilize Generico como tipo de dado do parâmetro.

Métodos Sobrecarregados

```
...
public static void main(String[] args)
{
 int[] fibonacci = {1, 1, 2, 3, 5, 8, 13};
 double[] temperaturas = {22.5, 25.9, 24.6, 21.2, 19.0};

  System.out.println("Sequencia Fibonacci...");
  mostrarElementos(fibonacci);

  System.out.println("Temperaturas...");
  mostrarElementos(temperaturas);
}
...
```

Métodos Sobrecarregados

```
...
public static void main(String[] args)
{
  Integer[] fibonacci = {1, 1, 2, 3, 5, 8, 13};
  Double[] temperaturas = {22.5, 25.9, 24.6, 21.2, 19.0};

  System.out.println("Sequencia Fibonacci...");
  mostrarElementos(fibonacci);

  System.out.println("Temperaturas...");
  mostrarElementos(temperaturas);
}
...
```

Métodos Sobrecarregados

```
...
private static void mostrarElementos(int[] elementos)
{
  for (int i=0; i < elementos.length; i++)
    System.out.println(elementos[i]);
}

private static void mostrarElementos(double[] elementos)
{
  for (int i=0; i < elementos.length; i++)
    System.out.println(elementos[i]);
}
...
```

Método Genérico

```
...
private static <T> void mostrarElementos(T[] elementos)
{
  for (int i=0; i < elementos.length; i++)
    System.out.println(elementos[i]);
}
...
```

Padronização das Letras

- E - Element
- K - Key
- V - Value
- T - Type
- N - Number
- S, U, V 2nd, 3rd, 4th Types

Classes Genéricas

- Generics podem ser utilizados na definição de classes genéricas. Uma classe genérica é a classe que terá o tipo do objeto utilizado definido como genérico.

Cenário para Classes Genéricas

- Uma classe para imprimir uma lista de objetos do tipo cliente.

Exemplo

```
public class Cliente
{
  private String nome;
  private String telefone;

  public Cliente() {}

  public Cliente(String nome, String telefone)
  {
    this.nome = nome;
    this.telefone = telefone;
  }

  // Getters e Setters
}
...
```

Exemplo

```
public class Listagem<E>
{
  private List<E> lista = new ArrayList<E>();

  public List<E> getLista()
  {
    return lista;
  }

  public void setLista(List<E> lista)
  {
    this.lista = lista;
  }

  public void adicionar(E item)
  {
    lista.add(item);
  }
}
...
```

Exemplo

```
public class ImprimirListagemGenerica
{
  public static void main(String[] args)
  {
    ListagemGenerica<Cliente> listagem = new
         ListagemGenerica<Cliente>();

    listagem.adicionar(new Cliente("Antonio Pedrusco",
         "99342123"));
    listagem.adicionar(new Cliente("Marieta da Penha",
         "32233344"));

    for (int i=0; i < listagem.getLista().size(); i++)
      System.out.println(listagem.getLista().get(i).getNome());
  }
}
```

Exercício 01

- Aplicar Generics para eliminar todos os warnings da versão do software CadastroBebidas para Hibernate.

Exercício 02

- No software CadastroBebidas, substituir a classe DAO atual por uma classe DAO genérica.

OBRIGADO!

ESTRITAMENTE **CONFIDENCIAL**

ELABORATA
INFORMÁTICA

www.**elaborata**.com.br

Horário de Atendimento Comercial
Segunda à sexta – das 9:00h às 19:30h e
Sábado - das 8:00h às 15:00h.

Rua Monsenhor Celso, 256 - 1º Andar
Centro - Curitiba - PR

41.3324.0015
41.99828.2468

cursos@elaborata.com.br