Generics

jbbf

1.Introdução

- Os Generics em Java foram introduzidos no JDK 5.0 com o objetivo de reduzir erros e melhorar a qualidade de nosso código.

2.A necessidade de usar Generics

- Vamos imaginar um cenário em que queremos criar uma lista em Java para armazenar Integer; podemos ser tentados a escrever o seguinte trecho de código:

```
1  List list = new LinkedList();
2  list.add(new Integer(1));
3  Integer i = list.iterator().next();
```

Vai dar erro de compilação aqui!

2.A necessidade de usar Generics

- O problema no código anterior é que não há garantia que o tipo de retornado pela lista seja um Integer.
- A lista poderia conter qualquer objeto.
- Sabemos apenas que é um objeto, portanto, uma conversão explícita para garantir que o tipo é seguro, é necessária.

```
1 Integer i = (Integer) list.iterator.next();
```

Um typecasting resolve o problema

2.A necessidade de usar Generics

- Seria muito mais fácil se pudéssemos expressar a intenção de usar tipos específicos e o compilador pudesse garantir a exatidão desse tipo.
- Essa é a ideia central por trás dos genéricos. Vamos modificar a primeira linha do código anterior para usar Generics:

```
1  List list = new LinkedList();
```



```
1  List<Integer> list = new LinkedList<>();
```

2.A necessidade de usar Generics

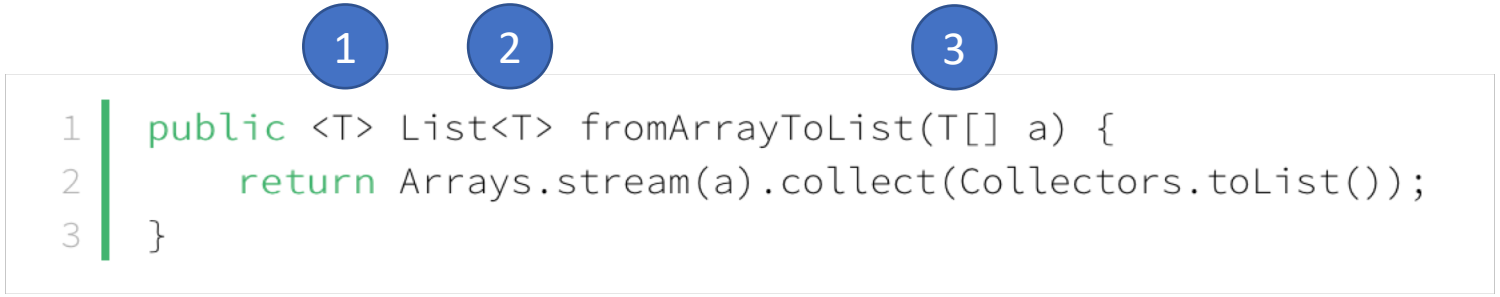
- Adicionando o operador <> contendo o tipo, restringimos a especialização desta lista apenas ao tipo Integer, ou seja, especificamos o tipo que será mantido dentro da lista.
- O compilador pode aplicar o tipo em tempo de compilação.
- Em pequenos programas, isso pode parecer uma adição trivial, no entanto, em programas maiores, isso pode adicionar robustez significativa e facilitar a leitura do programa.

3.Métodos Generics

- Métodos genéricos são métodos com uma única assinatura que podem ser chamados com diferentes tipos de argumentos. O compilador é que garante a exatidão de qualquer tipo usado.
- Estas são algumas propriedades de métodos genéricos:
  - Métodos genéricos possuem uma declaração de tipo de parâmetro antes do tipo de retorno da declaração de método;
  - Métodos genéricos podem ter diferentes tipos de parâmetros separados por vírgulas na assinatura do método
  - O Corpo do método é como um método normal

4.Exemplo

- Um exemplo de definição de um método genérico para converter uma matriz em uma lista:



1. Declaração de tipo de parâmetro antes do tipo de retorno
2. Retorna uma lista do tipo genérico T
3. Recebe uma matriz (array) do tipo genérico T