



08. Classes anônimas e Lambdas

📅 Date	@06/10/2022
▼ Categoria	Java
▼ Curso	Java e java.util: Coleções Wrappers e Lambda expressions

Tópicos

- Classes anônimas
- Qual objeto?
- O que acontece?
- Finalmente Lambdas
- E se fosse com lambda?
- Foreach com lambda
- O padrão Iterator
- Mãos na massa: Classes anônimas
- Mãos na massa: Lambdas
- Conclusão
- Projeto final

Qual objeto?

Como chamamos esse objeto que só encapsula uma função/método/procedimento?

- Function Object

- Um objeto que criamos para encapsular uma função ou método. As classes anônimas facilitam um pouco a criação desses objetos.

O padrão Iterator

Você já sabe agora que existem muitas coleções. Só nesse treinamento vimos `ArrayList`, `LinkedList` e `Vector`. Se você assistir ainda o curso dedicado as coleções você aprenderá as interfaces para fila (`Queue`), conjunto (`Set`) e mapa (`Map`) cada uma com várias implementações.

Aí vem uma pergunta: Como posso acessar (*iterar*) todas essas implementações de maneira uniforme sem saber os detalhes de cada implementação? A resposta está na "caixa de padrões de projeto" e se chama `Iterator`.

Uma `Iterator` é um objeto que possui no mínimo dois métodos: `hasNext()` e `next()`. Ou seja, você pode usá-lo para perguntar se existe um próximo elemento e pedir o próximo elemento. A notícia boa é que isso funciona com TODAS as implementações e aí a grande vantagem.

Veja o código para usar o `Iterator` de uma lista:

```
List<String> nomes =new ArrayList<>();
nomes.add("Super Mario");
nomes.add("Yoshi");
nomes.add("Donkey Kong");

Iterator<String> it = nomes.iterator();

while(it.hasNext()) {
    System.out.println(it.next());
}
```

Se você entendeu esse código, você já aprendeu como iterar com filas, conjuntos ou mapas. Veja o uso do `Iterator` através de um conjunto:

```
Set<String> nomes =new HashSet<>();
nomes.add("Super Mario");
nomes.add("Yoshi");
```

```
nomes.add("Donkey Kong");

Iterator<String> it = nomes.iterator();

while(it.hasNext()) {
    System.out.println(it.next());
}
```

Se ficou ainda com dúvida sobre o `Iterator`, sem problemas (!) pois o curso [Dominando as Collections](#) possui um capítulo dedicado.
