



06. Aplicando o Set no modole

📅 Date	@27/10/2022
📁 Categoria	Java
📖 Curso	Java Collections: Dominando Listas Sets e Mapas

Tópicos

- Aplicando o Set no modelo
- Criação da classe aluno
- Encapsulando o Set
- Collections.unmodifiableSet(..)
- Collections.emptySet()
- Para saber mais: Coleções threadsafe

Collections.emptySet()

No código abaixo usamos mais um método da classe `Collections`. Nesse caso criamos um conjunto vazio.

Será que podemos adicionar um elemento nesse conjunto? Execute o código a seguir para ter certeza!

```
import java.util.*;

public class TesteEmptySet {

    public static void main(String[] args) {
```

```
Set<String> nomes = Collections.emptySet();
nomes.add("Paulo"); //o que acontece aqui?
}

}
```

Também não podemos! Recebemos a mesma exceção: `UnsupportedOperationException`

Um conjunto destinado a ser vazio não pode ter um elemento, certo? Então faz sentido receber essa exceção. Mas para que um conjunto vazio poderia ser útil?

Por exemplo, imagina que você precisa representar um curso que foi cancelado pois não teve matrículas. Nesse caso faria todo sentido devolver um `Collections.emptySet()`

Para saber mais: Coleções thread-safe

Já usamos bastante a classe `java.util.Collections`. Vimos os métodos `Collections.unmodifiableSet(..)` e `Collections.emptySet(..)` entre vários outros. Vale muito explorar essa classe!

Tem um método que gostaria de chamar atenção:

```
Set<Aluno> alunosSincronizados = Collections.synchronizedSet(alunos);
```

Uma das características mais interessantes de JVM é que ela sabe trabalhar em paralelo. Internamente isso é feito por meio de **Threads** que funcionam como pequenos processos dentro da JVM.

O problema é que as coleções que estamos usando até agora não foram feitas para serem manipuladas em paralelo. No entanto, nada impede que usemos um método da classe `Collections` para transformar uma coleção comum em uma coleção para threads. É justamente isso que o método faz, retorna uma nova coleção que pode ser compartilhada entre threads sem perigos.

O tópico de Threads é importante pois elas são muito utilizadas dentro das bibliotecas que rodam no

mundo Java. Na Alura, temos dois treinamentos dedicados a threads:

- <https://cursos.alura.com.br/course/threads-java-1>
 - <https://cursos.alura.com.br/course/threads-java-2>
-