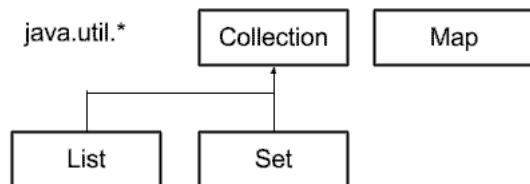


Para saber mais: Outras coleções thread-safe

Vimos na aula a classe `java.util.Vector` em ação, mas a API de Collections possuem muito mais interfaces e implementações.

Dentro dessa API existem 4 interfaces principais: `java.util.Collection`, `java.util.List`, `java.util.Set` e `java.util.Map`.



Se a classe `Vector` é a versão thread-safe de uma lista, será que existem para as outras interfaces implementações thread-safe? Claro que sim!

Para os mapas (`Map`) podemos usar a antiga classe `Hashtable`:

```
Map mapaThreadSafe = new Hashtable();
```

E também temos uma implementação mais recente e performática de mapas, a classe `ConcurrentHashMap`:

```
//do pacote java.util.concurrent
Map mapaThreadSafe = new ConcurrentHashMap();
```

Para o `Set` (conjunto) não existe uma implementação pronta na API padrão do Java mas podemos utilizar a classe `Collections` para construir um `Set` sincronizado:

```
Set conjunto = Collections.synchronizedSet(new HashSet());
```

Existe um curso na alura sobre as Collections [JAVA Collections](https://cursos.alura.com.br/course/java-collections) (<https://cursos.alura.com.br/course/java-collections>).