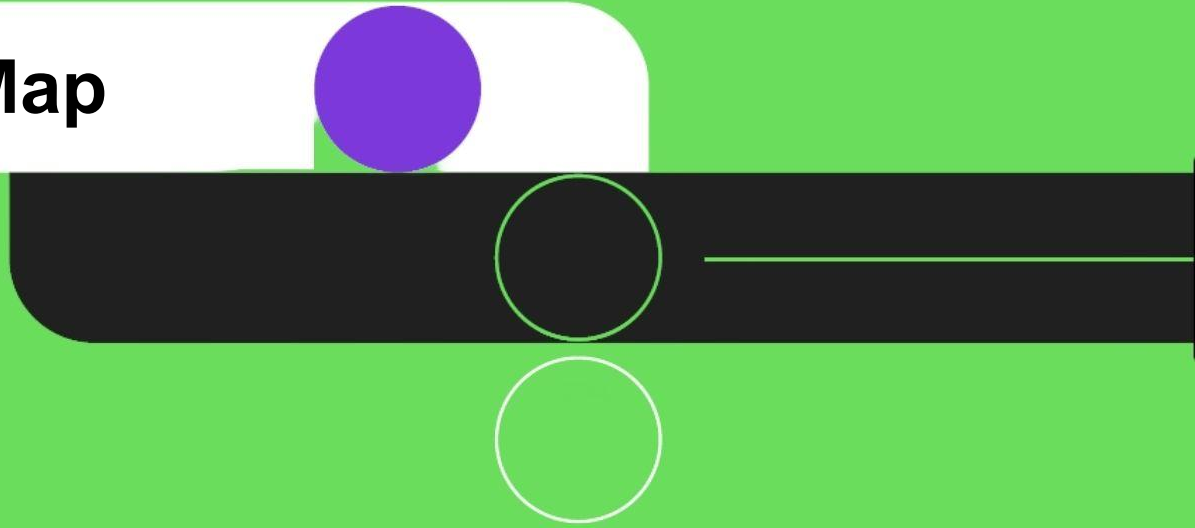
GAMA ACADEMY

Coleções

Queue, Stack e Map



01 Escolhendo uma coleção

02 Wrappers

03 **QUEUE**

04 MAP

Collections

Escolhendo uma coleção

A documentação para cada coleção discute os requisitos de memória e as características de desempenho dos métodos para operações como adição e remoção de elementos, pesquisa de elementos, classificação de elementos etc.

Antes de escolher uma coleção, revise a documentação on-line para a categoria da coleção que você está considerando (Set, List, Map, Queue etc.), então selecione a implementação que melhor atende às necessidades de seu aplicativo.

documentação

Classes empacotadoras de tipo

Wrappers

Todo tipo primitivo tem uma classe empacotadora de tipo correspondente (no pacote java.lang).

Essas classes chamam-se **Boolean, Byte, Character, Double, Float, Integer, Long e Short**.

Elas permitem manipular valores de tipo primitivo como objetos. Isso é importante porque as estruturas de dados Set, Queue e List (e suas classes derivadas) manipulam e compartilham objetos elas não podem manipular variáveis de tipos primitivos. Mas podem manipular objetos das classes empacotadoras de tipo(Wrappers), porque cada classe em última análise deriva de Object.

Autoboxing e auto-unboxing

O boxing converte um valor primitivo em um objeto da classe empacotadora de tipo correspondente. O unboxing converte um objeto empacotador de tipo no valor primitivo correspondente.

• O Java executa automaticamente conversões boxing e unboxing.

Autoboxing

```
Integer i = 1;
System.out.println(i instanceof Integer); // true
System.out.println(i.getClass().getSimpleName()); // Integer
```

QUEUE

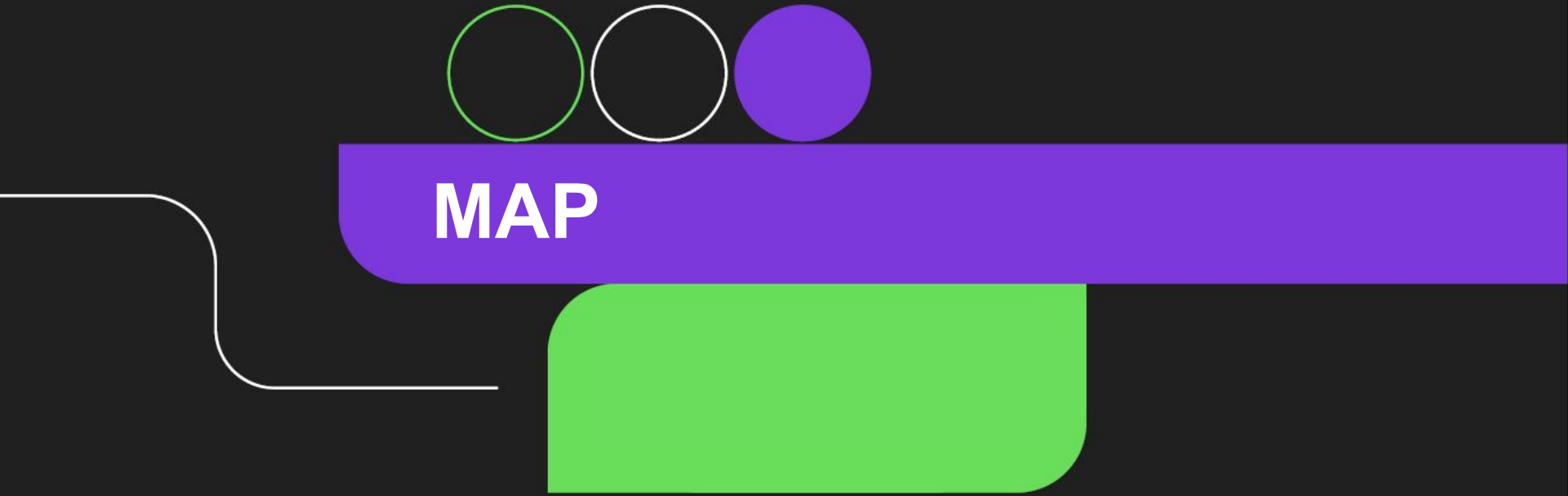
QUEUE

A interface Java Queue ordena o elemento de maneira FIFO (First In First Out). No FIFO, o primeiro elemento é removido primeiro e o último elemento é removido por último.

Lembre-se de que uma fila é uma coleção que representa uma fila de espera — normalmente, inserções são feitas na parte de trás de uma fila e exclusões são feitas a partir da frente.

Declaração - Queue

```
Queue<String> fila = new LinkedList<>();
```


MAP

MAP

Mapeia chaves para valores. Cada elemento tem na verdade dois objetos: **uma chave e um valor**. Valores podem ser duplicados, mas chaves não. SortedMap é uma interface que estende Map, e permite classificação ascendente das chaves. As chaves em um Map **devem ser únicas**, mas os valores associados **não precisam ser**.

Três das várias classes que implementam a interface Map são Hashtable, HashMap e TreeMap. Hashtables e HashMaps armazenam elementos em tabelas de hash e TreeMaps armazenam elementos em árvores.

Declaração - Map

```
// cria HashMap para armazenar chaves de Strings e valores Integer
Map<String, Integer> mapa = new HashMap<>();
```

GAMA ACADEMY

VAMOS PRATICAR?