

JAVA LEGENDS

Conquiste o Mundo da Programação



HANNIEL VIEIRA

Principais Collections em Java

O que é Collections?

As collections em Java são estruturas de dados que permitem armazenar e manipular grupos de objetos. Elas são essenciais para qualquer programador Java, oferecendo uma maneira eficiente de gerenciar dados. Vamos explorar as principais collections de Java e ver exemplos de como usá-las em contextos reais.



01

ArrayList:

Lista Dinâmica

A ArrayList é uma das implementações mais utilizadas da interface List. Ela permite o armazenamento dinâmico de elementos, ou seja, seu tamanho pode aumentar ou diminuir conforme necessário.

Exemplo: Gerenciando uma lista de tarefas

```
public class TaskManager {  
    public static void main(String[] args) {  
        ArrayList<String> tasks = new ArrayList<>();  
        tasks.add("Estudar Java");  
        tasks.add("Fazer exercícios");  
        tasks.add("Participar de reunião");  
  
        for (String task : tasks) {  
            System.out.println(task);  
        }  
  
        tasks.remove("Fazer exercícios");  
        System.out.println("Tarefas atualizadas: " + tasks);  
    }  
}
```

snappify.com



02

HashMap:

Mapeamento

de Chave-Valor

O HashMap permite o armazenamento de pares chave-valor. É muito útil quando precisamos associar um valor específico a uma chave única.

Exemplo: Contagem de itens em um inventário

```
import java.util.HashMap;

public class InventoryManager {
    public static void main(String[] args) {
        HashMap<String, Integer> inventory = new HashMap<>();

        inventory.put("Espada", 10);
        inventory.put("Escudo", 5);
        inventory.put("Poção de Cura", 20);

        for (String item : inventory.keySet()) {
            System.out.println(item + ": " + inventory.get(item));
        }

        inventory.put("Espada", 15);

        System.out.println("Inventário atualizado: " + inventory);
    }
}
```

snappify.com



03

LinkedList: Lista de Elementos Encadeados

A LinkedList é uma implementação da interface List que permite a inserção e remoção eficientes de elementos em qualquer posição da lista.

Exemplo: Gerenciando uma fila de atendimento

```
import java.util.LinkedList;
import java.util.Queue;

public class CustomerService {
    public static void main(String[] args) {
        Queue<String> customers = new LinkedList<>();

        customers.add("Cliente1");
        customers.add("Cliente2");
        customers.add("Cliente3");

        System.out.println("Atendendo: " + customers.poll());

        System.out.println("Fila atualizada: " + customers);
    }
}
```

snappify.com



04

TreeMap: Mapeamento Ordenado

O TreeMap é similar ao HashMap, mas mantém os pares chave-valor ordenados de acordo com as chaves.

Exemplo: Ordenando notas de alunos

```
import java.util.TreeMap;

public class GradeManager {
    public static void main(String[] args) {
        TreeMap<String, Integer> grades = new TreeMap<>();

        grades.put("Ana", 85);
        grades.put("Carlos", 92);
        grades.put("Bianca", 78);

        for (String student : grades.keySet()) {
            System.out.println(student + ": " + grades.get(student));
        }
    }
}
```

snappify.com



Conhecer e saber utilizar as diferentes collections em Java é fundamental para escrever código eficiente e organizado.

Cada tipo de collection tem suas particularidades e é mais adequado para certas situações. Experimente os exemplos acima e explore o poder das collections em seus projetos Java.



OBRIGADO POR VER ATÉ AQUI

Criado por Hanniel Viera com ajudar do chatgpt e midjourney

[hanniel09 \(Hanniel Vieira\) \(github.com\)](https://github.com/hanniel09)

