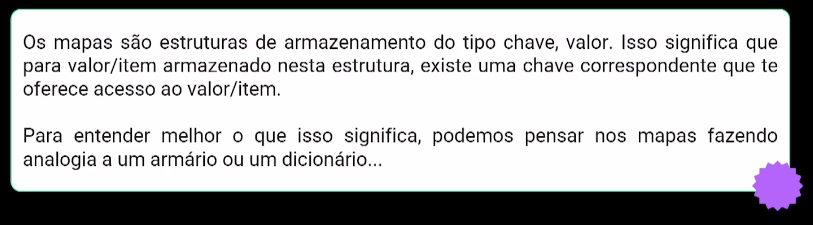
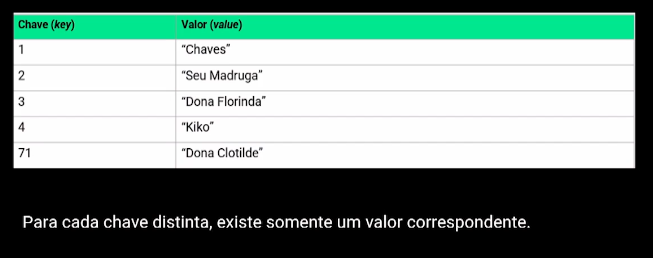
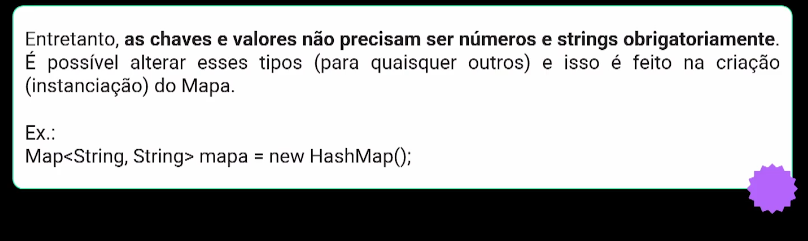
**Mapas:**

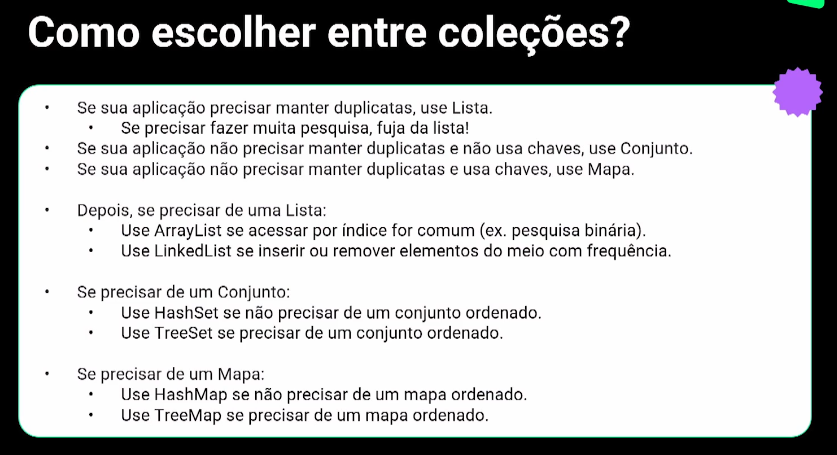




Procurar pela Chave.

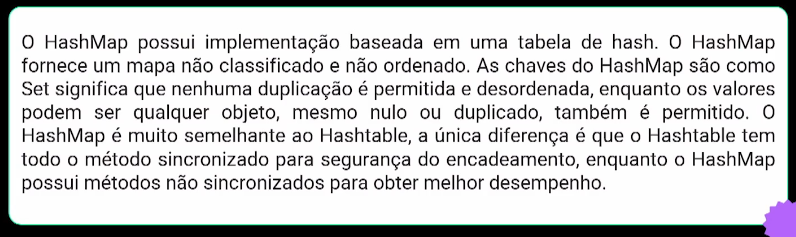


A 1ª diz o tipo de conteúdo da chave, a 2ª diz o tipo de conteúdo do nosso valor.

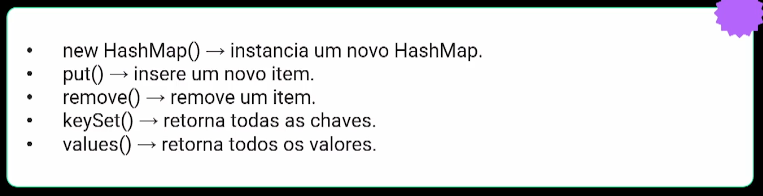


**HashMap e HashTable:**

**HashMap**: (não tem ordenação)

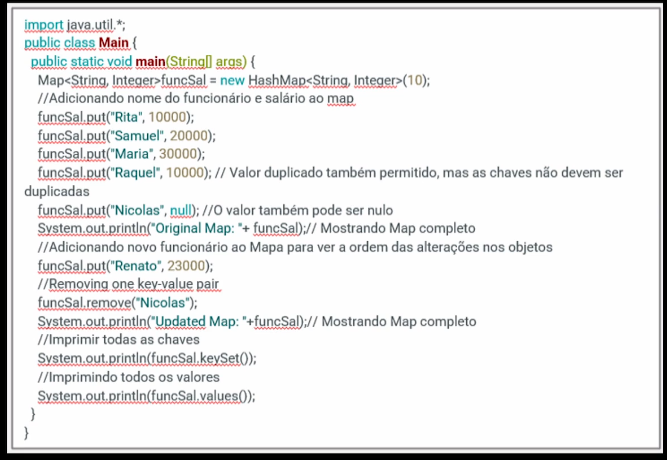


Principais métodos:

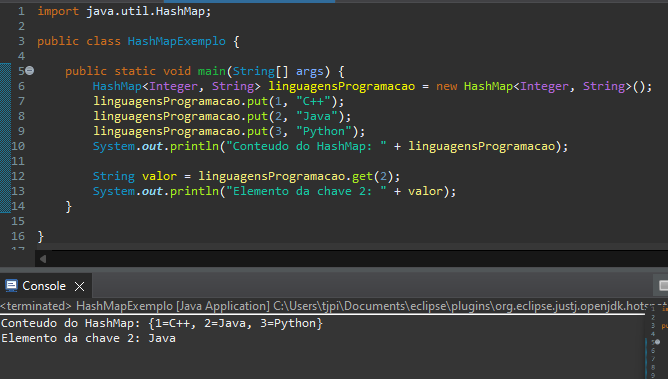


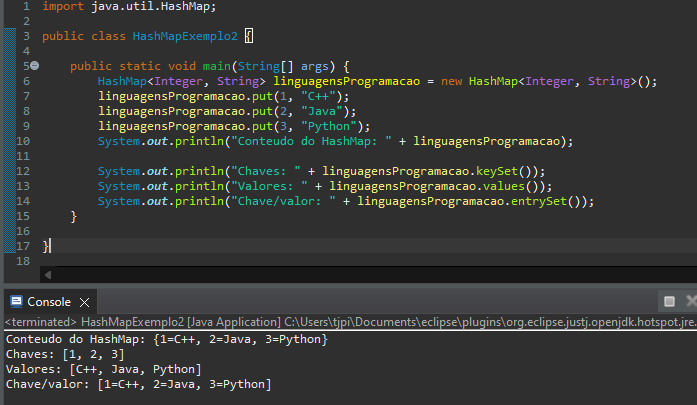
Na verdade, a coluna key é um set enquanto a coluna valor é um list.

EX:

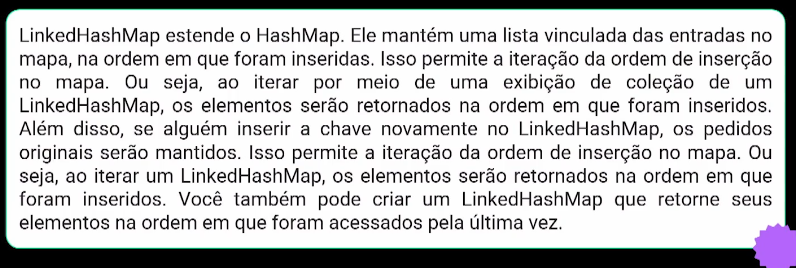


Pratica:

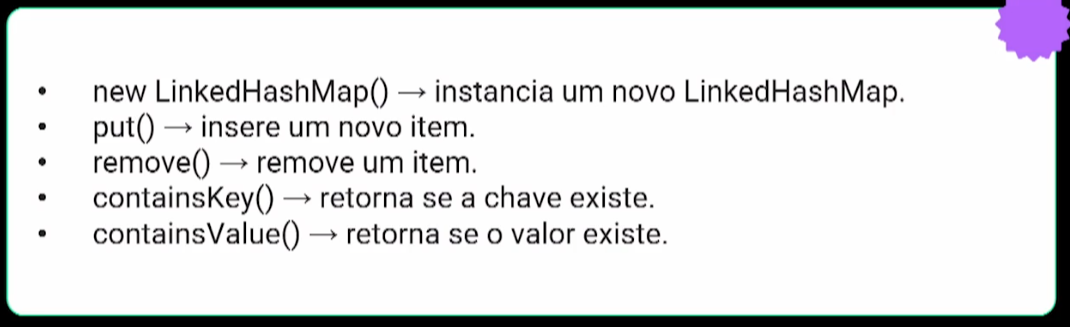




**LinkedHashMap: (**mantem a ordem cronologica**)**



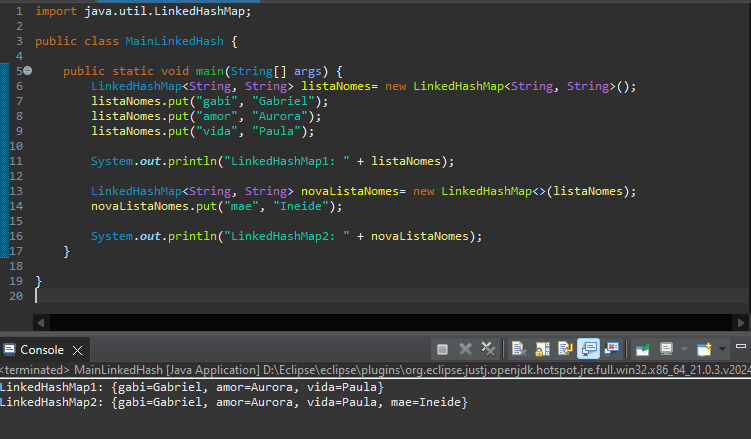
Principais Métodos:



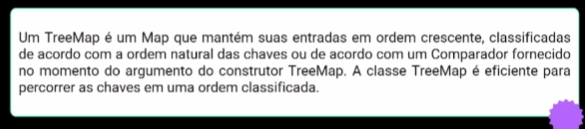
EX:



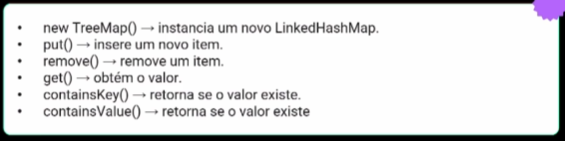
Pratica:



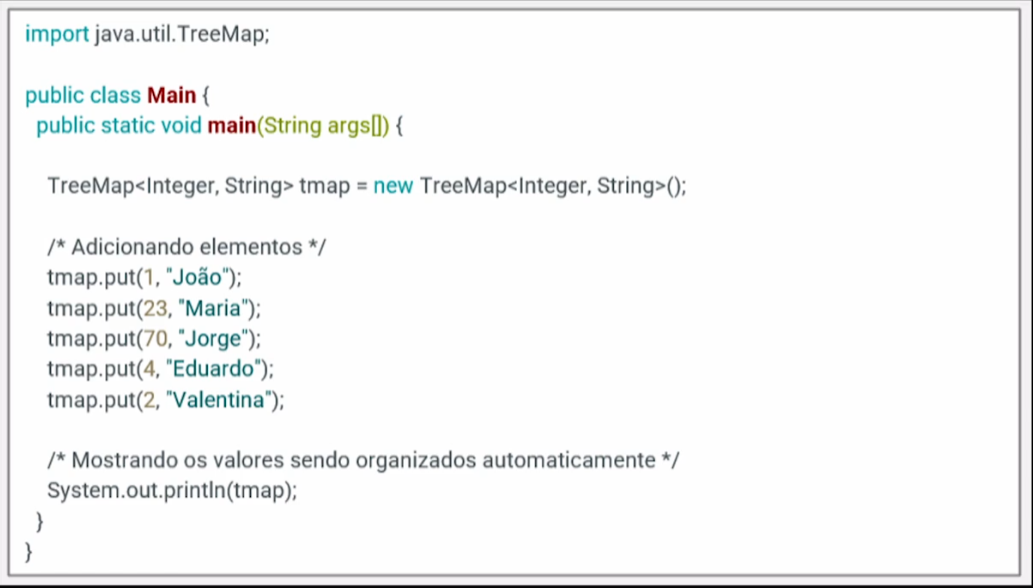
**TreeMap: (**ordem crescente**)**



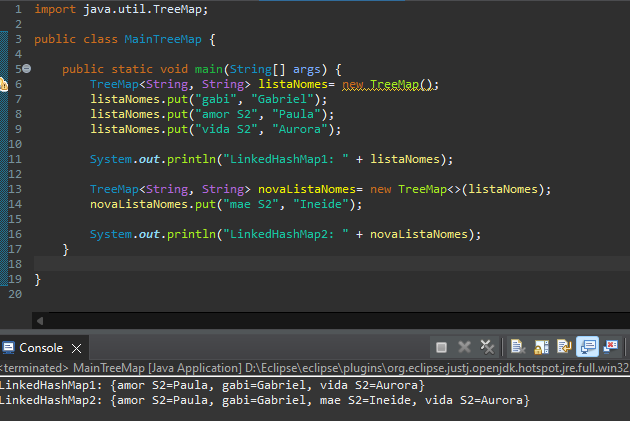
Principais Metodos:



Ex:



Pratica:



**Resumo Final**

* **HashMap** é ideal para cenários onde a ordem das chaves não importa e a velocidade é crucial.
* **LinkedHashMap** é útil quando a ordem de inserção ou a ordem de acesso precisa ser mantida.
* **TreeMap** deve ser usado quando é necessário manter as chaves ordenadas em uma sequência específica.

Atividade:

import java.util.HashMap;

import java.util.Map;

import java.util.Scanner;

public class CPF {

public static void main(String[] args) {

Scanner input = new Scanner(System.***in***);

String name, cpf, option;

Map<String, String> consultCpf = new HashMap();

do {

System.***out***.println("Escolha uma opção:");

System.***out***.println("1. Adicionar nova pessoa");

System.***out***.println("2. Remover pessoa por CPF");

System.***out***.println("3. Exibir todos os CPFs e nomes");

System.***out***.println("4. Buscar nome por CPF");

System.***out***.println("5. Sair");

option = input.nextLine();

switch (option) {

case "1":

System.***out***.println("Qual o seu nome?");

name = input.nextLine();

System.***out***.println("Digite seu CPF:");

cpf = input.nextLine();

if (consultCpf.containsKey(cpf)) {

System.***out***.println("CPF já cadastrado.");

} else {

consultCpf.put(cpf, name);

System.***out***.println("Pessoa adicionada com sucesso!");

}

break;

case "2":

System.***out***.println("Digite o CPF que deseja remover:");

cpf = input.nextLine();

if (consultCpf.remove(cpf) != null) {

System.***out***.println("Pessoa removida com sucesso!");

} else {

System.***out***.println("CPF não encontrado.");

}

break;

case "3":

System.***out***.println("Pessoas e CPFs cadastrados:");

for (Map.Entry<String, String> entry : consultCpf.entrySet()) {

System.***out***.println("CPF: " + entry.getKey() + ", Nome: " + entry.getValue());

}

break;

case "4":

System.***out***.println("Digite o CPF para buscar o nome:");

cpf = input.nextLine();

String result = consultCpf.get(cpf);

if (result != null) {

System.***out***.println("Nome correspondente: " + result);

} else {

System.***out***.println("CPF não encontrado.");

}

break;

case "5":

System.***out***.println("Saindo do sistema...");

break;

default:

System.***out***.println("Opção inválida. Tente novamente.");

break;

}

try {

Thread.*sleep*(3000);

}catch(InterruptedException e) {

e.printStackTrace();

}

System.***out***.println("-------------");

} while (!option.equals("5"));

input.close();

}

}