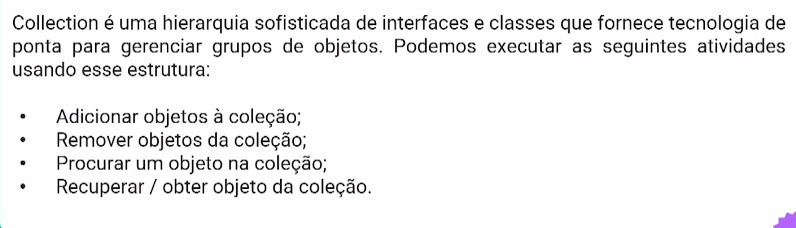
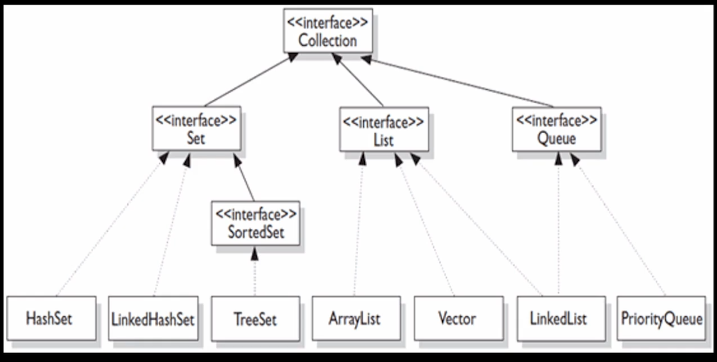
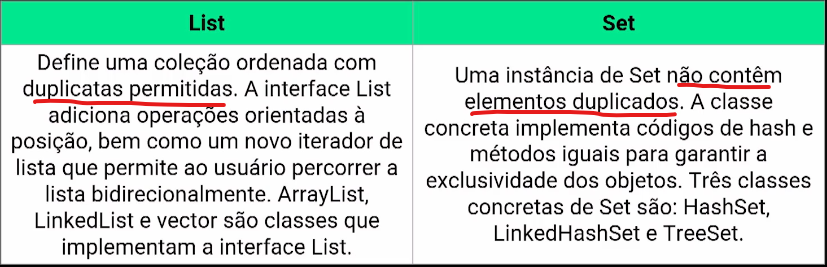
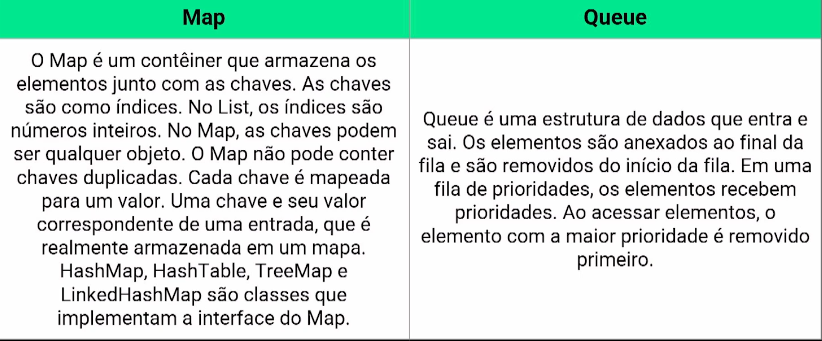
**Collection:**

Uma estrutura de dados é uma coleção de dados organizados de alguma maneira. A estrutura não armazena dados, mas também suporta operações para acessar e manipular os dados. O pacote java.util contem um dos subsistemas mais poderosos do java: Collections.



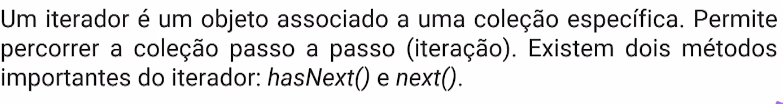




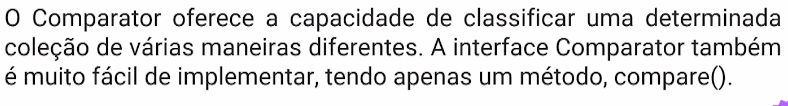


**Interface iterator e comparator:**

Iterator: para percorrer um vetor a gente não faz um for? Ele faz exatamente isso com uma coleção.



Comparator: comparar elementos.

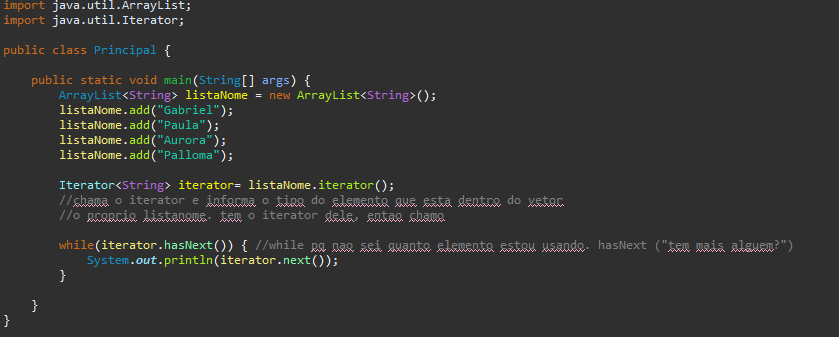




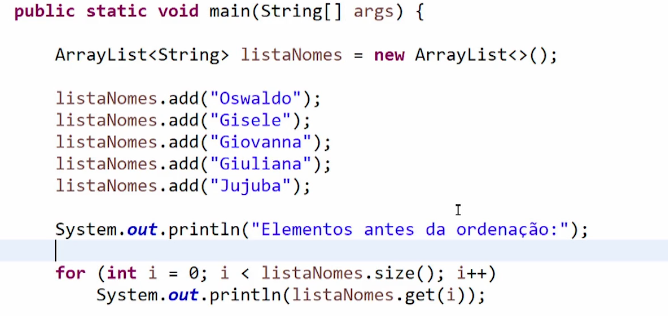
Ex pratico:

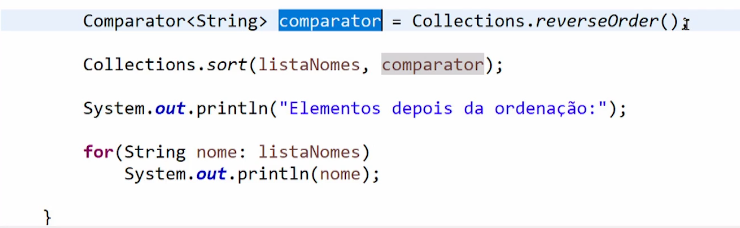


Embora funcione com o for, não é a melhor forma.

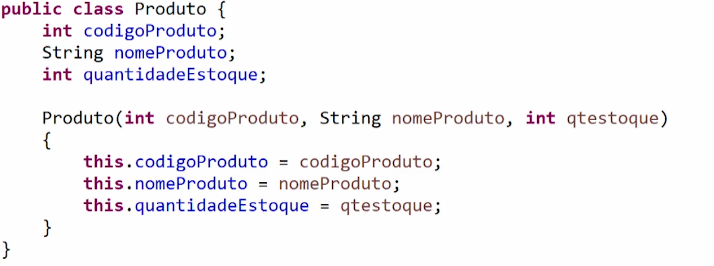


E o comparato?



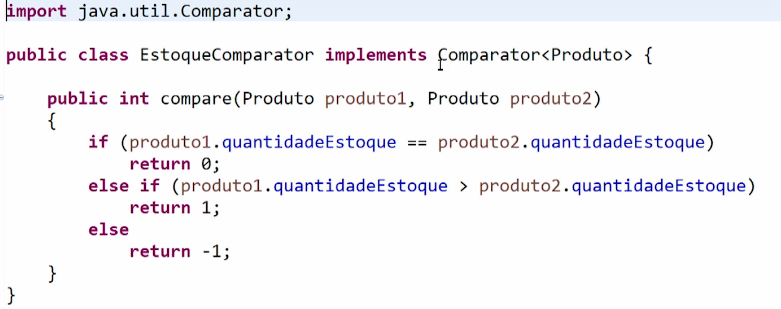


Outro exemplo:



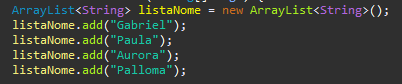


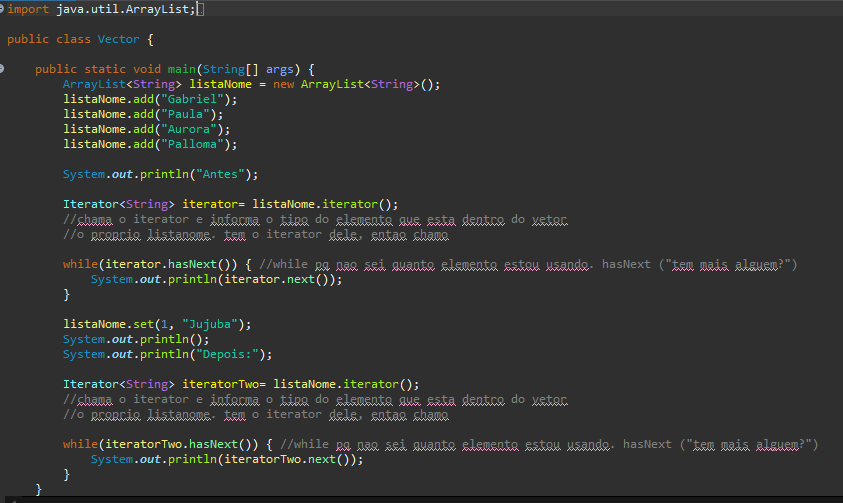


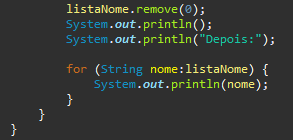
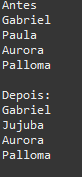


**ArrayList e Vector:**

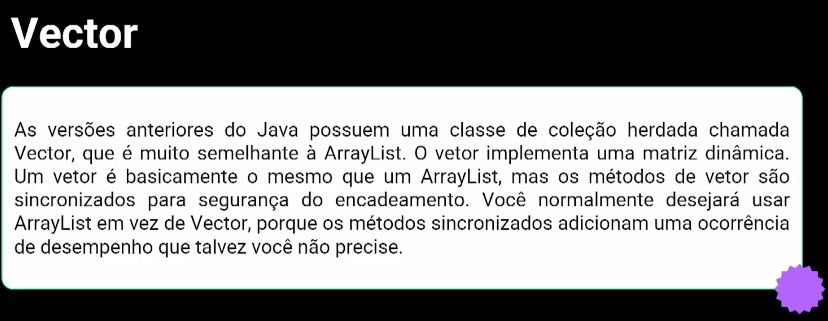
Array: (quando vou usar uma interface, só coloco List) mas a estrutura continua igual.

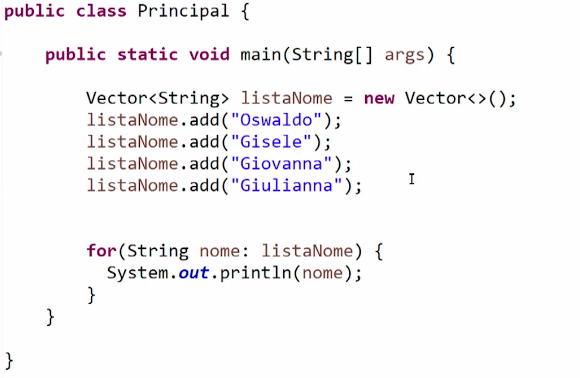






Vector:

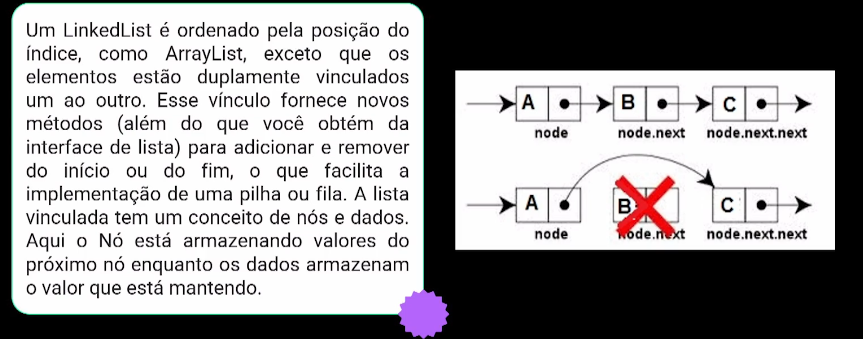




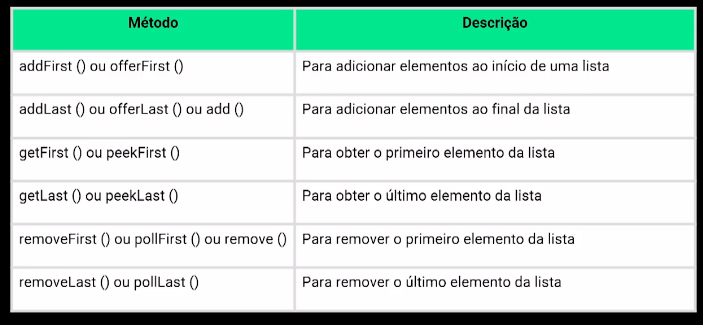
Igual ao arraylist, contudo ele é protegido enquanto o arraylist não.

**LinkedList:**

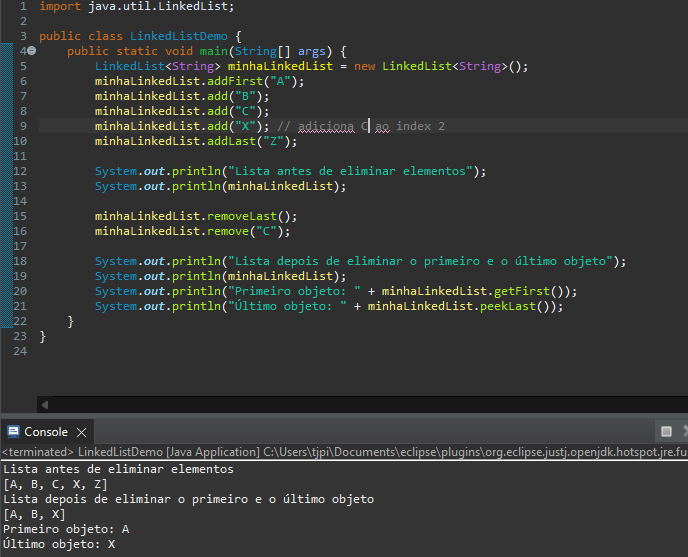
Parecido com o Arraylist, trabalha com referencia a próximos elementos.



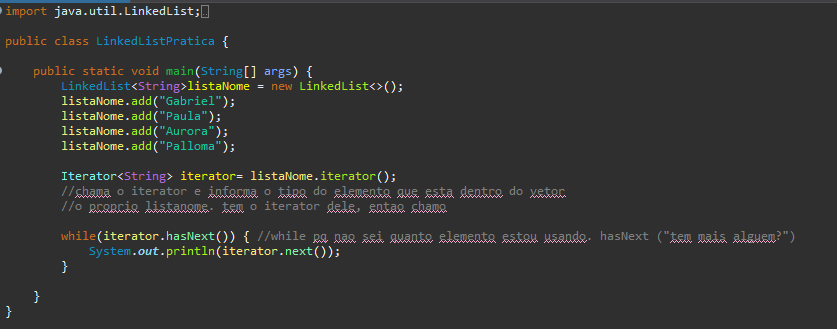
Métodos:



Ex:

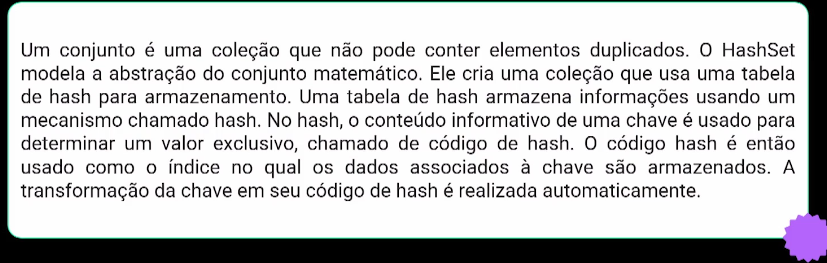


Na pratica:

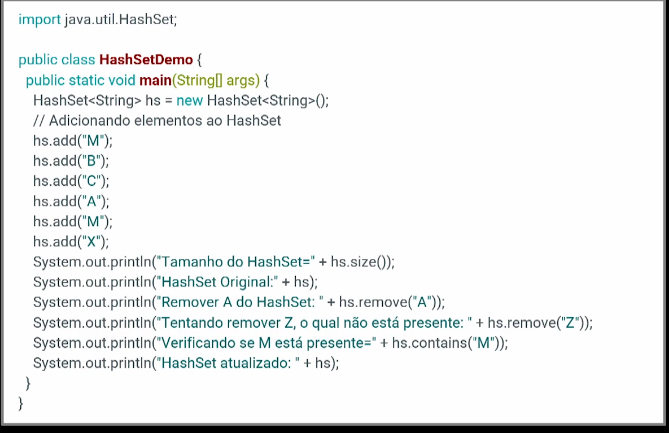


Não permitem duplicar: (termina com set):

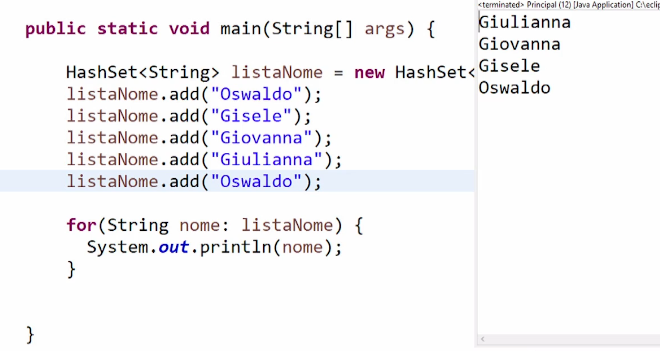
**HashSet:(calculado)**



Ex:

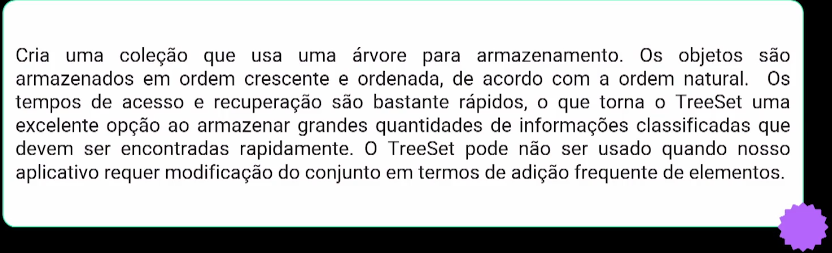


Na pratica:



Não repetiu oswaldo e também não tenho controle na ordenação que os elementos serão exibidos.

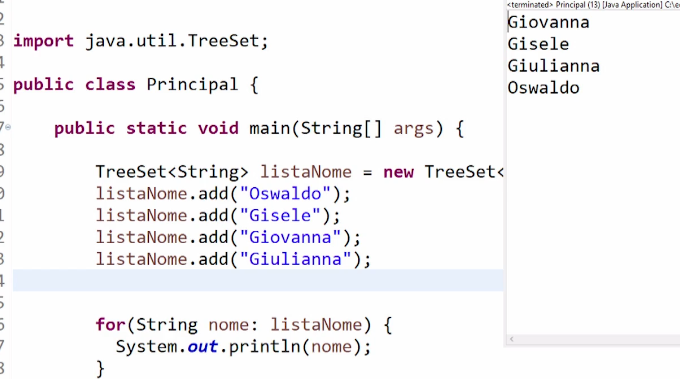
**TreeSet:(Arvore)**



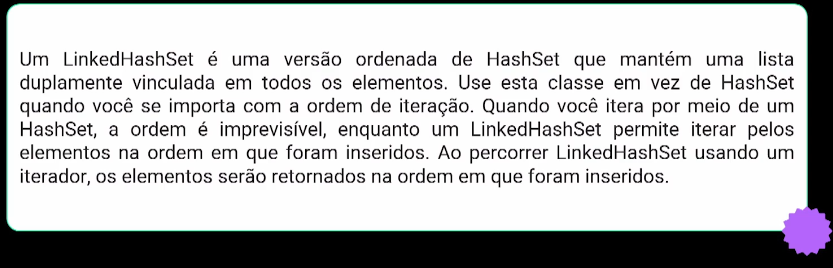
Ex:



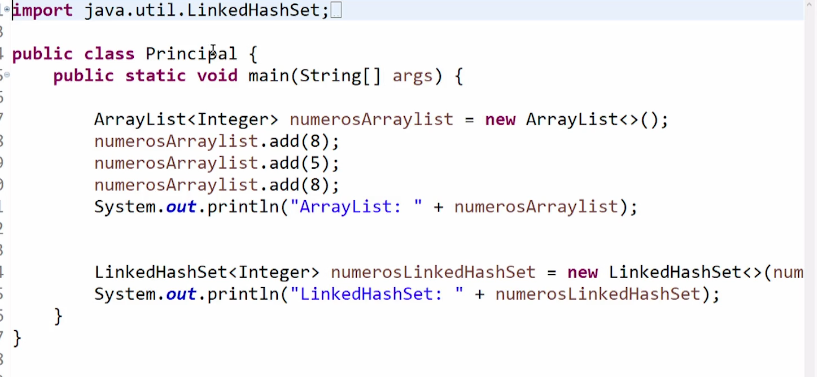
Na pratica:



**LinkedHashSet:**

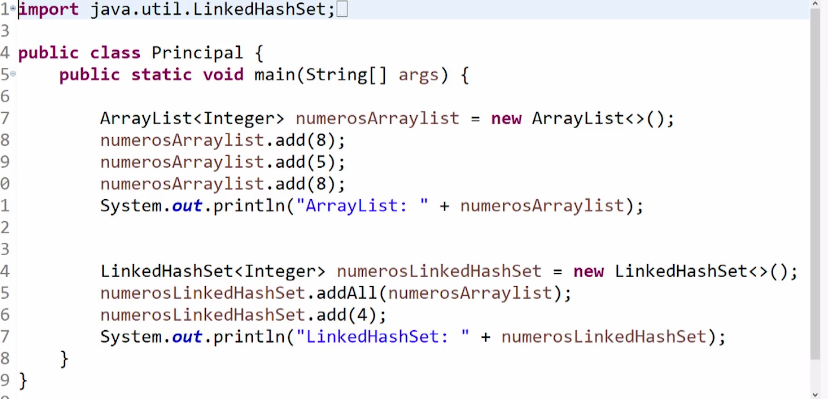


Na pratica:





EX2:





Ex3:

