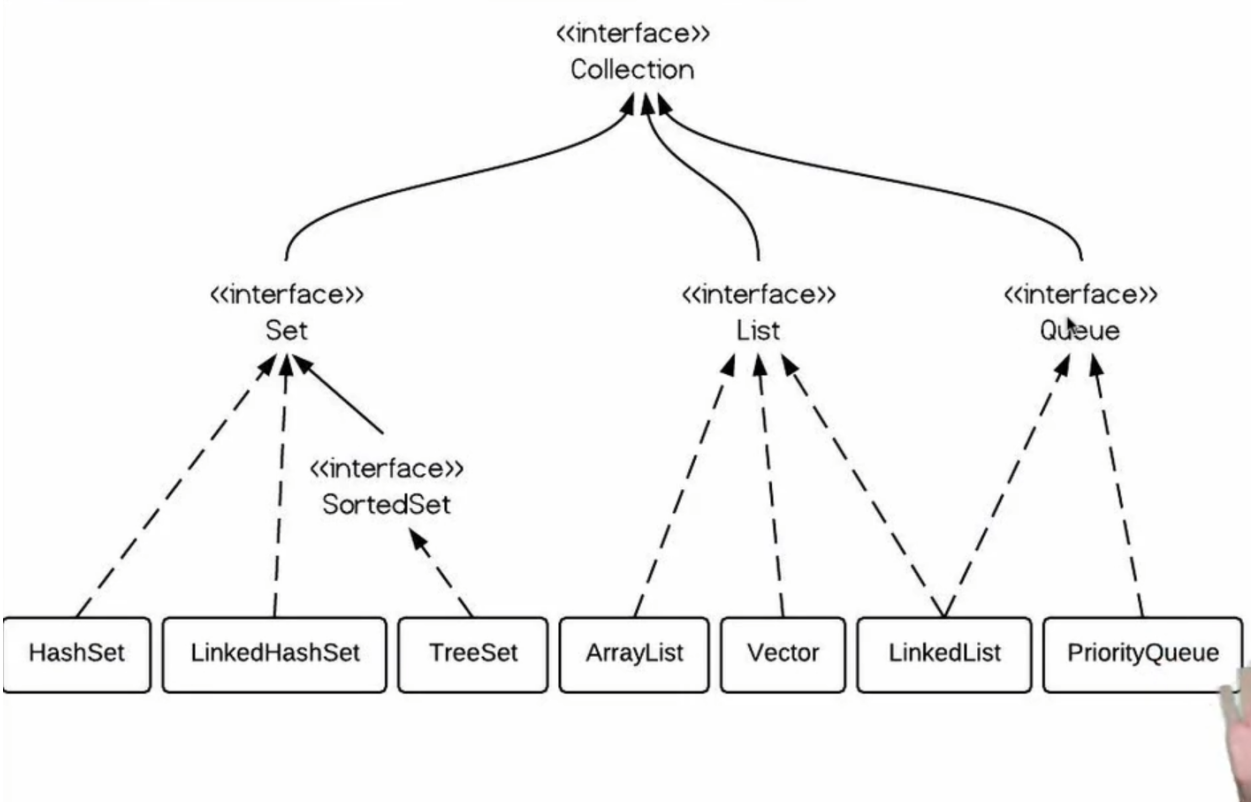
09. Qual Collection usar

Date	@27/10/2022
Categoria	Java
Curso	Java Collections: Dominando Listas Sets e Mapas

Tópicos

- Qual Collection usar
- List vs Set
- A implementações de Collection
- Qual usar?
- Qual coleção?

Hierarquia



List vs Set

Você já ouviu muito sobre essas duas interfaces, a (List) e a (Set), mas qual é a diferença mesmo?

- `List` é uma sequência e aceita elementos duplicados.
- `Set` não aceita duplicados e não define ordem.

Qual usar?

Como um programador deve proceder não sabendo qual das implementações de `Collection` servirá melhor para o nosso sistema?

Provavelmente, caso a modelagem do sistema ainda não esteja bem definida, o desenvolvedor irá utilizar a interface `Collection<E>`. Dessa maneira, terá acesso aos

métodos básicos de todas as implementações,
como `size()`, `add()`, `remove()` e `contains()`. Conforme for sentindo necessidade em algo específico, o desenvolvedor fará poucas mudanças em seu código.

Caso sinta necessidade de fazer uma requisição a um elemento específico através da sua posição, trocará de `Collection<E>` para `List<E>`. Caso perceba que ordem não importa, porém é necessária uma busca bem rápida (e sem repetições), um `Set<E>` é mais apropriado.

Enquanto não sentir essa necessidade, provavelmente a `Collection<E>` será a melhor escolha.

Qual coleção?

Você precisa guardar um monte de alunos em uma coleção e precisa decidir qual implementação irá utilizar.

Sabemos que:

- a coleção deve guardar os alunos ordenados pelo número de matrícula
- a coleção não pode ter elementos repetidos

Reposta

A implementação `TreeSet` já ordena os seus elementos na hora da inserção. Qual é o critério da ordenação depende e pode ser definido através de um `Comparator`.

Explicação de como o linkedHashSet:

LinkedHashSet

Um LinkedHashSet preserva a ordem com base nas inserções (qual elemento foi inserido primeiro).

O principal recurso do TreeSet é a classificação, o LinkedHashSet mantem a ordem de inserção e o HashSet é apenas uma coleção de uso geral para armazenar objetos. O TreeSet é uma implementação de SortedSet que permite manter os elementos na ordem

classificada definida pela interface Comparable ou Comparator, ou seja, quando você insere um novo elemento em um TreeSet, ele verifica em qual posição será inserido esse elemento para manter a ordenação.
