

HashSet é poderoso, mas se eu precisar de acesso ordenado?

No capítulo anterior, aprendemos a lidar com conjuntos e vimos as vantagens de usarmos um `HashSet`. Além de não aceitar elementos duplicados, característica dos conjuntos, podemos realizar buscas extremamente rápidas. Ter um código que processe rapidamente é algo que alegra não apenas o desenvolvedor, mas os usuários que esperam menos os resultados.

Contudo, o `HashSet`, por ser um conjunto, não possui uma ordem previsível. Como assim? Imagine uma sacola de bingo. Dentro dela adicionamos várias bolas numeradas que não se repetem. Se adicionarmos primeiro a bola 11, não quer dizer que quando acessarmos o primeiro elemento do conjunto será a bola 11. Pode ser 15, 34 ou 40. Veja que essa característica do conjunto de não termos uma ordem previsível pode limitar o uso do poderoso `HashSet` em situações que desejamos maior performance, mas precisamos ter um acesso ordenado e previsível.

Mas nem tudo está perdido. No `Collection Framework` há uma estrutura de dados que usa o poder do hash e que podemos acessar os elementos de maneira previsível, isto é, se adicionarmos os elementos A, B e C teremos certeza que A é o primeiro, B é o segundo e por aí vai.

Qual das opções abaixo possui essa estrutura de dados?

Selecione uma alternativa

A `LinkedHashSet`

B `LinkedSet`

C `LinkedList`

D `LinkedHash`