

SCC0504 - Programação Orientada a Objetos

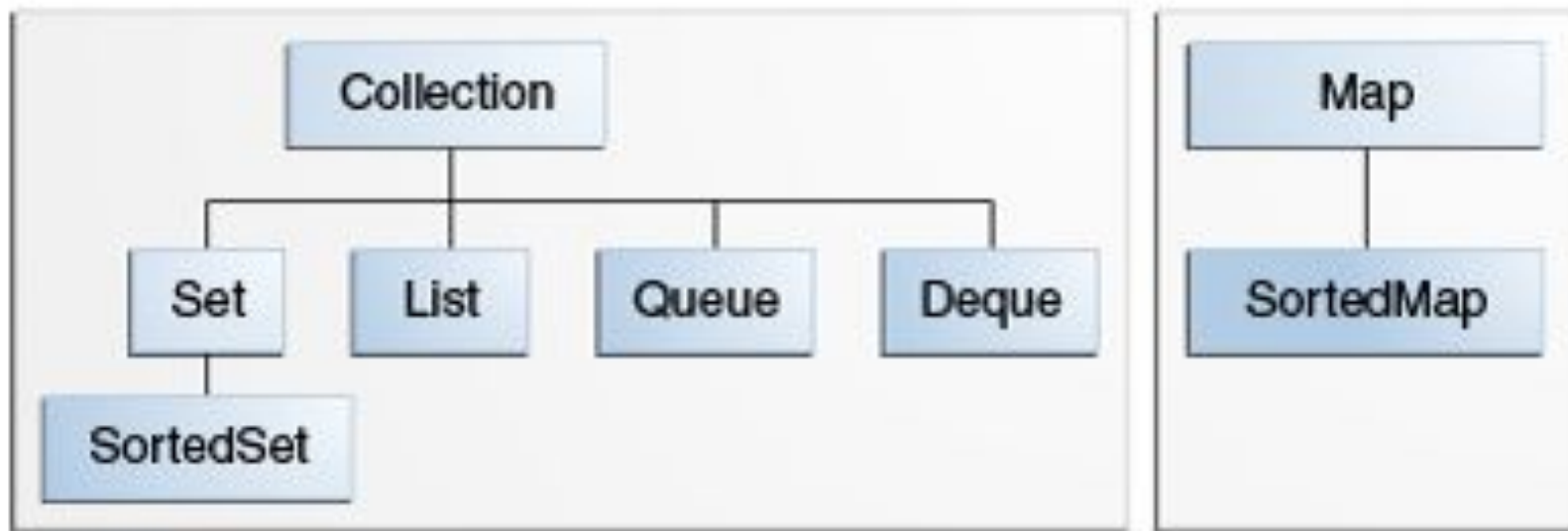
Collections e Iterator

Prof.: Leonardo Tórtoro Pereira

leonardop@usp.br

Collections [1]

- As interfaces *Collection* encapsulam diferentes tipos de coleções
- Elas permitem manipular coleções independentemente de suas representações
- Existe uma hierarquia formada pelas interfaces mostrada a seguir



Fonte: [1]

Collections [1]

- Um *Set* é um tipo especial de *Collection*, um *SortedSet* é um tipo especial de *Set* e assim por diante
- É importante notar que um mapa não é uma *Collection* propriamente dita
- Todas as interfaces de coleção são genéricas!
 - ◆ Vamos ver melhor sobre como montar algo genérico na próxima aula

Collections [1]

→ `public interface Collection<E>...`

- ◆ Essa sintaxe “<E>” indica que a interface é genérica
- ◆ Quando declaramos uma instância de coleção, devemos especificar o tipo de objeto contido nela
- ◆ Assim o compilador consegue verificar a tempo de compilação se os objetos colocados na coleção estão corretos

Collections [1]

- As coleções são limitadas.
 - ◆ Não existem variações de interface para todo tipo de coleção possível
- Mas isso é contornado ao deixar a implementação suportar ou não determinadas operações
 - ◆ Então, se o programador tentar executar uma operação não suportada, é levantada uma exceção

Collection

Collections [1]

- A raiz da hierarquia
- Representa um grupo de objetos chamados de *elements*
- A *Collection* é usada para passar coleções em chamadas e manipulá-las com o máximo de generalidade possível
- Alguns tipos de coleção permitem duplicar elementos
 - ◆ Outras não
- Algumas são ordenadas
 - ◆ Outras não

Set

Collections [1]

- Coleção que não pode conter elementos duplicados
- Modela a abstração matemática de conjuntos
- Usada para representar conjuntos como
 - ◆ Cartas de baralho
 - ◆ Cronograma de um estudante
 - ◆ Processos em uma máquina

List

Collections [1]

- Coleção ordenada (sequência)
- Listas podem conter elementos duplicados
- Geralmente o usuário tem controle preciso sobre onde cada elemento é inserido na lista
- Pode acessar os elementos por um índice inteiro. Ou seja, a posição do elemento
- Bem similar ao *Vector* de C++

Queue

Collections [1]

- Coleção usada para guardar múltiplos elementos antes de processar
- Provê operações extras de inserção, extração e inspeção
- Geralmente ordenam elementos em FIFO (*first-in, first-out*)
 - ◆ Uma exceção é a lista de prioridade
 - Ordena elementos a partir de um comparador fornecido ou por uma ordem natural de elementos

Collections [1]

- A cabeça da fila é o elemento removido ao chamar *remove()* ou *poll()*
- Todos os novos elementos são inseridos na cauda da fila
- Outros tipos de filas podem usar outras regras de colocação
- Toda implementação de fila precisa especificar as propriedades de ordenação

Deque

(Fila Duplamente Terminada)

Collections [1]

- Coleção usada para conter múltiplos elementos antes de processar
- Provê operações extras de inserção, extração e inspeção
- Podem ser usadas tanto como FIFO ou LIFO (*last-in, first-out*)
- Todos os novos elementos podem ser inseridos, recuperados ou removidos de ambos os fins

Map

Collections [1]

- Objeto que mapeia chaves a valores
- Não pode conter chaves duplicadas
 - ◆ Cada chave pode mapear apenas um valor
- Similar a uma *Hashtable*

Sorted Set

Collections [1]

- Um *Set* que mantém os elementos em ordem crescente
- Várias operações extras são providas para aproveitar da ordenação
- Usados para conjuntos naturalmente ordenados, como listas de palavras

Sorted Map

Collections [1]

- Um mapa que mantém seu mapeamento em ordem crescente de chaves
- Usados para coleções naturalmente ordenadas de pares chave/valor
 - ◆ Dicionários
 - ◆ Lista telefônica

Exemplos

Collections [1]

- Vamos ver alguns exemplos de uso das coleções
 - ◆ Agradecimentos ao material do prof. João do E. S. Batista Neto
 - <http://www.lcad.icmc.usp.br/~jbatista/sce537/src/collec/UsingArrays.java>
 - <http://www.lcad.icmc.usp.br/~jbatista/sce537/src/collec/CollectionTest.java>
 - <http://www.lcad.icmc.usp.br/~jbatista/sce537/src/collec/ListTest.java>
 - <http://www.lcad.icmc.usp.br/~jbatista/sce537/src/collec/UsingToArray.java>
 - <http://www.lcad.icmc.usp.br/~jbatista/sce537/src/collec/SetTest.java>
 - <http://www.lcad.icmc.usp.br/~jbatista/sce537/src/collec/WordTypeCount.java>
 - <http://www.lcad.icmc.usp.br/~jbatista/sce537/src/collec/MapTest.java>

Referências

1. <https://docs.oracle.com/javase/tutorial/collections/interfaces/index.html>
2. <http://www.lcad.icmc.usp.br/~jbatista/sce537/>