

Orientada a Objetos

List e ArrayList

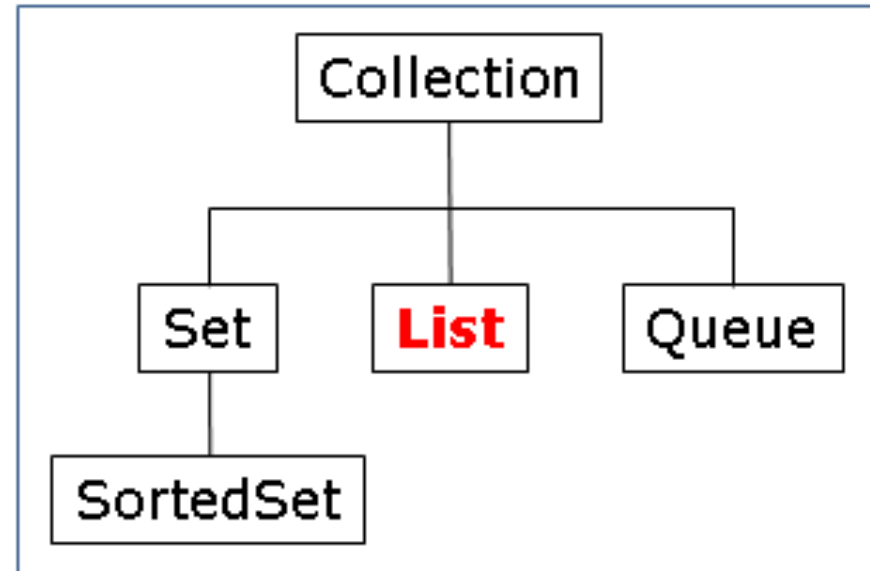
Profa.: Márcia Sampaio Lima

EST - UEA

List, ArrayList

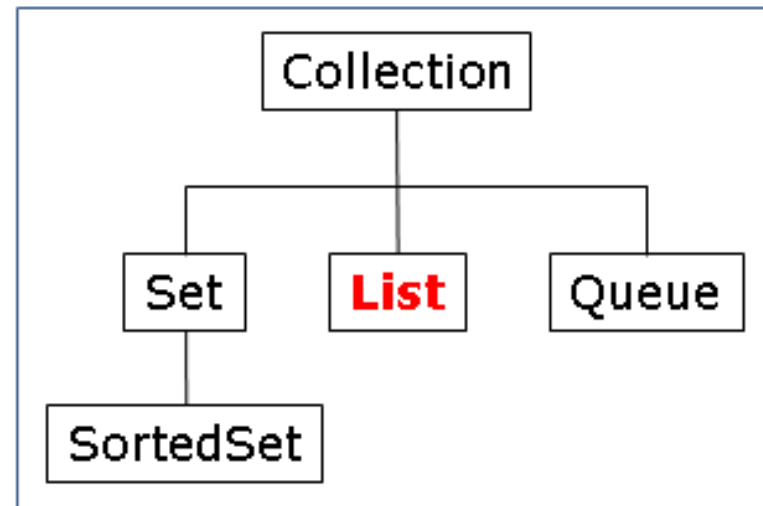
- O Java, por padrão, possui uma série de recursos prontos (APIs) para que possamos tratar de estrutura de dados, também chamados de coleções (collections)..

Collection



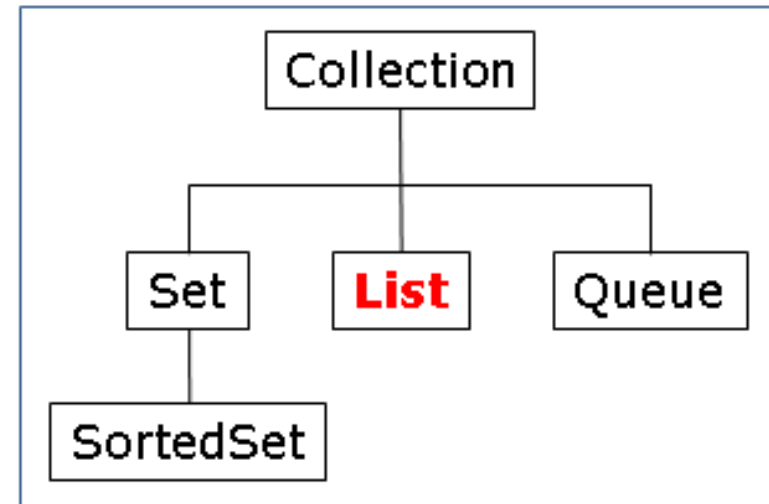
Set - Uma coleção de elementos que modela a abstração matemática para conjuntos. Não mantém indexação e nem contagem dos elementos pertencentes. Cada elemento pertence ou não pertence ao conjunto (não há elementos repetidos). Podem ser mantidos ordenados (SortedSet) ou não.

Collection



- **Lista (List):** Uma coleção indexada de objetos. Como nos vetores, o índice do primeiro elemento é zero. List fornece métodos para manipular elementos baseado na sua posição numérica na lista, remover determinado elemento, procurar as ocorrências de um dado.

Collection



- **Fila (Queue):** Uma coleção utilizada para manter uma "fila" de elementos. Existe uma ordem linear para as filas que é a "ordem de chegada". Nas filas, os itens deverão ser processados de acordo com a ordem "PRIMEIRO-QUE-CHEGA, PRIMEIRO-ATENDIDO". Por esta razão as filas são chamadas de Listas FIFO, termo formado a partir de "First-In, First-Out".

List, ArrayList

■ ArrayList

- ❑ ArrayList é uma classe para coleções.
 - ❑ Uma classe genérica (generic classes).
 - ❑ Coleções de qualquer tipo de 'coisa' ou dado, não somente de tipos primitivos.
 - ❑ Podemos criar objetos e agrupá-los através de ArrayList, gerando uma lista de objetos.
 - ❑ O List é uma interface e o ArrayList é a classe que a implementa.
-

List, ArrayList

- ArrayList

- Podemos criar objetos - através de uma classe - e agrupá-los através de ArrayList e realizar, nessa coleção, várias operações, como:

- adicionar e retirar elementos,
 - ordená-los,
 - procurar por um elemento específico,
 - apagar um elemento específico,
 - limpar o ArrayList dentre outras possibilidades.
-

Declaração

- **Importe:**

```
import java.util.ArrayList;
```

- **Sintaxe:**

```
ArrayList<Objeto> nomeArrayList = new ArrayList<Objeto>();
```

- `List<Objeto> nomeArrayList = new ArrayList<Objeto>();`

- **Exemplo: ArrayList de String**

```
ArrayList<String> bandas = new ArrayList<String> ();
```

ArrayList

- Adicionar uma String ao ArrayList:

```
bandas.add("Rush");
```

- Obter o primeiro elemento, o elemento '0', através do método 'get':

```
String nome = bandas.get(0);
```

- Adicionar o "Tiririca" na posição do "Rush", ou seja, na posição '0':

```
bandas.add(0, "Tiririca");
```

Declaração

- indexOf() retorna o índice em que ocorre "Rush"

```
bandas.indexOf("Rush");
```

- Tamanho que tem seu ArrayList: método size():

```
bandas.size();
```

- Remover o "Tiririca".

```
bandas.remove("Tiririca");
```

Declaração

- Limpar o ArrayList
`bandas.clear();`

List, ArrayList

```
List<Gato> lista_Cats = new ArrayList<Gato>();
```

```
Gato cat = new Gato("gato1",1,1);  
lista_Cats.add(cat);
```

```
lista_Cats.add(new Gato("gato2",2,2));  
lista_Cats.add(new Gato("gato3",3,3));  
lista_Cats.add(new Gato("gato4",4,4));
```

```
for(int i =0 ; i < lista_Cats.size(); i++){  
    cat = lista_Cats.get(i);  
    System.out.println(cat.nome);  
}
```

ArrayList

- Pesquisar / Estudar
 - Os métodos de um ArrayList
 - lista_Cats.**remove**(arg0);
 - lista_Cats.**set**(index, element);
 - lista_Cats.**clear**();
 - lista_Cats.**contains**(object);
-

List

- List
 - Vector
 - ArrayList
 - LinkedList

List

■ ArrayList:

- É implementado como um Array que é dimensionado dinamicamente.
 - Permite que elementos sejam acessados diretamente pelos métodos `get()` e `set()`, e adicionados através de `add()` e `remove()`.
-

List

- Vector:

- o Vector e o ArrayList são muito similares.
 - Vector ser sincronizado e o ArrayList não. Em uma aplicação thread-safe deve-se usar Vector.
 -
-

List

■ LinkedList:

- ❑ Implementa uma lista duplamente “linkada”.
 - ❑ A principal diferença entre o ArrayList é na performance entre os métodos add, remove, get e set.
 - ❑ Melhor performance nos métodos add e remove, do que os métodos add e remove do ArrayList.
 - ❑ Pior desempenho nos métodos get e set se comparado aos do ArrayList.
-