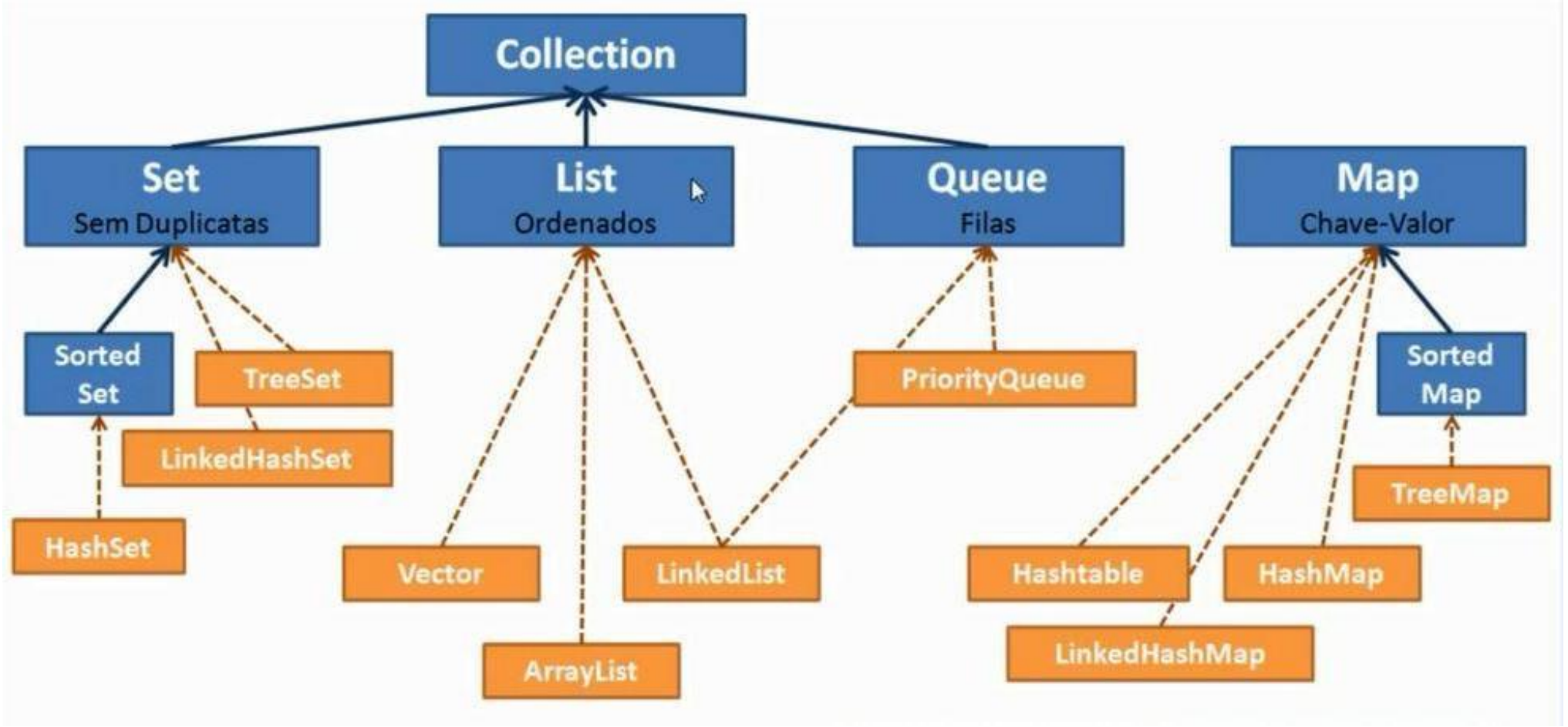
Curso Técnico em Desenvolvimento de Sistemas

UCR8 – Desenvolver e Organizar Interface Gráfica para Aplicações Desktop

Aula 09 –Coleções

Coleções

- Forma de se criar coleções de objetos com tamanho dinâmico; utilizado quando há necessidade de manipulação de dados constante.



Hierarquia das Coleções de Objetos em Java

ArrayList:

- Tipo de vetor cujo tamanho pode ser modificado ao longo do programa;
- A busca de um elemento em um *ArrayList* é rápida, mas inserções e exclusões são lentas,
- Permite elementos duplicados; recebe conjunto de *objects*;

Criação de um ArrayList:

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

```
ArrayList<Integer> idade = new ArrayList<Integer>(initialCapacity: 10);
```

```
// Uso dos tipos primitivos:
// Byte, Short, Integer, Long, Float, Double, Character.
```

ArrayList -> Principais Métodos:

```
// PRINCIPAIS MÉTODOS
nome.add(e: "Celso");  // Adiciona um item ao proximo index disponivel
nome.get(index: 0);    // Obtém o elemento do índice: "Celso"
nome.set(index: 0,element:"Luis"); // Substitui elemento index 0
nome.remove(index: 0);  // Elimina um elemento do Array
nome.clear();          // Elimina todos elementos de um Array
nome.size();           // Retorna o tamanho do Array
nome.isEmpty();        // Retorna true para Array populada
nome.indexOf(o: "Fulano"); // Retorna index objeto, se estiver na lista
```

ArrayList -> Exercício Básico.

- Crie uma classe (que possui método *main*) com um uma função que receba do usuário informações de nomes de alunos. Os nomes dos alunos informados deverão ser armazenados em um *ArrayList.* Após a finalização do programa, exiba os dados de todos os estudantes via *SystemOut*, de forma ordenada (ordenação alfabética) formatando cada objeto separado por linhas. *Pesquise sobre o método: Collection.Sort();*

Set:

- Interface Set (implementação padrão HashSet) permite a criação de conjuntos dinâmicos (mesmas aplicações que ArrayList, porém não impõe ordem aos elementos, nem permite dados duplicados, dados duplicados serão excluídos);

Criação de um HashSet:

```
HashSet<String> item = new HashSet<String>();
```

HashSet -> Principais Métodos:

```java
HashSet<String> item = new HashSet<String>();
item.add(e: "elemento");       // Adiciona elemento
item.remove(o: "elemento");    // Remove um elemento
item.clear();                  // Limpa a lista
item.contains(o: "elemento"); // 'True' se o elemento está no conjunto
item.isEmpty();                // 'True' se a lista está vazio
item.size();                   // Tamanho da lista
```

```java
HashSet<String> nomeCachorro = new HashSet<String>();
nomeCachorro.add(e: "Bidu");
nomeCachorro.add(e: "Rino");
nomeCachorro.add(e: "Duque");
nomeCachorro.add(e: "Lobo");
nomeCachorro.add(e: "Bidu");
nomeCachorro.add(e: "Lobo");

for(String nome : nomeCachorro){
    System.out.println(x: nome);
}
```

```
run:
Lobo
Duque
Bidu
Rino
BUILD SUCCESSFUL
```

Map:

- Interface Map (implementação padrão Map) permite a criação de dicionários dinâmicos;

  associa um objeto chave a um objeto valor (key -> value);

Criação de um Map:

```
HashMap<String, String> itens = new HashMap<String, String>();
```

```
itens.put(key:"chave", value: "valor");     // Adiciona item
itens.get(key:"chave");                     // Retorna valor index pela chave
itens.getOrDefault(key:"chave", defaultValue: "valor");
                                            // Retorna valor index pela chave
                                            // ou valor default informado
itens.keySet();                             // Retorna todas chaves
itens.remove(key:"chave");                  // Revome valor indexado pela chave
itens.clear();                              // Limpa a lista
itens.isEmpty();                            // True, se lista vazia
itens.size();                               // Retorna tamanho
```

HashMap -> Principais Métodos:

```java
HashMap<String, String> dddMunicipio = new HashMap<String, String>();

dddMunicipio.put(key:"51", value: "Porto Alegre");
dddMunicipio.put(key:"54", value: "Caxias");
dddMunicipio.put(key:"55", value: "Santa Maria");
dddMunicipio.put(key:"21", value: "Rio de Janeiro");
dddMunicipio.put(key:"11", value: "Sao Paulo");

for(String ddd : dddMunicipio.keySet()){
    System.out.println("O DDD " + ddd + " pertecence a cidade de "
            + dddMunicipio.getOrDefault(key:ddd, defaultValue: ddd));
}
```

```
run:
O DDD 55 pertecence a cidade de: Santa Maria
O DDD 11 pertecence a cidade de: Sao Paulo
O DDD 51 pertecence a cidade de: Porto Alegre
O DDD 54 pertecence a cidade de: Caxias
O DDD 21 pertecence a cidade de: Rio de Janeiro
BUILD SUCCESSFUL (total time: 0 seconds)
```