

# Fernando Anselmo

fernandoanselmo@yahoo.com.br

Sun Certified Programmer for  
the Java 2 Platform 1.4

Com base no livro:

- Complete Java 2 Certification – Study Guide  
Simon Roberts, Philip Heller e Michael Ernest

# Dia 9

## Object e Collections

- Indentificar e utilizar os métodos da classe Object
- Identificar corretamente as coleções;
- Entender o funcionamento de uma árvore e de uma hash;
- Verificar a classe Arrays; e
- Conhecer as interfaces List, Set e Map.

# Métodos da Object

- `public boolean equals(Object obj)`
  - Verifica o conteúdo de dois objetos
- `protected void finalize()`
  - Chamado quando o Garbage Collector for finalizar o objeto.
- `public int hashCode()`
  - Retorna um valor inteiro “hashcode” para o objeto.
- `final void notify()`
  - Para finalizar um estado “waiting” de uma “thread”.
- `final void notifyAll()`
  - Para finalizar um estado “waiting” de todas as “thread”.
- `final void wait()`
  - Causa um estado “waiting” na thread que chamar.
- `public String toString()`
  - Retorna uma “representação em texto” do objeto.

# Método hashCode

- Não pode ser:

$$\begin{aligned} \text{ALEX} &= \text{A (01)} + \text{L (12)} + \text{E (05)} + \text{X (24)} = 42 \\ \text{DIRK} &= \text{D (04)} + \text{I (09)} + \text{R (18)} + \text{K (11)} = 42 \end{aligned}$$

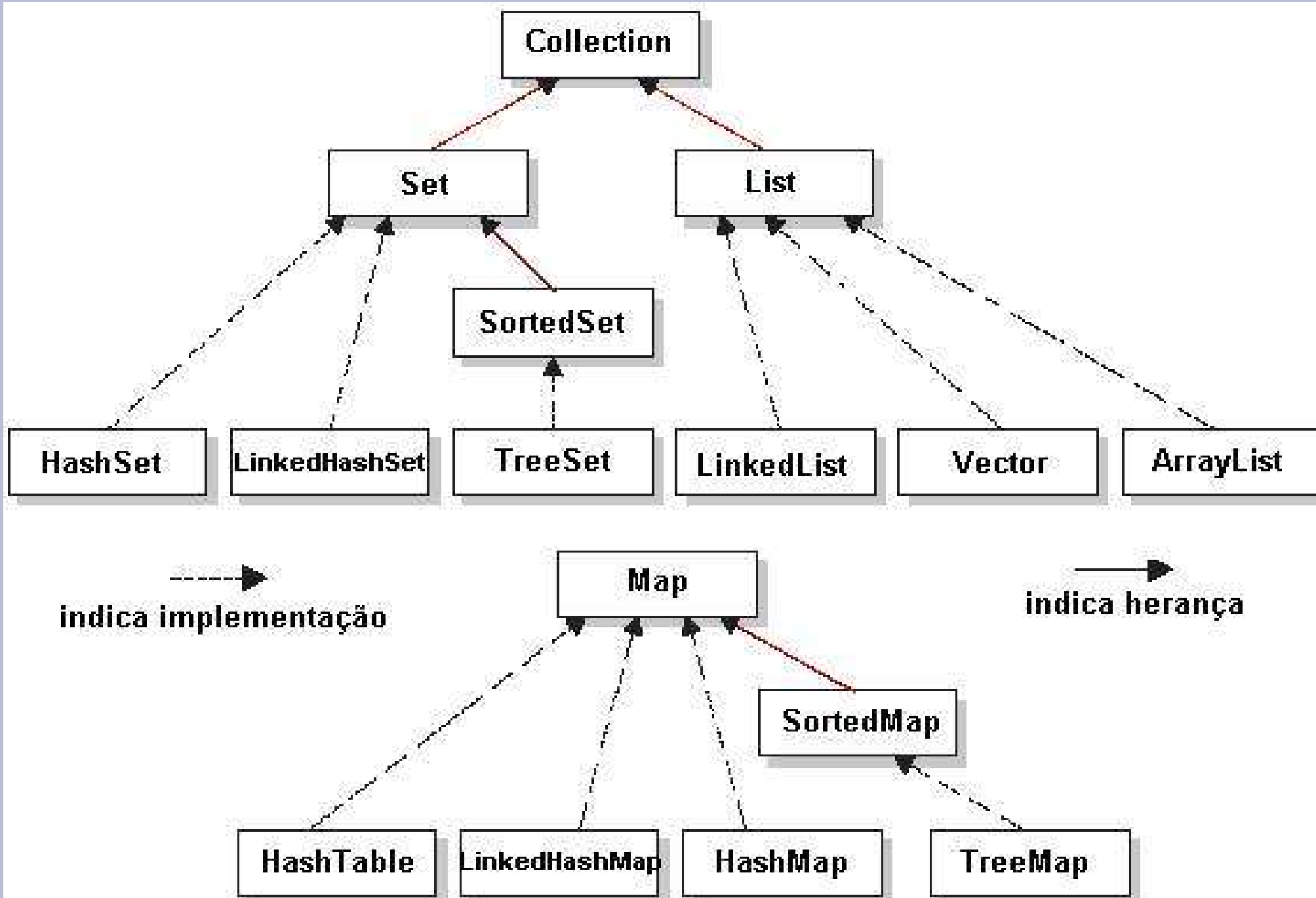
- É um cálculo matemático realizado para que dois objetos sejam únicos
- Entre o método hashCode e o equals:

<b>Condição</b>	<b>Requer</b>	<b>Não Requer</b>
<code>equals = true</code>	<code>hashCode =</code>	-
<code>hashCode =</code>	-	<code>equals = true</code>
<code>equals = false</code>	-	-
<code>hashCode !=</code>	<code>equals = false</code>	-

# Coleções

- Coleções são objetos que possuem a capacidade de armazenar outros objetos.
- Permite o acesso a estrutura de dados pré-empacotados e a algoritmos para manipular estas estruturas.
- As coleções fornecem interfaces que definem as operações a serem realizadas, encontram-se no pacote `java.util`, podendo ser implementadas das seguintes interfaces:
  - Set – Lista de coisas únicas
  - List – Lista de coisas
  - Map – Lista com chaves únicas

# Estrutura das Coleções



# Classe Collections

- Classe acessora que fornece diversos métodos para manipulação dos elementos das coleções:
  - sort – ordenar valores
  - shuffle – embaralha os elementos
  - reverse – invete a ordem dos elementos
  - fill – colocar um valor em todas as posições
  - binarySearch – pesquisar um array ordenado
  - copy, max e min – copia, maior e menor elemento.
- Uso: `Collections.sort(lst);`

# Classe Arrays

- Classe acessora que fornece diversos métodos de alto-nível para manipulação de arrays
  - `binarySearch` – pesquisar um array ordenado
  - `equals` – comparação de dois arrays
  - `fill` – colocar um valor em todas as posições
  - `sort` – ordenar valores
- Uso: `Arrays.metodo(objeto);`

# Interface Set

- É uma coleção de elementos únicos e não duplicados.
- Implementada pela Classe HashSet que armazena os elementos em uma tabela hash e a Classe TreeSet onde os elementos são armazenados em uma árvore.

# Interface List

- É uma coleção ordenada de objetos que pode conter elementos duplicados
- Ela pode ser implementada através das classes `ArrayList`, `LinkedList` e `Vector`.

- **Uso:**

```
String nomes[] = { "Fernando", "João", "Maria" };  
ArrayList lst = new ArrayList();  
for (int i = 0; i < nomes.length; i++)  
    lst.add(nomes[i]);
```

# Interface Map

- É uma coleção que associa chaves a objetos e não pode conter chaves duplicadas.
- Implementada pela Classe HashMap que armazena os elementos em uma tabela hash e a Classe TreeMap onde os elementos são armazenados em uma árvore.

# Time for Exercices

