Collections

Prof. Edwar Saliba Júnior
Fevereiro de 2011

Introdução

- Java fornece implementação de Estruturas de Dados recorrentemente utilizadas. Exemplos:
  - Fila,
  - Pilha,
  - Lista e
  - *Hash*;
- Estas estruturas são denominadas *collections* (coleções);
- O programador as utiliza sem se preocupar com a forma como foram implementadas.

Collections

- Em Java, uma coleção:
  - é um objeto,
  - pode armazenar referências a outros objetos,
  - possui um *iterator* (mecanismo que provê interação com o objeto);
- Exemplos de coleções:
  - *HashSet*,
  - *HashMap* e
  - *ArrayList*.

Arranjos

Array

- Classe java.util.Arrays;
- Este pacote fornece métodos estáticos (*static*) para manipulação de arranjos;
- Dentre os diversos métodos existentes podemos citar:
  - ***sort*** – Ordena os elementos de um arranjo,
  - ***copyOf*** – Faz uma cópia de um arranjo,
  - ***equals*** – compara dois arranjos.

Exemplo de *Arrays*

- Exemplo (.pdf)

Listas

List

- **java.util.List** interface de Java que define listas;
- A interface *List* é implementada por:
  - **Vector (em desuso!)**: uma implementação para arranjos em Java 1.0 antes das coleções serem implementadas;
  - **ArrayList**: realiza basicamente as mesmas operações que Vector, porém, com melhor desempenho;
  - **LinkedList**: implementação para listas encadeadas.

Exemplo de *ArrayList*

- Exemplo (.pdf)

Pilha

Stack

- Nesta coleção, o primeiro elemento que entra é o último a sair;
- Classe java.util.Stack;
- Principais métodos:
  - **peek**: verifica o elemento do topo da pilha;
  - **pop**: retira o elemento do topo da pilha;
  - **push**: coloca um elemento no topo da pilha;
  - **empty**: verifica se a pilha está vazia.

Exemplo de *Stack*

- Exemplo (.pdf)

Conjuntos

HashSet

- Um conjunto não pode conter elementos duplicados;
- java.util.Set é a interface que define conjuntos em Java;
- A classe java.util.HashSet é uma classe que implementa a interface *Set*.

HashSet

- Principais métodos:
  - **add**: inclui um elemento no conjunto se ele ainda não estiver lá;
  - **contains**: verifica se o conjunto contém um elemento;
  - **isEmpty**: verifica se o conjunto está vazio.

Exemplo de *HashSet*

- Exemplo (.pdf)

Mapas

HashMap

- Mapas (mapeamento):
  - Associam chaves a valores;
  - Chaves não podem ser duplicadas;

- Diferença entre mapa e conjuntos:
  - Conjuntos possuem somente valores;
  - Mapas possuem chaves que são associadas a valores.

HashMap

- Classe java.util.HashMap;
- Principais métodos:
  - **containsKey(Object key)**: verifica se o mapa possui a chave informada;
  - **containsValue(Object value)**: verifica se há no mapa alguma chave para o valor informado;
  - **get(Object key)**: retorna o valor para o qual a chave informada está mapeada;
  - **put(Object key, Object value)**: insere um mapeamento da chave para o valor no mapa;
  - **keySet()**: retorna um conjunto de chaves do mapa;
  - **values()**: retorna uma coleção com os valores presentes no mapa.

Exemplo de *HashMap*

- Exemplo (.pdf)

Bibliografia

- DEITEL, H. M.; DEITEL, P. J. **Java Como Programar**; tradução Edson Furmankiewicz; revisão técnica Fábio Lucchini. 6a. ed., São Paulo: Pearson, 2005.
- FERREIRA, Kecia Aline Marques. *Slides* da disciplina de Programação de Computadores II. CEFET-MG, 2009.