

AfroDev

Java Básico - Aula 6 - Parte 2

Collections Framework - Map

- Mapa é um objeto que mapeia uma chave a um único valor (um valor pode ser um objeto e portanto pode ser uma lista)
- Não pode conter chaves duplicadas
- Implementações: HashMap, TreeMap e LinkedHashMap
- Análogo aos equivalentes em Set (HashSet, TreeSet e LinkedTreeSet)

Collections Framework - SortedMap

- As entradas são ordenadas na ordem crescente de acordo com sua chave
- Será que você precisa de um mapa ordenado?

Collections Framework - Map - Exemplo

- Vamos analisar juntos o código em: [Freq.java](#)

Collections Framework - Map - Exercício

- Seu amigo lhe disse que para calcular a função log de um número você deve usar a seguinte API Java: `Math.log(42D)`;
- Porém ele também lhe disse que essa função é lenta
- Você, muito sagaz, pensou num cache...
- Cache é um espaço onde coisas já processadas ficam “prontas” apenas para serem consultadas
- Escreva um programa que invoque o método `Math.log` somente quando o argumento NÃO estiver no cache

Collections Framework - Ordenação

- Precisamos da interface Comparable para comparar o objeto corrente com outro objeto
- Segredo contido no método compareTo. Vejamos a documentação: [Comparable JavaDoc](#)
- Vamos analisar o exemplo [“Name”](#)
- Podemos usar um [Comparator](#) (objeto que encapsula a ordem) e comparar dois objetos

Collections Framework - Ordenação

Classes Implementing Comparable

Class	Natural Ordering
Byte	Signed numerical
Character	Unsigned numerical
Long	Signed numerical
Integer	Signed numerical
Short	Signed numerical
Double	Signed numerical
Float	Signed numerical
BigInteger	Signed numerical
BigDecimal	Signed numerical
Boolean	Boolean.FALSE < Boolean.TRUE
File	System-dependent lexicographic on path name
String	Lexicographic
Date	Chronological
CollationKey	Locale-specific lexicographic

Collections Framework - Ordenação - Exercício

- Crie um objeto complexo “Pessoa” com nome (String) e idade (int)
- Crie uma lista com várias pessoas
- Ordene a lista em ordem crescente com o seguinte critério: nome, idade
- Isso significa que se na lista tiverem duas pessoas com o mesmo nome, o próximo critério de ordenação será a idade

Revisão

- Vimos que mapas são estruturas que armazenam dados no formato chave -> valor e um uso interessante de mapas seria fazer um cache
- Vimos que a ordenação natural de elementos vem do objeto que implementa Comparable OU se você implementar um Comparator que compare dois objetos