Bài 7.8: Lớp Hashtable trong thư viện Java

- ✓ Tổng quan
- ✓ Các phương thức và mô tả
- ✓ Ví dụ minh họa
- ✓ Bài tập thực hành

Tổng quan

- Hashtable là lớp triển khai một bảng băm lưu trữ dữ liệu dạng key-value.
- Bất kì đối tượng nào khác null đều có thể được sử dụng làm key hoặc value trong bảng băm.
- Để có thể được sử dụng làm key trong bảng băm, đối tượng phải triển khai phương thức equals() và hashCode().
- 2 yếu tố ảnh hưởng tới hiệu năng của bảng băm: load factor và khả năng lưu trữ(capacity) ban đầu.
- Giá trị capacity là số lượng vùng(ô) nhớ của bảng băm. Hay còn gọi lượng phần tử tối đa có thể lưu trữ.
- Nếu capacity ban đầu quá lớn thì sẽ dư thừa bộ nhớ không sử dụng. Nếu quá thấp thì phải băm lại nhiều lần trong quá trình sử dụng.

Tổng quan

- Bảng băm thể hiện tính mở: khi xảy ra xung đột băm, một vùng chứa của bảng băm sẽ chứa nhiều entry khác nhau liên kết với nhau theo kiểu danh sách liên kết.
- Load factor là giá trị thể hiện khả năng chứa các phần tử trước khi cần phải rehash.
- Thông thường giá trị của load factor được fix là 0.75. Đây là giá trị tốt nhất trong tương quan giữa chi phí thời gian và lượng bộ nhớ được sử dụng.
- Hashtable là một lớp đồng bộ hóa. Nếu không cần tính đồng bộ, có thể sử dụng lớp HashMap thay thế.
- Trường hợp cần tính đồng bộ cao hơn ta sử dụng lớp ConcurentHashMap.

Các phương thức và mô tả

Phương thức	Mô tả
Hashtable()	Tạo một bảng băm rỗng với khả năng khởi tạo là 11, load factor là 0.75
Hashtable(int capacity)	Tạo một bảng băm rỗng với khả năng khởi tạo là capacity, load factor là 0.75
Hashtable(int capacity, float load)	Tạo một bảng băm rỗng với khả năng khởi tạo là capacity, load factor là load
Hashtable(Map<? extends K, ? extends V> map)	Tạo một bảng băm mới với các phần tử cho trước trong map.
void clear()	Xóa toàn bộ các phần tử hiện có trong bảng băm.
Object clone()	Tạo bản sao của đối tượng hiện thời.
V comput(K key, BiFunction<? super K, ? extends V> remappingFunction)	Thử tính toán ánh xạ cho key và value hiện tại liên kết với key.
boolean contains(Object value)	Kiểm tra xem một key nào đó có ánh xạ tới một giá trị cụ thể nào trong bảng băm không.
boolean containsKey(Object key)	Kiểm tra xem một key cụ thể có tồn tại trong bảng băm không.
boolean containsValue(Object value)	Trả về true nếu bảng băm có 1 hoặc nhiều key liên kết tới value này.
Enumeration<V> elements()	Trả về danh sách các giá trị trong bảng băm hiện thời ở dạng Enumeration.
Set<Map.Entry<K, V>> entrySet()	Trả về tập các ánh xạ có trong bảng băm hiện tại.

Các phương thức và mô tả

boolean equals(Object o)	So sánh đối tượng o với Map hiện tại xem chúng có tương đương không.
void forEeach(BiConsumer<? super K, ? super V> action)	Thực hiện hành động cho trước với từng entry của bảng băm tới khi duyệt hết các entry hoặc tới khi văng ngoại lệ.
V get(Object key)	Trả về value được liên kết với key được chỉ định hoặc trả về null nếu không tồn tại cặp key-value.
V getOrDefault(Object key, V defaultValue)	Trả về value được liên kết với key cho trước hoặc trả về defaultValue nếu không tìm thấy cặp key-value nào.
int hashCode()	Trả về mã băm của bảng băm hiện thời.
boolean isEmpty()	Kiểm tra xem bảng băm hiện tại có rỗng không.
Enumeration<K> keys()	Trả về danh sách key có trong bảng băm hiện tại.
Set<K> keySet()	Trả về tập các key chứa trong bảng băm hiện thời dưới dạng đối tượng Set.
V merge(K key, V value, BiFuntion<? extend K, ? extends V> remappingFunc)	Nếu key chưa được liên kết với value hoặc key liên kết với null thì liên kết nó với giá trị value khác null.
V put(K key, V value)	Ánh xạ key với value được chỉ định và lưu vào trong bảng băm.
void putAll(Map<? extends K, ? extends V> t)	Copy tất cả các cặp key-value trong map t vào bảng băm hiện tại.

Các phương thức và mô tả

V putIfAbsent(K key, V value)	Nếu key cho trước mà chưa liên kết với value nào hoặc liên kết với value null thì liên kết nó với value cho trước và trả về null. Ngược lại, trả về value đã liên kết với key.
protected void rehash()	Tăng kích thước bảng băm và phân phối lại các phần tử trong bảng băm vào bảng mới.
V remove(key)	Xóa cặp key-value có key trùng với key cho trước.
boolean remove(Object key, Object value)	Xóa entry chứa cặp key-value như trong tham số.
V replace(K key, V value)	Thay thế value của entry có key trùng với key cho trước bằng giá trị value cho trước.
boolean replace(K key, V oldValue, V newValue)	Thay thế value cũ của key bằng newValue.
void replaceAll(BiFuntion<? super K, ? super V, ? extends V> function)	Thay thế mỗi entry bởi entry có được từ việc thực hiện function. Công việc được thực hiện cho tới khi tất cả các entry đã được xử lý hoặc tới khi xảy ra ngoại lệ.
int size()	Trả về số lượng phần tử hiện có trong bảng băm.
String toString()	Trả về một String đại diện cho bảng băm hiện tại dưới dạng tập các entry.
Collection<V> values()	Trả về một Collection chứa các giá trị value có trong bảng băm.

Nội dung tiếp theo

Tìm hiểu lớp HashMap