Found in the HashMap class:

abstract class HashIterator {  
 Node<K,V> next; // next entry to return  
 Node<K,V> current; // current entry  
 int expectedModCount; // for fast-fail  
 int index; // current slot  
  
 HashIterator() {  
 expectedModCount = modCount;  
 Node<K,V>[] t = table;  
 current = next = null;  
 index = 0;  
 if (t != null && size > 0) { // advance to first entry  
 do {} while (index < t.length && (next = t[index++]) == null);  
 }  
 }  
  
 public final boolean hasNext() {  
 return next != null;  
 }  
  
 final Node<K,V> nextNode() {  
 Node<K,V>[] t;  
 Node<K,V> e = next;  
 if (modCount != expectedModCount)  
 throw new ConcurrentModificationException();  
 if (e == null)  
 throw new NoSuchElementException();  
 if ((next = (current = e).next) == null && (t = table) != null) {  
 do {} while (index < t.length && (next = t[index++]) == null);  
 }  
 return e;  
 }  
  
 public final void remove() {  
 Node<K,V> p = current;  
 if (p == null)  
 throw new IllegalStateException();  
 if (modCount != expectedModCount)  
 throw new ConcurrentModificationException();  
 current = null;  
 removeNode(p.hash, p.key, null, false, false);  
 expectedModCount = modCount;  
 }  
}  
  
final class KeyIterator extends HashIterator   
 implements Iterator<K> {  
 public final K next() {   
 return nextNode().key;   
 }  
}  
  
final class ValueIterator extends HashIterator   
 implements Iterator<V> {  
 public final V next() {   
 return nextNode().value;   
 }  
}  
  
final class EntryIterator extends HashIterator   
 implements Iterator<Map.Entry<K,V>> {  
 public final Map.Entry<K,V> next() {   
 return nextNode();   
 }  
}