

## Bài 7.6: Lấy danh sách key, value

- 
- ✓ Lấy danh sách keys
  - ✓ Lấy danh sách values
  - ✓ Lấy danh sách các cặp key-value
  - ✓ Hiển thị các danh sách ra màn hình
  - ✓ Ví dụ minh họa
  - ✓ Bài tập thực hành

# Lấy danh sách các key

## ➤ Mã giả:

```
// lấy ra tất cả các key hiện có trong bảng băm
keys() {
    if (count > 0) { // nếu số phần tử lớn hơn 0
        keyList = new ArrayList<>(); // tạo danh sách chứa key
        for (e : table) { // duyệt tất cả các ô trong bảng băm
            var current = e; // lấy node đầu tiên tại vị trí đang xét
            while (current != null) { // tìm tất cả các node tại vị trí đang xét
                keyList.add((K) current.getKey()); // thêm key vào danh sách
                current = current.next; // chuyển đến node kế tiếp
            }
        }
        return keyList; // trả về danh sách các key
    }
    return null; // trả về danh sách rỗng
}
```

# Lấy danh sách các key

➤ Mã thật:

```
public List<K> keys() {  
    if (count > 0) {  
        List<K> keyList = new ArrayList<>();  
        for (var e : table) {  
            var current : HashTable<...>.Entry<...> = e;  
            while (current != null) {  
                keyList.add((K) current.getKey());  
                current = current.next;  
            }  
        }  
        return keyList;  
    }  
    return null;  
}
```

# Lấy danh sách values

## ➤ Mã giả:

```
// lấy ra tất cả các value hiện có trong bảng băm
values() {
    if (count > 0) { // nếu số phần tử lớn hơn 0
        valueList = new ArrayList<>(); // tạo danh sách chứa value
        for (e : table) { // duyệt tất cả các ô trong bảng băm
            var current = e; // lấy node đầu tiên tại vị trí đang xét
            while (current != null) { // tìm tất cả các node tại vị trí đang xét
                valueList.add((V) current.getValue()); // thêm value vào danh sách
                current = current.next; // chuyển đến node kế tiếp
            }
        }
        return valueList; // trả về danh sách các value
    }
    return null; // trả về danh sách rỗng
}
```

# Lấy danh sách values

➤ Mã thật:

```
public List<V> values() {
    if (count > 0) {
        List<V> valueList = new ArrayList<>();
        for (var e : table) {
            var current : HashTable<...>.Entry<...> = e;
            while (current != null) {
                valueList.add((V) current.getValue());
                current = current.next;
            }
        }
        return valueList;
    }
    return null;
}
```

# Lấy danh sách cặp key-value

➤ Mã giả:

```
// lấy ra tất cả các cặp key-value hiện có trong bảng băm
elements() {
    if (count > 0) { // nếu số phần tử lớn hơn 0
        entries = new ArrayList<>(); // tạo danh sách chứa các cặp key-value
        for (e : table) { // duyệt tất cả các ô trong bảng băm
            var current = e; // lấy node đầu tiên tại vị trí đang xét
            while (current != null) { // tìm tất cả các node tại vị trí đang xét
                entries.add((Entry<K, V>) current); // thêm key-value vào danh sách
                current = current.next; // chuyển đến node kế tiếp
            }
        }
        return entries; // trả về danh sách các cặp key-value
    }
    return null; // trả về danh sách rỗng
}
```



# Lấy danh sách cặp key-value

➤ Mã thuật:

```
public List<Entry<K, V>> elements() {
    if (count > 0) {
        List<Entry<K, V>> entries = new ArrayList<>();
        for (var e : table) {
            var current : HashTable<...>.Entry<...> = e;
            while (current != null) {
                entries.add((Entry<K, V>) current);
                current = current.next;
            }
        }
        return entries;
    }
    return null;
}
```

# Hiển thị danh sách ra màn hình

## ➤ Mã giả:

```
// hiển thị các cặp key-value ra màn hình
showElements() {
    entries = elements(); // lấy danh sách cặp key-value
    print("["); // hiển thị dấu mở móc vuông
    for (i = 0; i < entries.size(); i++) { // duyệt toàn bộ danh sách chứa các entry
        e = entries.get(i); // lấy phần tử entry chứa cặp key-value tại vị trí i
        delimiter = ", "; // dấu phân tách các phần tử với nhau
        printf("(%s-%s)", e.getKey(), e.getValue()); // in ra cặp (key, value)
        if (i != entries.size() - 1) { // nếu chưa phải phần tử cuối
            System.out.print(delimiter); // hiển thị dấu phân tách
        }
    }
    System.out.println("]"); // hiển thị dấu đóng móc vuông
}
```

# Hiển thị danh sách ra màn hình

## ➤ Mã thuật:

```
/*
 * Hiển thị danh sách các cặp key-value ra màn hình
 */
public void showElements() {
    var entries : List<HashTable<...>.Entry<...>> = elements();
    System.out.print("[");
    for (int i = 0; i < entries.size(); i++) {
        var e : HashTable<...>.Entry<...> = entries.get(i);
        var delimiter = ", " // dấu phân tách các phần tử với nhau
        System.out.printf("(%s-%s)", e.getKey(), e.getValue());
        if (i != entries.size() - 1) {
            System.out.print(delimiter);
        }
    }
    System.out.println("]");
}
```

# Nội dung tiếp theo

## Thuật toán sắp xếp bucket sort