1. Lý thuyết HashMap trong Java
2. Các thao tác cơ bản với HashMap

- HashMap là một cấu trúc dữ liệu dạng key-value trong Java, thuộc gói java.util. Dưới đây là các thao tác quan trọng:

1. Duyệt phần tử trong HashMap:

- Có nhiều cách để duyệt HashMap:

+ Sử dụng entrySet(): Duyệt qua tập hợp các cặp key-value (Map.Entry)

+ Sử dụng keySet(): Duyệt qua tập hợp các key, sau đó lấy value qua get(key)

+ Sử dụng values(): Duyệt qua tập hợp các value (bỏ qua key)

+ Sử dụng Iterator: Duyệt thủ công, hữu ích khi cần xóa phần tử trong khi duyệt

+ Sử dụng forEach: Cú pháp ngắn gọn, sử dụng lambda để duyệt.

1. Các thao tác khác:

- containsKey(key): Kiểm tra xem key có tồn tại không

- containsValue(value): Kiểm tra xem value có tồn tại không

- size(): Trả về số lượng cặp key-value

- isEmpty(): Kiểm tra HashMap có rỗng không

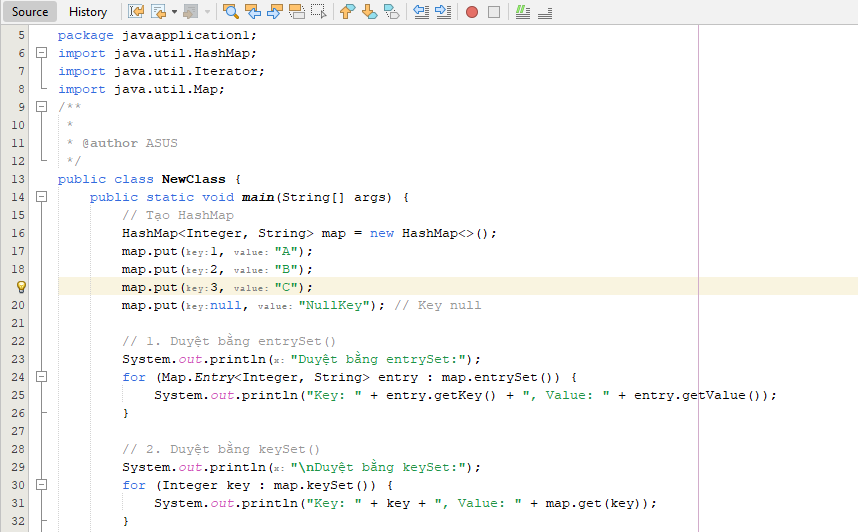
- get(key): Lấy value theo key

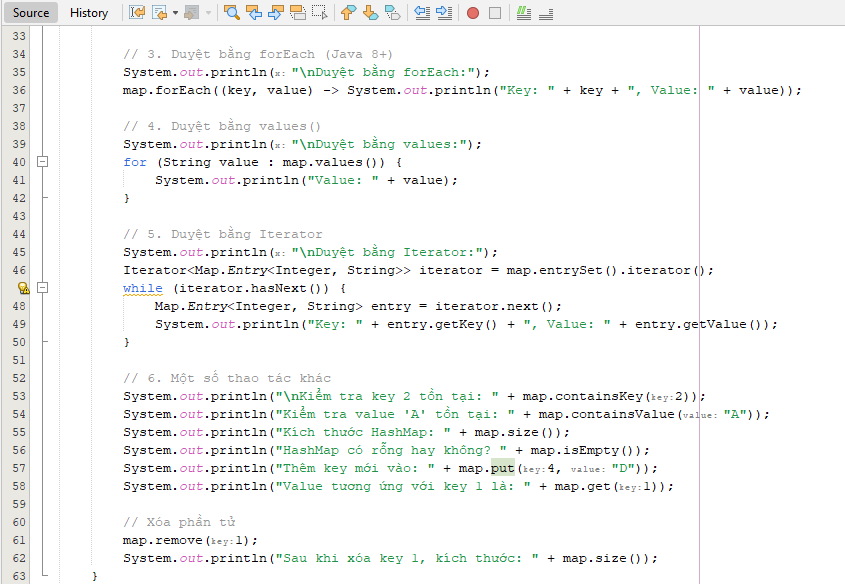
- put(key, value): Thêm cặp key-value

- remove(key): Xóa phần tử theo key

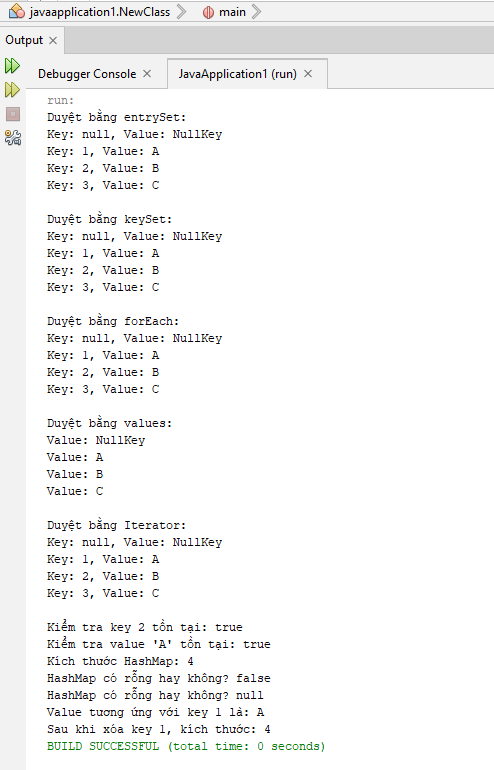
- clear(): Xóa tất cả phần tử

II. Viết mã Java minh họa duyệt phần tử





- Kết quả:



Giải thích chi tiết:

// Tạo HashMap

HashMap<Integer, String> map = new HashMap<>();

map.put(1, "A");

map.put(2, "B");

map.put(3, "C");

map.put(null, "NullKey"); // Key null

- HashMap<Integer, String>: Khai báo một biến map là kiểu HashMap với: integer là kiểu của key và String là kiểu của value.

- Gán các giá trị cho các cặp key và value bằng map.put.

// 1. Duyệt bằng entrySet()

System.out.println("Duyệt bằng entrySet:");

for (Map.Entry<Integer, String> entry : map.entrySet()) {

System.out.println("Key: " + entry.getKey() + ", Value: " + entry.getValue());

}

- Khai báo biến entry và sử dụng map.entrySet() để duyệt qua tất cả các cặp key và value có trong HashMap, entry.getKey() và entry.getValue() là trả về dữ liệu của key và value.

// 2. Duyệt bằng keySet()

System.out.println("\nDuyệt bằng keySet:");

for (Integer key : map.keySet()) {

System.out.println("Key: " + key + ", Value: " + map.get(key));

}

- Duyệt qua tập hợp các key, key ở đây có kiểu integer, sử dụng map.keySet() để duyệt các key, sau đó lấy value qua map.get(key).

// 3. Duyệt bằng forEach

System.out.println("\nDuyệt bằng forEach:");

map.forEach((key, value) -> System.out.println("Key: " + key + ", Value: " + value));

- Sử dụng map.forEach() để duyệt qua từng cặp key và value. Cặp key và value được truy cập trực tiếp thông qua biểu thức lambda (key, value) -> {...} sau đó in ra giá trị tương ứng.

// 4. Duyệt bằng values()

System.out.println("\nDuyệt bằng values:");

for (String value : map.values()) {

System.out.println("Value: " + value);

}

- Duyệt qua tập hợp các value (bỏ qua key), value có kiểu String và ở đây sử dụng map.values() để duyệt qua các value.

// 5. Duyệt bằng Iterator

System.out.println("\nDuyệt bằng Iterator:");

Iterator<Map.Entry<Integer, String>> iterator = map.entrySet().iterator();

while (iterator.hasNext()) {

Map.Entry<Integer, String> entry = iterator.next();

System.out.println("Key: " + entry.getKey() + ", Value: " + entry.getValue());

}

- Sử dụng Iterator để duyệt qua từng phần tử trong HashMap. Đầu tiên, gọi map.entrySet().iterator() để lấy Iterator duyệt qua các cặp key và value. Trong vòng lặp while, kiểm tra phần tử kế tiếp bằng iterator.hasNext(). Dùng iterator.next() để lấy từng phần tử (entry) và in ra key và value.

// 6. Một số thao tác khác

System.out.println("\nKiểm tra key 2 tồn tại: " + map.containsKey(2));

System.out.println("Kiểm tra value 'A' tồn tại: " + map.containsValue("A"));

System.out.println("Kích thước HashMap: " + map.size());

System.out.println("HashMap có rỗng hay không? " + map.isEmpty());

System.out.println("Thêm key mới vào: " + map.put(4, "D"));

System.out.println("Value tương ứng với key 1 là: " + map.get(1));

// Xóa phần tử

map.remove(1);

System.out.println("Sau khi xóa key 1, kích thước: " + map.size());

- map.containsKey(2): Kiểm tra xem key 2 có tồn tại không, nếu có trả về true, không thì trả về false

- map.containsValue(“A”): Kiểm tra xem value A có tồn tại không, nếu có trả về true, không thì trả về false

- map.size(): Trả về số lượng cặp key-value, ở đây HashMap có chứa 4 cặp key-value nên sẽ trả về 4

- map.isEmpty(): Kiểm tra HashMap có rỗng không, vì HashMap đã chứa 4 cặp key-value nên sẽ không rỗng, nếu có trả về true, không thì trả về false

- map.put(4, “D”): Thêm cặp key-value mới (4, “D”), do cặp value này chưa tồn tại trong HashMap nên sẽ trả về null và cặp key-value này đã được thêm thành công

- map.get(1): Lấy value theo key 1, trả về value là A

- map.remove(1): Xóa phần tử theo key 1, ở đây do ban đầu HashMap đã có 4 cặp key-value, sau đó thêm một cặp key-value mới ở map.put(4, “D”) nên sẽ thành 5 cặp key-value ở thời điểm hiện tại, sau khi xóa đi key 1 thì kết quả trả về sẽ là kích thước 4