**HashMap trong Java**

Đăng bởi: VanNguyen- Vào ngày: 04-11-2017- View: 4002

Trong bài này, tôi sẽ hướng dẫn đến các bạn tìm hiểu loại Interface Collection tiếp theo - đó là **HashMap trong Java**. Nội dung của bài này sẽ mô tả đặc điểm, các phương thức thường dùng của Collection này. Với mỗi phương thức được liệt kê, tôi sẽ đưa ra ví dụ đơn giản để cho các bạn nắm bắt được.

|  |  |
| --- | --- |
| **STT** | **Phương thức và Miêu tả** |
| 1 | **void clear()**  Gỡ bỏ tất cả mappings (ánh xạ) từ map này |
| 2 | **Object clone()**  Trả về một shallow copy của đối tượng HashMap này: các key và value chính nó không bị mô phỏng |
| 3 | **boolean containsKey(Object key)**  Trả về true nếu map này chứa một ánh xạ với key đã cho |
| 4 | **boolean containsValue(Object value)**  Trả về true nếu map này so khớp một hoặc nhiều key tới value đã cho |
| 5 | **Set entrySet()**  Trả về một tập hợp các ánh xạ được chứa trong map này |
| 6 | **Object get(Object key)**  Trả về value từ đó key đã cho được so khớp với trong hash map này, hoặc trả về null nếu map không chứa ánh xạ nào cho key này |
| 7 | **boolean isEmpty()**  Trả về true nếu map này không chứa các ánh xạ key-value nào |
| 8 | **Set keySet()**  Trả về một tập hợp các key được chứa trong map này |
| 9 | **Object put(Object key, Object value)**  Liên kết với value đã cho với key đã xác định trong map này |
| 10 | **putAll(Map m)**  Sao chép tất cả ánh xạ từ map đã xác định tới map này. Những ánh xạ này sẽ thay thế bất kỳ ánh xạ nào mà map này có cho bất kỳ key hiện tại nào trong map đã xác định |
| 11 | **Object remove(Object key)**  Gỡ bỏ ánh xạ cho key này từ map đã xác định nếu có mặt |
| 12 | **int size()**  Trả về số ánh xạ key-value trong map này |
| 13 | **Collection values()**  Trả về một tập hợp của các value được chứa trong map này |

**Nội dung chính**

* [1. Đặc điểm](https://freetuts.net/hashmap-trong-java-1099.html#goto-h2-0)
* [2. Các thao tác cơ bản trên HashMap](https://freetuts.net/hashmap-trong-java-1099.html#goto-h2-1)
  + [Tạo mới một HashMap](https://freetuts.net/hashmap-trong-java-1099.html#goto-h3-0)
  + [Hiển thị các phần tử có trong HashMap.](https://freetuts.net/hashmap-trong-java-1099.html#goto-h3-1)
  + [Thêm dữ liệu vào trong HashMap](https://freetuts.net/hashmap-trong-java-1099.html#goto-h3-2)
  + [Lấy dữ liệu value trong HashMap khi biết được key](https://freetuts.net/hashmap-trong-java-1099.html#goto-h3-3)
  + [Xóa 1 entry trong HashMap](https://freetuts.net/hashmap-trong-java-1099.html#goto-h3-4)
  + [Thay thế value của 1 entry trong HashMap](https://freetuts.net/hashmap-trong-java-1099.html#goto-h3-5)
  + [Sao chép HashMap](https://freetuts.net/hashmap-trong-java-1099.html#goto-h3-6)
* [3. Lời kết](https://freetuts.net/hashmap-trong-java-1099.html#goto-h2-2)

**1. Đặc điểm**

HashMap là một lớp triển khai của Map Interface trong Collections Framework nên nó sẽ có một vài đặc điểm và phương thức tương đồng với Map. Giá trị của mỗi phần tử trong HashMap bao gồm 2 phần đó là khóa (*key*) và giá trị tương ứng của key đó (*value*) và khóa của các phần tử này là duy nhất. HashMap cho phép truy xuất trực tiếp dữ liệu bằng khóa duy nhất của nó. Thứ tự các phần tử trong HashMap không dựa theo thứ tự lúc thêm vào.

**2. Các thao tác cơ bản trên HashMap**

**Tạo mới một HashMap**

Để khai báo một HashMap, chúng ta cần phải import gói thư viện java.util.HashMap của Java. Cú pháp import như sau:

**Cú pháp**

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | // Khai báo HashMap  // thì import gói thư viện java.util.HashMap  import java.util.HashMap;  public class TênClass {      // ...  } |

Sau đây là ví dụ cách tạo mới một HashMap trong Java:

**Ví dụ**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22 | public static void main(String[] args) {      // khai báo 1 HashMap tên hashMap1      // mỗi phần tử trong hashMap1 bao gồm 2 phần      // key (Integer) và value (Float)      HashMap<Integer, Float> hashMap1 = new HashMap<>();        // khai báo 1 HashMap tên hashMap2      // mỗi phần tử trong hashMap2 bao gồm 2 phần      // key (String) và value (String)      // có kích thước khởi tạo = 10      HashMap<String, String> hashMap2 = new HashMap<>(10);        // khai báo 1 HashMap có kích thước khởi tạo = 4      // và yếu tố tải = 0.75f (mặc định)      // chi tiết về kích thước khởi tạo và yếu tố tải      // các bạn xem lại bài HashSet      HashMap<Double, Double> hashMap3 = new HashMap<>(4, 0.75f);        // khai báo 1 HashMap được tạo thành từ 1 Collection khác      HashMap<Float, Integer> hashMap4 = new HashMap<>(new TreeMap<>());  } |

**Hiển thị các phần tử có trong HashMap.**

Để hiển thị các phần tử có trong HashMap, chúng ta có các cách như sau:

**Sử dụng vòng lặp for cải tiến.**

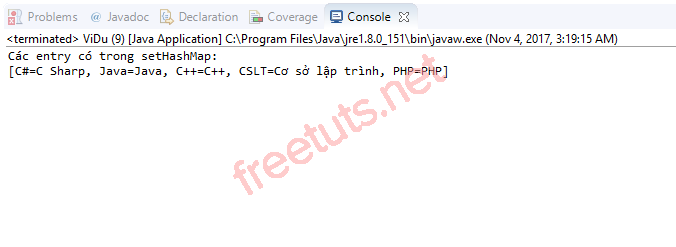
***Hiển thị toàn bộ các entry của HashMap.***

Để lấy toàn bộ các entry của HashMap (*1 entry sẽ bao gồm key và value tương ứng với key đó)*, Java cung cấp cho chúng ta phương thức entrySet(). Phương thức này sẽ trả về 1 Set bao gồm các entry có trong HashMap. Ví dụ dưới đây sẽ minh họa cách sử dụng phương thức này.

**Ví dụ**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23 | public static void main(String[] args) {      // khai báo 1 HashMap có tên là hashMap      // kiểu dữ liệu của key và value là String      HashMap<String, String> hashMap = new HashMap<>();        // Thêm value vào trong hashMap với key tương ứng      // sử dụng phương thức put()      // đối số thứ nhất trong put là key có kiểu là String      // và đối số thứ hai là value có kiểu là String      hashMap.put("CSLT", "Cơ sở lập trình");      hashMap.put("C++", "C++");      hashMap.put("C#", "C Sharp");      hashMap.put("PHP", "PHP");      hashMap.put("Java", "Java");        // tạo 1 Set có tên là setHashMap      // chứa toàn bộ các entry (vừa key vừa value)      // của hashMap      Set<Map.Entry<String, String>> setHashMap = hashMap.entrySet();        System.out.println("Các entry có trong setHashMap:");      System.out.println(setHashMap);  } |

Kết quả sau khi biên dịch chương trình;

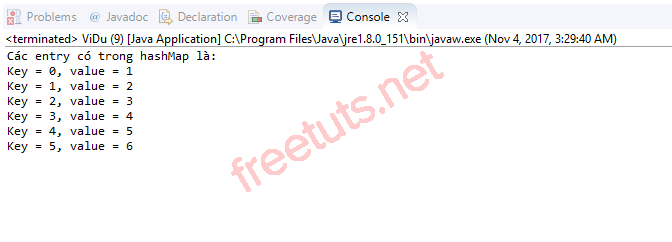


Ngoài ra, kể từ Java 8 trở đi chúng ta có thể lấy toàn bộ các entry trong HashMap bằng cách sử dụng forEach() như sau:

**Ví dụ**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | public static void main(String[] args) {      HashMap<Integer, Integer> hashMap = new HashMap<>();      hashMap.put(0, 1);      hashMap.put(1, 2);      hashMap.put(2, 3);      hashMap.put(3, 4);      hashMap.put(4, 5);      hashMap.put(5, 6);        // Cách duyệt HashMap với forEach() trong Java 8      // đối số thứ nhất bên trong forEach là key      // đối số thứ hai bên trong forEach là value      System.out.println("Các entry có trong hashMap là: ");      hashMap.forEach((keyInt, valueInt) -> System.out.println(          "Key = " + keyInt + ", value = " + valueInt));  } |

Kết quả sau khi biên dịch chương trình:



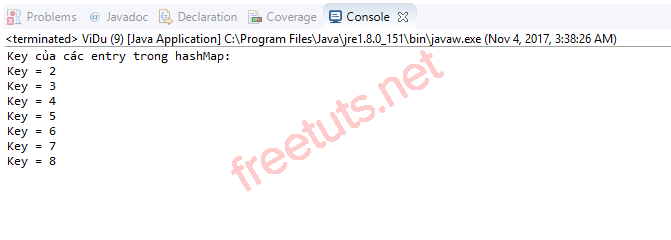
***Lấy toàn bộ key của HashMap.***

Để lấy toàn bộ key của HashMap, Java cung cấp cho chúng ta phương thức keySet(). Phương thức này sẽ trả về 1 Set bao gồm các key có trong HashMap. Ví dụ dưới đây sẽ minh họa cách sử dụng phương thức này.

**Ví dụ**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | public static void main(String[] args) {      HashMap<Integer, String> hashMap = new HashMap<>();      hashMap.put(2, "Thứ hai");      hashMap.put(3, "Thứ ba");      hashMap.put(4, "Thứ tư");      hashMap.put(5, "Thứ năm");      hashMap.put(6, "Thứ sáu");      hashMap.put(7, "Thứ bảy");      hashMap.put(8, "Chủ nhật");        // phương thức keySet()      // sẽ trả về 1 Set chứa key có trong hashMap      System.out.println("Key của các entry trong hashMap: ");      for (int key : hashMap.keySet()) {          System.out.println("Key = " + key);      }  } |

Kết quả sau khi biên dịch chương trình:



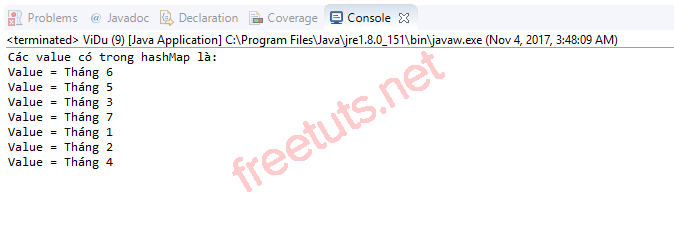
***Lấy toàn bộ value của HashMap.***

Để lấy toàn bộ value của HashMap, Java cung cấp cho chúng ta phương thức values(). Phương thức này sẽ trả về 1 tập hợp bao gồm các value có trong HashMap. Ví dụ dưới đây sẽ minh họa cách sử dụng phương thức này.

**Ví dụ**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | public static void main(String[] args) {      HashMap<String, String> hashMap = new HashMap<>();      hashMap.put("January", "Tháng 1");      hashMap.put("February", "Tháng 2");      hashMap.put("March", "Tháng 3");      hashMap.put("April", "Tháng 4");      hashMap.put("May", "Tháng 5");      hashMap.put("June", "Tháng 6");      hashMap.put("July", "Tháng 7");        // phương thức values() sẽ trả về      // một tập hợp gồm các value có trong hashMap      System.out.println("Các value có trong hashMap là: ");      for (String value : hashMap.values()) {          System.out.println("Value = " + value);      }  } |

Kết quả sau khi biên dịch chương trình:



**Sử dụng Iterator.**

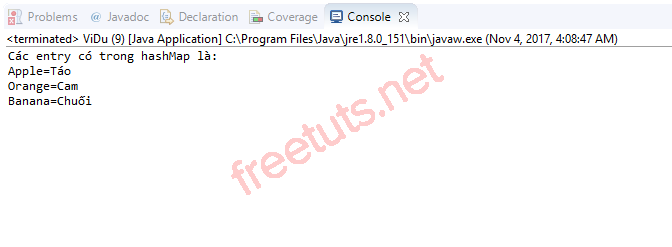
Để sử dụng được Iterator chúng ta cần phải import gói thư viện java.util.Iterator của Java.

***Lấy toàn bộ các entry của HashMap.***

**Ví dụ**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | public static void main(String[] args) {      Map<String, String> hashMap = new HashMap<>();      hashMap.put("Apple", "Táo");      hashMap.put("Banana", "Chuối");      hashMap.put("Orange", "Cam");        // sử dụng Iterator để lấy toàn bộ entry của hashMap      // vì 1 entry bao gồm key và value      // nên kiểu dữ liệu của Iterator sẽ bao gồm      // kiểu dữ liệu của cả key và value      Iterator<Map.Entry<String, String>> iterator = hashMap.entrySet().iterator();        System.out.println("Các entry có trong hashMap là: ");      while (iterator.hasNext()) {          System.out.println(iterator.next());      }  } |

Kết quả sau khi biên dịch chương trình:

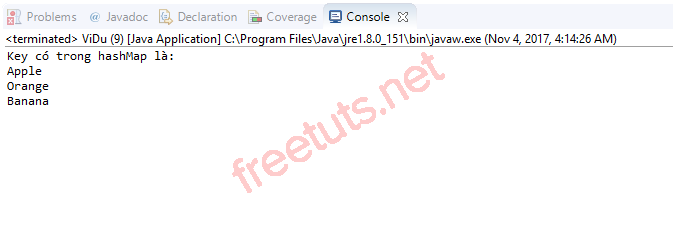


***Lấy toàn bộ key của HashMap.***

***Ví dụ***

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | public static void main(String[] args) {      Map<String, String> hashMap = new HashMap<>();      hashMap.put("Apple", "Táo");      hashMap.put("Banana", "Chuối");      hashMap.put("Orange", "Cam");        // sử dụng Iterator để lấy toàn bộ key của hashMap      // thông qua phương thức keySet()      // vì các key có kiểu là String      // nên iterator cũng có kiểu là String      Iterator<String> iterator = hashMap.keySet().iterator();        System.out.println("Key có trong hashMap là: ");      while (iterator.hasNext()) {          System.out.println(iterator.next());      }  } |

Kết quả sau khi biên dịch chương trình:

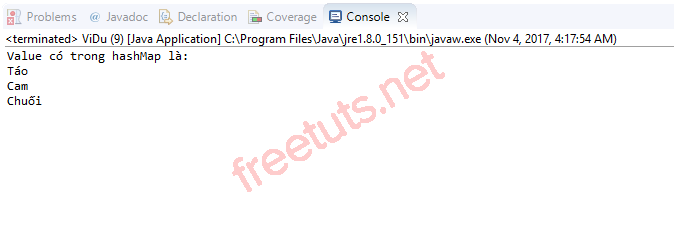


***Lấy toàn bộ value của HashMap.***

**Ví dụ**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | public static void main(String[] args) {      Map<String, String> hashMap = new HashMap<>();      hashMap.put("Apple", "Táo");      hashMap.put("Banana", "Chuối");      hashMap.put("Orange", "Cam");        // sử dụng Iterator để lấy toàn bộ value của HashMap      // thông qua phương thức values()      // vì các value có kiểu là String      // nên iterator cũng có kiểu là String      Iterator<String> iterator = hashMap.values().iterator();        System.out.println("Value có trong hashMap là: ");      while (iterator.hasNext()) {          System.out.println(iterator.next());      }  } |

Kết quả sau khi biên dịch chương trình:



**Thêm dữ liệu vào trong HashMap**

Để thêm dữ liệu vào trong HashMap, Java cung cấp cho chúng ta phương thức put().

**Cú pháp**

|  |  |
| --- | --- |
| 1 | put(K key, V value); |

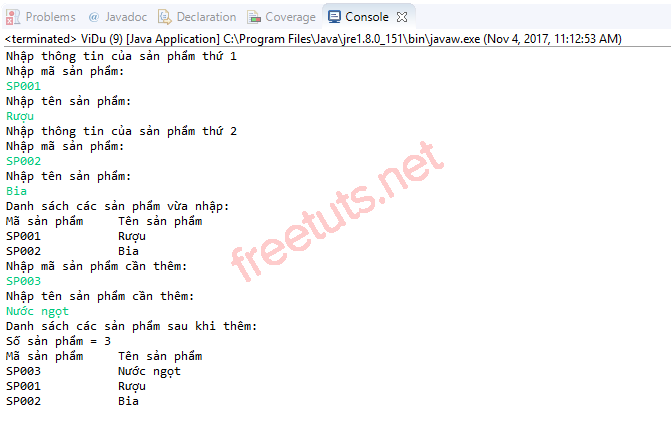
, trong đó: key là khóa, value là giá trị. Mỗi key sẽ tương ứng với một value cụ thể.

**Ví dụ**

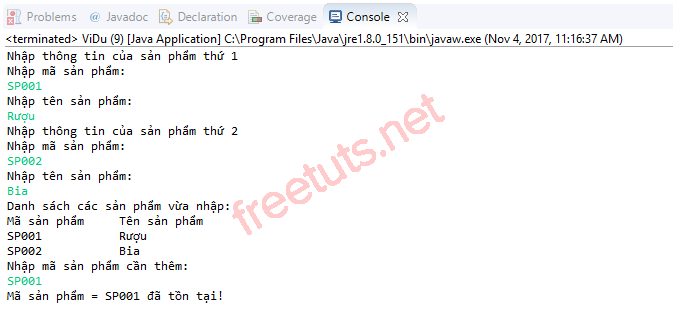
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57 | public static void main(String[] args) {      int soSanPham = 2;      HashMap<String, String> hashMapProducts = new HashMap<>();      Scanner scanner = new Scanner(System.in);      String maSanPham, tenSanPham;        // thêm thông tin của 2 sản phẩm      // vào trong hashMapProducts      // trong đó key là mã sản phẩm, còn value là tên của sản phẩm đó      for (int i = 1; i <= soSanPham; i++) {          System.out.println("Nhập thông tin của sản phẩm thứ " + i);          System.out.println("Nhập mã sản phẩm: ");          maSanPham = scanner.nextLine();          System.out.println("Nhập tên sản phẩm: ");          tenSanPham = scanner.nextLine();          hashMapProducts.put(maSanPham, tenSanPham);      }        // hiển thị danh sách sản phẩm sử dụng Iterator      System.out.println("Danh sách các sản phẩm vừa nhập: ");      System.out.println("Mã sản phẩm\tTên sản phẩm");      Iterator<Map.Entry<String, String>> iterator = hashMapProducts.entrySet().iterator();      while (iterator.hasNext()) {          // tạo 1 entry          Map.Entry<String, String> entry = iterator.next();          System.out.println(entry.getKey() + "\t\t" + entry.getValue());      }        // thêm 1 sản phẩm mới vào trong hashMapProducts      // nếu mã sản phẩm đó đã tồn tại thì thông báo mã đã tồn tại      // ngược lại thì mới nhập tên sản phẩm      // và thêm vào bình thường sau đó thông báo "Thêm thành công"      // tăng số sản phẩm lên 1      System.out.println("Nhập mã sản phẩm cần thêm: ");      String maSanPhamMoi = scanner.nextLine();        // phương thức containsKey() sẽ kiểm tra mã sản phẩm mới nhập vào      // có tồn tại trong hashMapProducts hay chưa      if (hashMapProducts.containsKey(maSanPhamMoi)) {          System.out.println("Mã sản phẩm = " + maSanPhamMoi + " đã tồn tại!");      } else {          System.out.println("Nhập tên sản phẩm cần thêm: ");          String tenSanPhamMoi = scanner.nextLine();            hashMapProducts.put(maSanPhamMoi, tenSanPhamMoi);          soSanPham++;          System.out.println("Danh sách các sản phẩm sau khi thêm: ");          System.out.println("Số sản phẩm = " + soSanPham);          System.out.println("Mã sản phẩm\tTên sản phẩm");          iterator = hashMapProducts.entrySet().iterator();          while (iterator.hasNext()) {              // tạo 1 entry              Map.Entry<String, String> entry = iterator.next();              System.out.println(entry.getKey() + "\t\t" + entry.getValue());          }      }  } |

**Kết quả sau khi biên dịch chương trình:**

Thêm dữ liệu thành công:



Thêm dữ liệu thất bại:



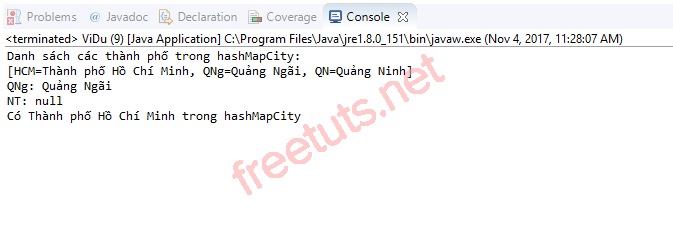
**Lấy dữ liệu value trong HashMap khi biết được key**

Để lấy dữ liệu value trong HashMap khi biết được key liên kết với value đó, Java cung cấp cho chúng ta phương thức get(). Phương thức này sẽ trả về giá trị (*value*) tương ứng với key đó, nếu trong HashMap không có key đó thì phương thức này sẽ trả về giá trị null.

**Ví dụ**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30 | public static void main(String[] args) {      HashMap<String, String> hashMapCity = new HashMap<>();      hashMapCity.put("QNg", "Quảng Ngãi");      hashMapCity.put("QN", "Quảng Nam");      // trong trường hợp này ta thấy      // key của Quảng Nam và Quảng Ninh      // đều là QN nên chương trình sẽ thêm      // vào trong hashMapCity value đứng sau (tức là Quảng Ninh)      hashMapCity.put("QN", "Quảng Ninh");      hashMapCity.put("HCM", "Thành phố Hồ Chí Minh");        System.out.println("Danh sách các thành phố trong hashMapCity: ");      Set<Map.Entry<String, String>> setCity = hashMapCity.entrySet();      System.out.println(setCity);        // lấy thành phố có mã là QNg      // và hiển thị tên thành phố      System.out.println("QNg: " + hashMapCity.get("QNg"));        // lấy thành phố có mã là NT      // vì trong hashMapCity không có thành phố nào có mã là NT      // nên sẽ hiển thị giá trị null      System.out.println("NT: " + hashMapCity.get("NT"));        // Để kiểm tra xem 1 value có trong HashMap hay không      // chúng ta sẽ dùng phương thức containsValue()      if (hashMapCity.containsValue("Thành phố Hồ Chí Minh")) {          System.out.println("Có Thành phố Hồ Chí Minh trong hashMapCity");      }  } |

Kết quả sau khi biên dịch chương trình:



**Xóa 1 entry trong HashMap**

Để xóa 1 entry trong HashMap, Java cung cấp cho chúng ta phương thức remove().

**Ví dụ**

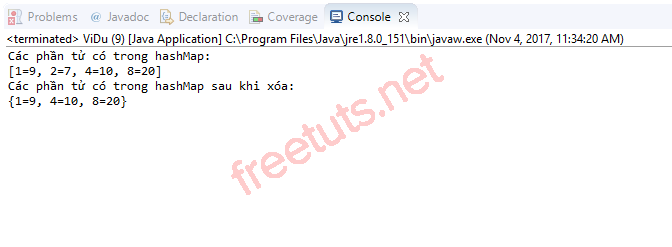
|  |  |
| --- | --- |
| 1 | remove(K key); |

, trong đó key là khóa của entry cần xóa.

**Ví dụ**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | public static void main(String[] args) {      HashMap<Integer, Integer> hashMap = new HashMap<>();      hashMap.put(1, 9);      hashMap.put(4, 10);      hashMap.put(2, 7);      hashMap.put(8, 20);        System.out.println("Các phần tử có trong hashMap: ");      Set<Map.Entry<Integer, Integer>> setMap = hashMap.entrySet();      System.out.println(setMap);        // xóa entry có khóa bằng 2 ra khỏi hashMap      // sử dụng phương thức remove()      hashMap.remove(2);        System.out.println("Các phần tử có trong hashMap sau khi xóa: ");      System.out.println(hashMap);  } |

Kết quả sau khi biên dịch chương trình:



**Thay thế value của 1 entry trong HashMap**

Để thay thế value của 1 entry trong HashMap, Java cung cấp cho chúng ta 2 dạng của phương thức replace() như sau:

**Cú pháp**

|  |  |
| --- | --- |
| 1 | replace(K key, V value); |

,trong đó key là khóa của entry cần thay thế, value là giá trị mới được thay thế.

**Cú pháp**

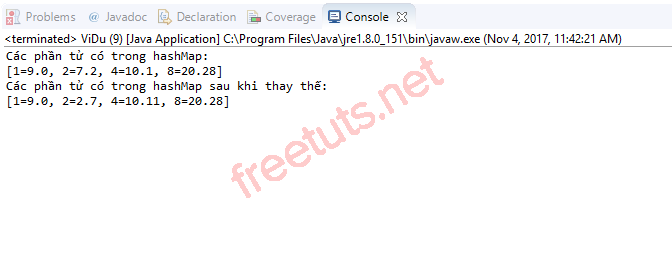
|  |  |
| --- | --- |
| 1 | replace(K key, V oldValue, V newValue); |

, trong đó key là khóa của entry cần thay thế, oldValue là giá trị cần thay thế, newValue là giá trị mới được thay thế.

**Ví dụ**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24 | public static void main(String[] args) {      HashMap<Integer, Double> hashMap = new HashMap<>();      hashMap.put(1, 9d);      hashMap.put(4, 10.1d);      hashMap.put(2, 7.2d);      hashMap.put(8, 20.28d);        System.out.println("Các phần tử có trong hashMap: ");      Set<Map.Entry<Integer, Double>> setMap = hashMap.entrySet();      System.out.println(setMap);        // thay thế value của entry có khóa là 4      // thành 10.11d      hashMap.replace(4, 10.11d);        // ngoài ra chúng ta có thế thay thế như sau      // câu lệnh bên dưới sẽ thay thế entry      // có key là 2, value là 7.2d thành 2.7d      hashMap.replace(2, 7.2d, 2.7d);        System.out.println("Các phần tử có trong hashMap sau khi thay thế: ");      setMap = hashMap.entrySet();      System.out.println(setMap);  } |

Kết quả sau khi biên dịch chương trình:



**Sao chép HashMap**

Để sao chép các entry có trong HashMap này vào trong 1 HashMap khác, Java cung cấp cho chúng ta phương thức putAll().

**Cú pháp**

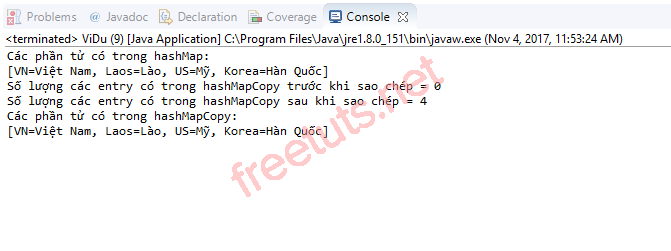
|  |  |
| --- | --- |
| 1 | putAll(HashMap m); |

, trong đó m là tên của HashMap được sao chép.

**Ví dụ**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28 | public static void main(String[] args) {      HashMap<String, String> hashMap = new HashMap<>();      hashMap.put("VN", "Việt Nam");      hashMap.put("Laos", "Lào");      hashMap.put("Korea", "Hàn Quốc");      hashMap.put("US", "Mỹ");        System.out.println("Các phần tử có trong hashMap: ");      Set<Map.Entry<String, String>> setHashMap = hashMap.entrySet();      System.out.println(setHashMap);        // tạo 1 HashMap rỗng      HashMap<String, String> hashMapCopy = new HashMap<>();        // phương thức size() sẽ trả về số lượng entry có trong hashMapCopy      System.out.println("Số lượng các entry có trong hashMapCopy "          + "trước khi sao chép = " + (hashMapCopy.size()));        // sao chép các entry của hashMap      // vào trong hashMapCopy      hashMapCopy.putAll(hashMap);        System.out.println("Số lượng các entry có trong hashMapCopy "          + "sau khi sao chép = " + (hashMapCopy.size()));      System.out.println("Các phần tử có trong hashMapCopy: ");      Set<Map.Entry<String, String>> setHashMapCopy = hashMapCopy.entrySet();      System.out.println(setHashMapCopy);  } |

Kết quả sau khi biên dịch chương trình:



**3. Lời kết**

Trong bài này, tôi đã giới thiệu cho các bạn đặc điểm, các phương thức thường dùng đối với HashMaptrong Java. Sang bài sau tôi sẽ giới thiệu đến các bạn dạng Class Collection cuối cùng, đó là **TreeMap trong Java**. Các bạn theo dõi nhé!