**TreeMap trong Java**

**1. Đặc điểm**

Tương tự như HashMap, TreeMap 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. Tuy nhiên, giữa HashMap và TreeMap có sự khác biệt chính đó là thứ tự các phần tử trong HashMap không được sắp xếp theo bất kỳ thứ tự nào, trong khi đó TreeMap sắp xếp các phần tử theo thứ tự tăng dần của khóa.

**2. Các phương thức phổ biến**

### **2.1. Tạo mới một TreeMap**

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

**Cú pháp**

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

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

**Ví dụ**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | public static void main(String[] args) {      // khai báo 1 TreeMap tên treeMap1      // mỗi phần tử trong treeMap1 bao gồm 2 phần      // key (String) và value (Double)      TreeMap<String, Double> treeMap1 = new TreeMap<>();        // khai báo 1 TreeMap được tạo thành từ 1 Map      Map<Float, Integer> map = new HashMap<>();      TreeMap<Float, Integer> treeMap2 = new TreeMap<>(map);  } |

### **2.2. Hiển thị các phần tử có trong TreeMap**

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

**Sử dụng vòng lặp for cải tiến.**

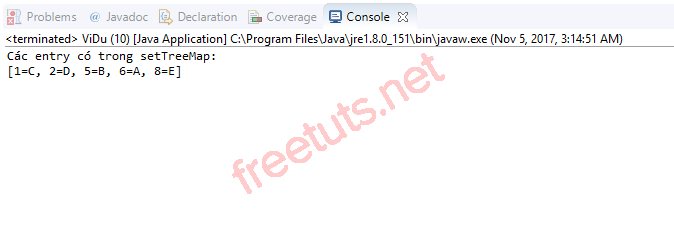
**Hiển thị toàn bộ các entry của TreeMap.**

Để lấy toàn bộ các entry của TreeMap (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 TreeMap. 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  24  25 | public static void main(String[] args) {      // khai báo 1 TreeMap có tên là treeMap      // kiểu dữ liệu của key là Integer      // và value là Character      TreeMap<Integer, Character> treeMap = new TreeMap<>();        // Thêm value vào trong treeMap với key tương ứng      // sử dụng phương thức put()      // đối số thứ nhất trong put là key có kiểu là Integer      // và đối số thứ hai là value có kiểu là Character      treeMap.put(6, 'A');      treeMap.put(5, 'B');      treeMap.put(1, 'C');      treeMap.put(2, 'D');      treeMap.put(8, 'E');        // tạo 1 Set có tên là setTreeMap      // chứa toàn bộ các entry (vừa key vừa value)      // của treeMap      Set<Entry<Integer, Character>> setTreeMap = treeMap.entrySet();        // các entry trong setTreeMap sẽ được sắp xếp tăng dần theo khóa      System.out.println("Các entry có trong setTreeMap:");      System.out.println(setTreeMap);  } |

Kết quả sau khi biên dịch chương trình:

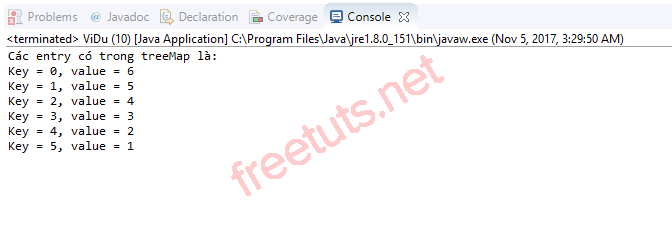


Ngoài ra, kể từ Java 8 trở đi chúng ta có thể lấy toàn bộ các entry trong TreeMap bằng cách sử dụng forEach() như sau:

**Ví dụ**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | public static void main(String[] args) {      TreeMap<Integer, Integer> treeMap = new TreeMap<>();      treeMap.put(0, 6);      treeMap.put(1, 5);      treeMap.put(2, 4);      treeMap.put(3, 3);      treeMap.put(4, 2);      treeMap.put(5, 1);        // Cách duyệt treeMap 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 treeMap là: ");      treeMap.forEach((keyInt, valueInt) -> System.out.println(          "Key = " + keyInt + ", value = " + valueInt));  } |

Kết quả sau khi biên dịch chương trình:



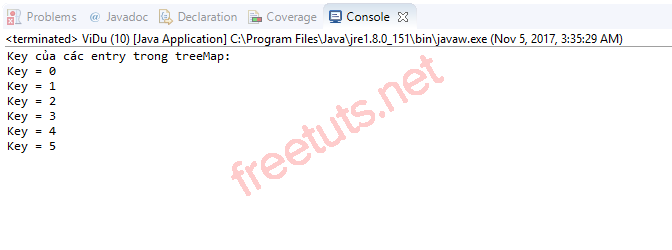
**Lấy toàn bộ key của TreeMap.**

Để lấy toàn bộ key của TreeMap, Java cung cấp cho chúng ta phương thức keySet(). Phương thức này sẽ trả về 1 Set bao gồm các key có trong TreeMap. 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 | public static void main(String[] args) {      TreeMap<Integer, Integer> treeMap = new TreeMap<>();      treeMap.put(0, 6);      treeMap.put(1, 5);      treeMap.put(2, 4);      treeMap.put(3, 3);      treeMap.put(4, 2);      treeMap.put(5, 1);        // phương thức keySet()      // sẽ trả về 1 Set chứa key có trong treeMap      System.out.println("Key của các entry trong treeMap: ");      for (int key : treeMap.keySet()) {          System.out.println("Key = " + key);      }  } |

Kết quả sau khi biên dịch chương trình:



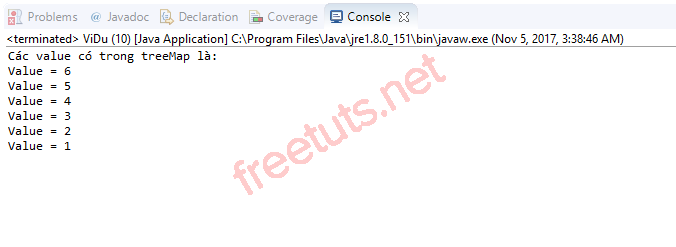
**Lấy toàn bộ value của TreeMap.**

Để lấy toàn bộ value của TreeMap, Java cung cấp cho chúng ta phương thức values(). Phương thức này sẽ trả về 1 tập hợp bao gồm các value có trong TreeMap. 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 | public static void main(String[] args) {      TreeMap<Integer, Integer> treeMap = new TreeMap<>();      treeMap.put(0, 6);      treeMap.put(1, 5);      treeMap.put(2, 4);      treeMap.put(3, 3);      treeMap.put(4, 2);      treeMap.put(5, 1);        // phương thức values() sẽ trả về      // một tập hợp gồm các value có trong treeMap      System.out.println("Các value có trong treeMap là: ");      for (Integer value : treeMap.values()) {          System.out.println("Value = " + value);      }  } |

Kết quả sau khi biên dịch chương trình:



**Sử dụng Iterator.**

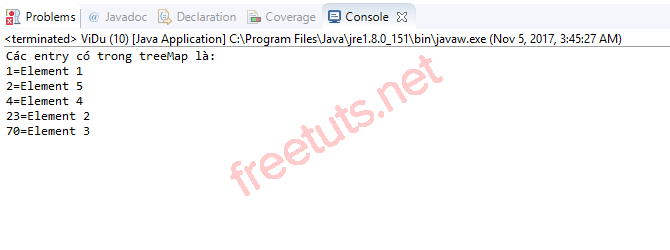
Để sử dụng được Iterator chúng ta cần phải import gói thư viện java.util.Iterator của Java.

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

**Ví dụ**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | public static void main(String[] args) {      TreeMap<Integer, String> treeMap = new TreeMap<>();      treeMap.put(1, "Element 1");      treeMap.put(23, "Element 2");      treeMap.put(70, "Element 3");      treeMap.put(4, "Element 4");      treeMap.put(2, "Element 5");        // sử dụng Iterator để lấy toàn bộ entry của treeMap      // 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<Integer, String>> iterator = treeMap.entrySet().iterator();        System.out.println("Các entry có trong treeMap là: ");      while (iterator.hasNext()) {          System.out.println(iterator.next());      }  } |

Kết quả sau khi biên dịch chương trình:

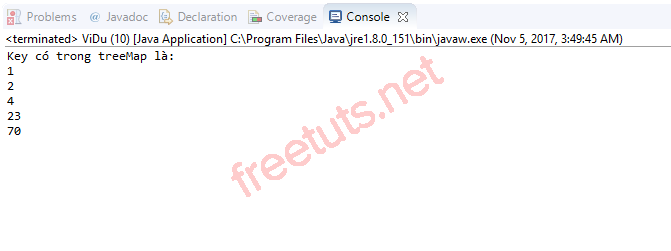


**Lấy toàn bộ key của TreeMap.**

**Ví dụ**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | public static void main(String[] args) {      TreeMap<Integer, String> treeMap = new TreeMap<>();      treeMap.put(1, "Element 1");      treeMap.put(23, "Element 2");      treeMap.put(70, "Element 3");      treeMap.put(4, "Element 4");      treeMap.put(2, "Element 5");        // sử dụng Iterator để lấy toàn bộ key của treeMap      // thông qua phương thức keySet()      // vì các key có kiểu là Integer      // nên iterator cũng có kiểu là Integer      Iterator<Integer> iterator = treeMap.keySet().iterator();        System.out.println("Key có trong treeMap là: ");      while (iterator.hasNext()) {          System.out.println(iterator.next());      }  } |

Kết quả sau khi biên dịch chương trình:

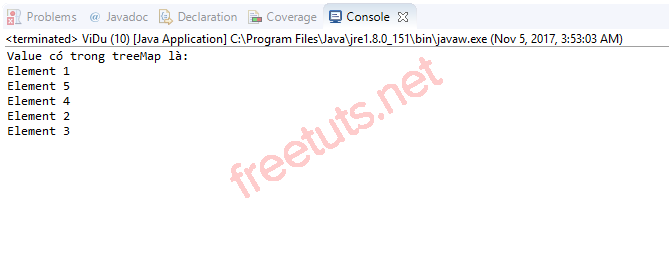


**Lấy toàn bộ value của TreeMap.**

**Ví dụ**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | public static void main(String[] args) {      TreeMap<Integer, String> treeMap = new TreeMap<>();      treeMap.put(1, "Element 1");      treeMap.put(23, "Element 2");      treeMap.put(70, "Element 3");      treeMap.put(4, "Element 4");      treeMap.put(2, "Element 5");        // sử dụng Iterator để lấy toàn bộ value của treeMap      // thông qua phương thức values()      // vì các value có kiểu là String      // nên iterator cũng có kiểu là String      Iterator<String> iterator = treeMap.values().iterator();        // các value của treeMap được sắp xếp theo chiều tăng dần của khóa      System.out.println("Value có trong treeMap là: ");      while (iterator.hasNext()) {          System.out.println(iterator.next());      }  } |

Kết quả sau khi biên dịch chương trình:



### **2.3. Thêm dữ liệu vào trong TreeMap**

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

**Cú pháp**

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

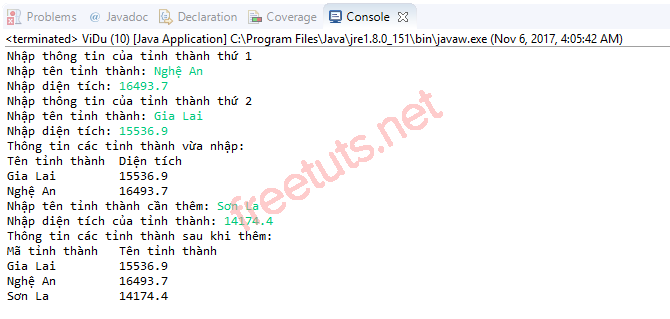
, trong đó: key là khóa, value là giá trị. Mỗi key sẽ tương ứng với một value cụ thể.

**Ví dụ**

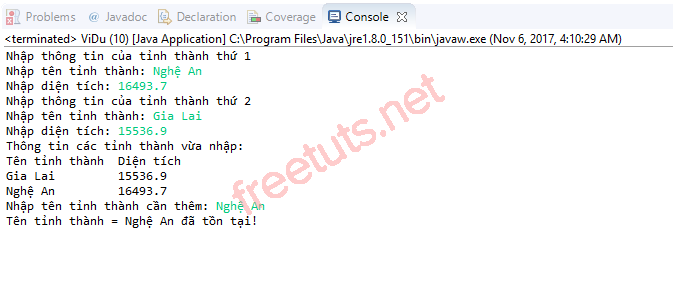
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54 | public static void main(String[] args) {      TreeMap<String, Float>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



### **2.4. Lấy dữ liệu value trong TreeMap khi biết được key**

Để lấy dữ liệu value trong TreeMap 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 TreeMap 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 | public static void main(String[] args) {      TreeMap<Integer, String> treeMap = new TreeMap<>();      treeMap.put(2, "Two");      treeMap.put(1, "One");      treeMap.put(4, "Four");      treeMap.put(5, "Five");      treeMap.put(3, "Three");        System.out.println("Danh sách các entry trong treeMap: ");      Set<Map.Entry<Integer, String>> setTreeMap = treeMap.entrySet();      System.out.println(setTreeMap);        // lấy entry có key = 3      // và hiển thị value      System.out.println("3: " + treeMap.get(3));        // lấy entry có key = 7      // vì trong treeMap không có key = 7      // nên sẽ hiển thị giá trị null      System.out.println("7: " + treeMap.get(7));        // Để kiểm tra xem 1 value có trong TreeMap hay không      // chúng ta sẽ dùng phương thức containsValue()      if (treeMap.containsValue("Two")) {          System.out.println("Có Two trong treeMap.");      }  } |

Kết quả sau khi biên dịch chương trình:



### **2.5. Xóa 1 entry trong TreeMap**

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

**Cú pháp**

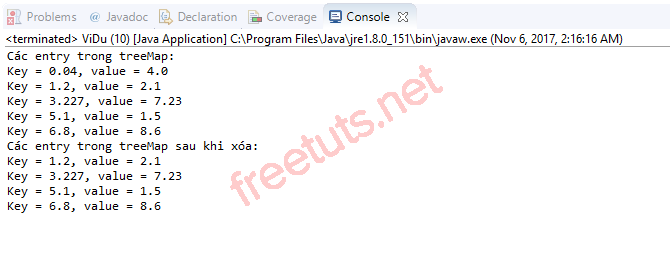
|  |  |
| --- | --- |
| 1 | remove(K key); |

, 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  19  20 | public static void main(String[] args) {      TreeMap<Double, Float> treeMap = new TreeMap<>();      treeMap.put(6.8d, 8.6f);      treeMap.put(1.2d, 2.1f);      treeMap.put(0.04d, 4f);      treeMap.put(5.1d, 1.5f);      treeMap.put(3.227d, 7.23f);        System.out.println("Các entry trong treeMap: ");      treeMap.forEach((keyInt, valueInt) -> System.out.println(          "Key = " + keyInt + ", value = " + valueInt));        // xóa entry có khóa bằng 0.04d ra khỏi treeMap      // sử dụng phương thức remove()      treeMap.remove(0.04d);        System.out.println("Các entry trong treeMap sau khi xóa: ");      treeMap.forEach((keyInt, valueInt) -> System.out.println(          "Key = " + keyInt + ", value = " + valueInt));  } |

Kết quả sau khi biên dịch chương trình:



### **2.6. Thay thế value của 1 entry trong TreeMap**

Để thay thế value của 1 entry trong TreeMap, 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**

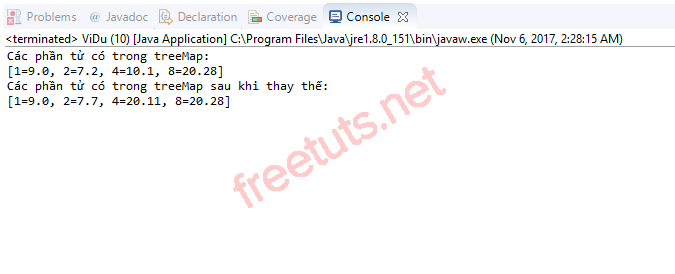
|  |  |
| --- | --- |
| 1 | replace(K key, V oldValue, V newValue); |

, 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) {      TreeMap<Integer, Double> treeMap = new TreeMap<>();      treeMap.put(1, 9d);      treeMap.put(4, 10.1d);      treeMap.put(2, 7.2d);      treeMap.put(8, 20.28d);        System.out.println("Các phần tử có trong treeMap: ");      Set<Map.Entry<Integer, Double>> setTreeMap = treeMap.entrySet();      System.out.println(setTreeMap);        // thay thế value của entry có khóa là 4      // thành 20.11d      treeMap.replace(4, 20.11d);        // 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 7.7d      treeMap.replace(2, 7.2d, 7.7d);        System.out.println("Các phần tử có trong treeMap sau khi thay thế: ");      setTreeMap = treeMap.entrySet();      System.out.println(setTreeMap);  } |

Kết quả sau khi biên dịch chương trình:



### **2.7. Sao chép TreeMap**

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

**Cú pháp**

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

, trong đó m là tên của TreeMap 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  29  30  31  32 | public static void main(String[] args) {      TreeMap<String, String> treeMap = new TreeMap<>();      treeMap.put("Arctic Ocean", "Bắc Băng Dương");      treeMap.put("Pacific Ocean", "Thái Bình Dương");      treeMap.put("Atlantic Ocean", "Đại Tây Dương");      treeMap.put("Indian Ocean", "Ấn Độ Dương");        System.out.println("Các phần tử có trong treeMap: ");      Iterator<Map.Entry<String, String>> iterator = treeMap.entrySet().iterator();      while (iterator.hasNext()) {          System.out.println(iterator.next());      }        // tạo 1 TreeMap rỗng      TreeMap<String, String> treeMapCopy = new TreeMap<>();        // phương thức size() sẽ trả về số lượng entry có trong treeMapCopy      System.out.println("\nSố lượng các entry có trong treeMapCopy "          + "trước khi sao chép = " + (treeMapCopy.size()));        // sao chép các entry của treeMap      // vào trong treeMapCopy      treeMapCopy.putAll(treeMap);        System.out.println("Số lượng các entry có trong treeMapCopy "          + "sau khi sao chép = " + (treeMapCopy.size()));      System.out.println("Các phần tử có trong treeMapCopy: ");      iterator = treeMap.entrySet().iterator();      while (iterator.hasNext()) {          System.out.println(iterator.next());      }  } |

Kết quả sau khi biên dịch chương trình:

\