

- Set: cấu trúc dữ liệu lưu trữ các đối tượng không có thứ tự, các phần tử không lặp lại
 - add: thêm một phần tử vào tập hợp
 - remove: loại bỏ một phần tử khỏi tập hợp
 - contains: kiểm tra xem một phần tử có thuộc tập hợp hay không?
 - Set: là interface, để khởi tạo một tập hợp, ta có thể dùng:
 - TreeSet: cài đặt bằng cây tìm kiếm cân bằng
 - HashSet: cài đặt bằng bảng băm

- Set: cấu trúc dữ liệu lưu trữ các đối tượng không có thứ tự, các phần tử không lặp lại
 - add: thêm một phần tử vào tập hợp
 - remove: loại bỏ một phần tử khỏi tập hợp
 - contains: kiểm tra xem một phần tử có thuộc tập hợp hay không?
 - Set: là interface, để khởi tạo một tập hợp, ta có thể dùng:
 - TreeSet: cài đặt bằng cây tìm kiếm cân bằng
 - HashSet: cài đặt bằng bảng băm

```
import java.util.HashSet;
import java.util.Set;
import java.util.TreeSet;

public class Main{
    public static void main(String[] args){
        Set<String> S = new HashSet<String>();
        for(int i = 1; i <= 5; i++) S.add(i + "");
        for(String e: S) System.out.println(e);
        S.add("3");
        System.out.println("after add 3");
        for(String e: S) System.out.println(e);
    }
}
```

- Set: cấu trúc dữ liệu lưu trữ các đối tượng không có thứ tự, các phần tử không lặp lại
 - add: thêm một phần tử vào tập hợp
 - remove: loại bỏ một phần tử khỏi tập hợp
 - contains: kiểm tra xem một phần tử có thuộc tập hợp hay không?
 - Set: là interface, để khởi tạo một tập hợp, ta có thể dùng:
 - TreeSet: cài đặt bằng cây tìm kiếm cân bằng
 - HashSet: cài đặt bằng bảng băm

```
import java.util.HashSet;
import java.util.Set;
import java.util.TreeSet;

public class Main{
    public static void main(String[] args){
        Set<String> S = new HashSet<String>();
        for(int i = 1; i <= 5; i++) S.add(i + "");
        for(String e: S) System.out.println(e);
        S.add("3");
        System.out.println("after add 3");
        for(String e: S) System.out.println(e);
    }
}
```



```
1
2
3
4
5
after add 3
1
2
3
4
5
```

- Set: cấu trúc dữ liệu lưu trữ các đối tượng không có thứ tự, các phần tử không lặp lại
 - add: thêm một phần tử vào tập hợp
 - remove: loại bỏ một phần tử khỏi tập hợp
 - contains: kiểm tra xem một phần tử có thuộc tập hợp hay không?
 - Set: là interface, để khởi tạo một tập hợp, ta có thể dùng:
 - TreeSet: cài đặt bằng cây tìm kiếm cân bằng
 - HashSet: cài đặt bằng bảng băm

```
import java.util.HashSet;
import java.util.Set;
import java.util.TreeSet;
public class Main{
    public static void main(String[] args){
        Set<String> S = new HashSet<String>();
        for(int i = 1; i <= 5; i++) S.add(i + "");
        for(String e: S) System.out.println(e);
        System.out.println(S.contains("2") + " " + S.contains("10"));
        S.remove("3");
        System.out.println("after removing 3");
        for(String e: S) System.out.println(e);
    }
}
```

- Set: cấu trúc dữ liệu lưu trữ các đối tượng không có thứ tự, các phần tử không lặp lại
 - add: thêm một phần tử vào tập hợp
 - remove: loại bỏ một phần tử khỏi tập hợp
 - contains: kiểm tra xem một phần tử có thuộc tập hợp hay không?
 - Set: là interface, để khởi tạo một tập hợp, ta có thể dùng:
 - TreeSet: cài đặt bằng cây tìm kiếm cân bằng
 - HashSet: cài đặt bằng bảng băm

```
import java.util.HashSet;
import java.util.Set;
import java.util.TreeSet;
public class Main{
    public static void main(String[] args){
        Set<String> S = new HashSet<String>();
        for(int i = 1; i <= 5; i++) S.add(i + "");
        for(String e: S) System.out.println(e);
        System.out.println(S.contains("2") + " " + S.contains("10"));
        S.remove("3");
        System.out.println("after removing 3");
        for(String e: S) System.out.println(e);
    }
}
```



```
1
2
3
4
5
true false
after removing 3
1
2
4
5
```