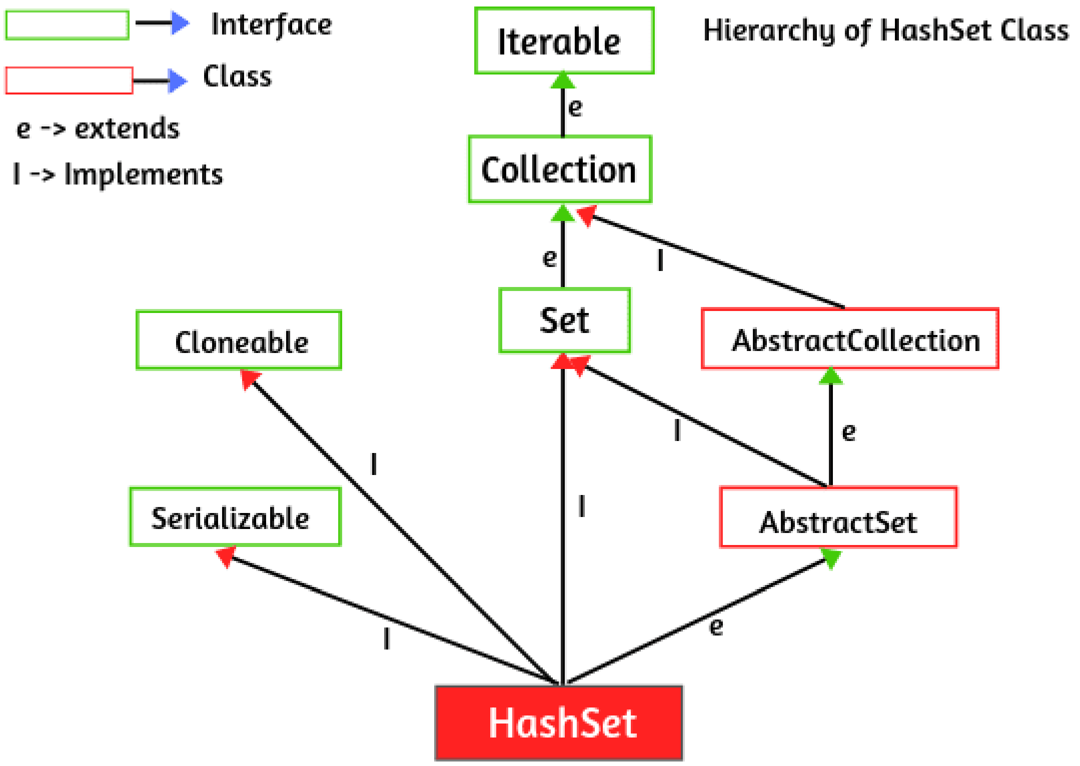
HashSet

HashSet in Java is an unordered collection of elements (objects) that contains only unique elements. That is, it allows duplicate free elements to be stored.

It internally uses Hashtable data structure to store a set of unique elements. It is much faster and
gives constant-time performance for searching and retrieving elements.



```
package hashSetTest;
import java.util.HashSet;
public class HashSetExample1 {
public static void main(String[] args)
{
// Create a HashSet object.
   HashSet<String> set = new HashSet<String>(); // An empty hash set.

// Adding elements to HashSet.
   set.add("First");
   set.add("Second");
   set.add("Third");
   set.add("Fourth");
   set.add("Fifth");
// Adding duplicate elements that will be ignored.
   set.add("First");
   set.add("Third");

// Adding of null elements.
   set.add(null);
   set.add(null); // Ignored.

// Displaying hash set elements.
   System.out.println("Unordered and no duplicate HashSet elements");
   System.out.println(set);
 }
}
```

```
Output:
       Unordered and no duplicate HashSet elements
       [null, Second, Third, First, Fourth, Fifth]
```

```
package hashSetTest;
import java.util.ArrayList;
import java.util.HashSet;
public class HashSetExample2 {
public static void main(String[] args)
{
// Create an ArrayList object.
   ArrayList<String> al = new ArrayList<String>();
    al.add("Monday");
    al.add("Tuesday");
    al.add("Wednesday");
    al.add("Thursday");
    al.add("Friday");

// Adding duplicate elements.
    al.add("Monday");
    al.add("Friday");
    System.out.println("Original elements order: ");
    System.out.println(al);

// Create HashSet object.
   HashSet<String> hset = new HashSet<String>();

// Call addAll() method for adding all elements from existing collection to HashSet.
   hset.addAll(al);
   System.out.println("Unordered HashSet elements without duplicate elements: ");
   System.out.println(hset);
  }
}
```

```
Output:
       Original elements order:
       [Monday, Tuesday, Wednesday, Thursday, Friday, Monday, Friday]
       Unordered HashSet elements without duplicate elements:
       [Monday, Thursday, Friday, Wednesday, Tuesday]
```

```
package hashSetTest;
import java.util.HashSet;
public class HashSetExample3 {
public static void main(String[] args)
{
// Creating a hash set object of type integer.
   HashSet<Integer> hset = new HashSet<Integer>();

// Adding elements to hash set.
   hset.add(5);
   hset.add(10);
   hset.add(15);
   hset.add(20);

// Displaying the hash set elements.
   System.out.println("Initial list of elements");
   System.out.println(hset);

// Removing a specific element from HashSet.
   hset.remove(10);
   System.out.println("List of elements after removing 10");
   System.out.println(hset);

// Creating another HashSet object of type integer and adding elements.
   HashSet<Integer> hset2 = new HashSet<Integer>();
   hset2.add(10);
   hset2.add(25);
// Adding elements of hset2 into hset.
   hset.addAll(hset2);
   System.out.println("List of elements after adding elements from existing collectio
n");
   System.out.println(hset);

// Removing all new elements from HashSet.
   hset.removeAll(hset2);
   System.out.println("List of elements after removing elements from hset2");
   System.out.println(hset);

// Removing all elements available in HashSet.
   clear();
   System.out.println("After invoking clear() method: "+hset);
  }
}
```

```
Output:
      Initial list of elements
      [20, 5, 10, 15]
      List of elements after removing 10
      [20, 5, 15]
      List of elements after adding elements from existing collection
      [20, 5, 25, 10, 15]
      List of elements after removing elements from hset2
      [20, 5, 15]
      After invoking clear() method: [ ]
```

```
package hashSetTest;
import java.util.HashSet;
import java.util.Set;
public class HashSetExample4 {
public static void main(String[] args)
{
   Set<String> pCountry = new HashSet<String>();

// Check that HashSet is empty or not.
   System.out.println("Is popularCountries set empty? : " + pCountry.isEmpty());
   System.out.println("Number of countries in HashSet before adding: " +pCountry.size
());

// Adding elements to hash set.
   pCountry.add("INDIA");
   pCountry.add("USA");
   pCountry.add("UK");
   pCountry.add("FRANCE");

// Find size of HashSet.
   System.out.println("Number of countries in HashSet after adding: " + pCountry.size
());
  }
}
```

```
Output:
      Is popularCountries set empty? : true
      Number of countries in HashSet before adding: 0
      Number of countries in HashSet after adding: 4
```

```
package hashSetTest;
import java.util.HashSet;
import java.util.Set;
public class HashSetExample4 {
public static void main(String[] args)
{
  Set<String> set = new HashSet<String>();
  System.out.println("Is set empty? : " + set.isEmpty());
  System.out.println("Number of elements in HashSet before adding: " +set.size());

  set.add("Dollar");
  set.add("Indian Rupee");
  set.add("Euro");
  set.add("Yen");

  System.out.println("List of Elements in set");
  System.out.println(set);
  System.out.println("Number of elements in the HashSet after adding: " + set.size());

// Call contains() method to check an element exists in set or not.
   if(set.contains("Dollar"))
   {
     System.out.println("Does Element 'Dollar' exist in set?");
     System.out.println("Yes, Element 'Dollar' exists in set");
   }
   else {
       System.out.println("No, Element 'Dollar' does not exist in set");
   }
   System.out.println("Does Element 'Dinar' exist in set?");
   if(set.contains("Dinar"))
   {
      System.out.println("Yes, Element 'Dinar' exists in set ");
   }
   else {
      System.out.println("No, Element 'Dinar' does not exist in set");
   }
  }
}
```

```
Output:
       Is set empty? : true
       Number of elements in HashSet before adding: 0
       List of Elements in set
       [Yen, Dollar, Indian Rupee, Euro]
       Number of elements in HashSet after adding: 4
       Does Element 'Dollar' exist in set?
       Yes, Element 'Dollar' exists in set
       Does Element 'Dinar' exist in the set?
       No, Element 'Dinar' does not exist in the set
```

```
package hashSetTest;
public class Student
{
// Declare instance variables.
   String name, sName;
   int id;
public Student(String name, String sName, int id)
{
  this.name = name;
  this.sName = sName;
  this.id = id;
  }
 }
package hashSetTest;
```

```
import java.util.HashSet;
public class HashSetExample6
{
 public static void main(String[] args)
 {
// Create a user-defined HashSet object of type Student.
   HashSet<Student> hset = new HashSet<Student>();

// Create objects of Student class and pass the parameters to their constructors.
   Student s1 = new Student("John", "RSVM", 0012);
   Student s2 = new Student("Shubh", "DPS", 1234);
   Student s3 = new Student("Ricky", "DAV", 9876);

// Adding elements to HashSet and pass reference variables s1, s2, s3.
    hset.add(s1);
    hset.add(s2);
    hset.add(s3);

// Traversing HashSet.
   for(Student s:hset)
   {
     System.out.println(s.name+" "+s.sName+" "+s.id);
    }
  }
}
```

```
Output:
       Ricky DAV 9876
       John RSVM 10
       Shubh DPS 1234
```