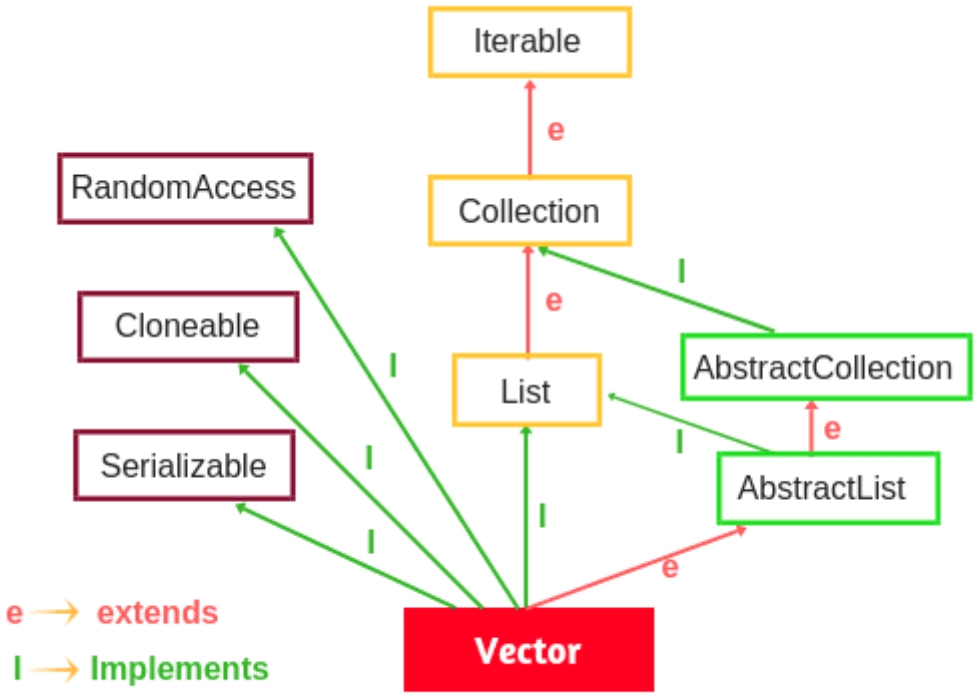
Vector

Vector in Java | Vector Class, Example

Vector class in Java was introduced in JDK 1.0 version. It is present in Java.util package.
It is a dynamically resizable array (growable array) which means it can grow or shrink as required.

Java Vector class is similar to ArrayList class with two main differences.

- Vector is synchronized. It is used for thread safety.
- It contains many legacy methods that are not now a part of the collections framework.




```
package vectorTest;
import java.util.ArrayList;
import java.util.Vector;
public class VectorTest1 {
public static void main(String[] args)
```

```
{
// Create an empty vector object with an initial capacity of 5.
   Vector v = new Vector();

// Check the size of vector before adding elements.
   int size = v.size();
   System.out.println("Size of vector before adding elements: " +size);

// Adding elements to a vector using reference variable v.
    v.add("Red");
    v.add("Green");
    v.add("Yellow");
    v.add("Pink");
    v.add("White");
  System.out.println("Vector elements: " +v);

// Check size and capacity.
   int vectorsize = v.size();
   System.out.println("Size: " +vectorsize);

   int capacity = v.capacity();
   System.out.println("Default capacity: " +capacity);

// Adding elements using addElement() method.
    v.addElement(null);
    v.addElement(20);
    v.add(5, 15);
   int newsize = v.size();
   System.out.println("New size after adding elements: " +newsize);

  int newcapacity = v.capacity();
  System.out.println("New capacity after adding elements: " +newcapacity);
  System.out.println("Elements order after addition: " +v);

// Create an ArrayList with an initial capacity of 10.
   ArrayList al = new ArrayList();

// Adding elements using reference variable al.
    al.add("Brown");
    al.add("Black");

// Call addAll() method to add collection of elements into vector.
    v.addAll(al);

// Now check size and capacity of the vector.
   int vsize = v.size();
   System.out.println("Size: " +vsize);
   int vcapacity = v.capacity();
   System.out.println("Vcapacity: " +vcapacity);

// Adding 11th element to check size and capacity.
   v.add(10);
   System.out.println("Elements: " +v);

   int vecsize = v.size();
   System.out.println("Size after adding 11th element: " +vecsize);

    int cap = v.capacity();
```

```
    System.out.println("Capacity after adding 11th element: " +cap);
  }
}
```

```
Output:
      Size of vector before adding elements: 0
      Vector elements: [Red, Green, Yellow, Pink, White]
      Size: 5
      Default capacity: 10
      New size after adding elements: 8
      New capacity after adding elements: 10
      Elements order after addition: [Red, Green, Yellow, Pink, White, 15, null, 20]
      Size: 10
      Vcapacity: 10
      Elements: [Red, Green, Yellow, Pink, White, 15, null, 20, Brown, Black, 10]
      Size after adding 11th element: 11
      Capacity after adding 11th element: 20
```

```
package vectorTest;
import java.util.Vector;
public class VectorTest2 {
public static void main(String[] args)
{
// Create an empty generic vector with an initial capacity of 10.
   Vector<String> v = new Vector<String>();

// Adding elements to vector.
    v.add("A");
    v.add("B");
    v.add("C");
    v.add("D");
    v.add("E");
  System.out.println("Elements: " +v);

// Call firstElement() method to get the first element using reference variable v.
// Since the return type of firstElement method is String.
// Therefore, we will store it using variable firstElement of data type String.
   String firstElement = v.firstElement();
   System.out.println("First Element: " +firstElement);

   String lastElement = v.lastElement();
   System.out.println("Last Element: " +lastElement);

   String element = v.elementAt(3);
   System.out.println("Element at position 3: " +element);

   boolean checkElement = v.contains("F"); // Return type of contains method is boolea
n.
   System.out.println("Element F: " +checkElement);

// Create an array object with an initial capacity of 5.
   String[] arr = new String[5];
```

```
// Copy collection of elements into an array str.
    v.copyInto(arr);
    System.out.println("Elements in an array arr");
    for(String str:arr)
    {
       System.out.println(str);
    }
// Call clone() method to create a clone of a vector.
   Object obj = v.clone(); // Return type of clone method is an Object.
   System.out.println("Clone of v: " +obj);
  }
}
```

```
Output:
      Elements: [A, B, C, D, E]
      First Element: A
      Last Element: E
      Element at position 3: D
      Element F: false
      Elements in an array arr A B C D E
      Clone of v: [A, B, C, D, E]
```

```
package vectorTest;
import java.util.Vector;
public class VectorTest3 {
public static void main(String[] args)
{
// Creating a vector class object.
  Vector v = new Vector();

// Adding elements to vector.
  v.add("Bat");
  v.add("Ball");
  v.add("Wicket");
  v.add("Stump");
  v.add("Pitch");
  v.add(25);
  v.add(null);

  System.out.println("Elements: " +v);
  Object getElement = v.get(5); // Return type of get method is an Object.
  System.out.println("Element at position 5: " +getElement);

  Integer hashcode = v.hashCode(); // Return type is an Integer.
  System.out.println("HashCode value: " +hashcode);

  Integer indexOfElement = v.indexOf(null);
  System.out.println("Index of element null: " +indexOfElement);

  v.insertElementAt("Gloves", 6);
  v.removeElement(25);
  v.remove(6);
  System.out.println("Elements after removing: " +v);
```

```
 }
}
```

```
Output:
      Elements: [Bat, Ball, Wicket, Stump, Pitch, 25, null]
      Element at position 5: 25
      HashCode value: 461290222
      Index of element null: 6
      Elements after removing: [Bat, Ball, Wicket, Stump, Pitch, Gloves]
```

```
package vectorTest;
import java.util.Vector;
public class VectorTest4
{
public static void main(String[] args)
{
   Vector<String> vec = new Vector<String>();
// Check vector is empty or not.
   boolean check = vec.isEmpty();
   System.out.println("Vector is empty: " +check);

   vec.add("Hydrogen");
   vec.add("Oxygen");
   vec.add("Boron");
   vec.add("Beryllium");
   vec.add("Boron");
   System.out.println("Elements: " +vec);

   boolean check1 = vec.isEmpty();
   System.out.println("Vector is empty: " +check1);

// Replace element oxygen with helium.
   vec.setElementAt("Helium", 1);
   vec.set(2, "Lithium");
   System.out.println("Elements after replacing: " +vec);

// Check size of the vector.
   int size = vec.size();
   System.out.println("Size of the vector: " +size);

// Setting new size of the vector.
   vec.setSize(8);
   System.out.println("Size of the vector after setting: " +vec.size());

// Check capacity of the vector.
   int capacity = vec.capacity();
   System.out.println("Capacity of the vector: " +capacity);

// Check the equality of element carbon.
   boolean equality = vec.equals("Carbon");
   System.out.println("Is Carbon present: " +equality);
  }
}
```

```
Output:
      Vector is empty: true
      Elements: [Hydrogen, Oxygen, Boron, Beryllium, Boron]
      Vector is empty: false
      Elements after replacing: [Hydrogen, Helium, Lithium, Beryllium, Boron]
      Size of the vector: 5
      Size of the vector after setting: 8
      Capacity of the vector: 10
      Is Carbon present: false
```

```
package vectorTest;
import java.util.Vector;
public class VectorTest5 {
public static void main(String[] args)
{
 Vector<Integer> v = new Vector<Integer>();
 for(int i = 0; i < = 10; i++)
 {
  if(i % 2 == 0)
  {
    v.add(i);
    System.out.println(i);
   }
 }
 int size = v.size();
 System.out.println("Size of the vector: " +size);
 int capacity = v.capacity();
 System.out.println("Capacity of the vector: " +capacity);

// Ensuring capacity.
   v.ensureCapacity(25);
// Checking capacity.
   System.out.println("Minimum capacity: " +v.capacity());

// Trim the capacity of the vector to the actual size.
   v.trimToSize();
   System.out.println("Minimum capacity after trimming: " +v.capacity());

// String representation of the vector.
   String str = v.toString();
   System.out.println("String equivalent of the vector: " +str);

// Get elements of vector as an array form.
    v.toArray();
    System.out.println(v);
   }
 }
```

```
Output:
       0 2 4 6 8 10
```

```
Size of the vector: 6
Capacity of the vector: 10
Minimum capacity: 25
Minimum capacity after trimming: 6
String equivalent of the vector: [0, 2, 4, 6, 8, 10]
[0, 2, 4, 6, 8, 10]
```