As you know, an ***AVL tree*** is a binary search tree where each node is balanced in height, that is, where the difference between the height of its right and left subtrees are not more than 1. The aim of this worksheet is to make an implementation of the ***AVL tree*** based on the ***BST class***.

As with the ***BST*** class, the ***AVL*** class uses the generic parameter ***E*** to designate the element type stored at the nodes.

**Part 1**

The generic class AVL< E> extends the BST class and includes the following methods:

```
public class AVL <E extends Comparable<E>> extends BST<E> {
     int balanceFactor(Node<E> node);
     Node<E> rightRotation(Node<E> node);
     Node<E> leftRotation(Node<E> node);
     Node<E> twoRotations(Node<E> node);
     Node<E> balanceNode(Node<E> node);
     void insert(E element);
     void remove(E element);
}
```

1. Complete the generic class AVL< E> implementing all the methods above.
2. Test your implementation running the tests in the AVLTest class.