```
using System;
using System.Collections;
using System.Collections.Generic;

namespace Practica29
{
    class Program
    {
        public class AvlArbol<TKey, TValue> : IEnumerable<TValue>
        {
            private IComparer<TKey> _comparer;
            private AvlNodo _root;

            sealed class AvlNodo
            {
                public AvlNodo Parent;
                public AvlNodo Left;
                public AvlNodo Right;
                public TKey Key;
                public TValue Value;
                public int Balance;
            }

            public AvlArbol(IComparer<TKey> compara)
            {
                _comparer = compara;
            }

            public AvlArbol()
              : this(Comparer<TKey>.Default)
            {

            }

            private AvlNodo RotateLeft(AvlNodo node)
            {
                AvlNodo right = node.Right;
                AvlNodo rightLeft = right.Left;
                AvlNodo parent = node.Parent;

                right.Parent = parent;
                right.Left = node;
                node.Right = rightLeft;
                node.Parent = right;

                if (rightLeft != null)
```

```
    {
        rightLeft.Parent = node;
    }

    if (node == _root)
    {
        _root = right;
    }
    else if (parent.Right == node)
    {
        parent.Right = right;
    }
    else
    {
        parent.Left = right;
    }

    right.Balance++;
    node.Balance = -right.Balance;

    return right;
}

private AvlNodo RotateRightLeft(AvlNodo node)
{
    AvlNodo right = node.Right;
    AvlNodo rightLeft = right.Left;
    AvlNodo parent = node.Parent;
    AvlNodo rightLeftLeft = rightLeft.Left;
    AvlNodo rightLeftRight = rightLeft.Right;

    rightLeft.Parent = parent;
    node.Right = rightLeftLeft;
    right.Left = rightLeftRight;
    rightLeft.Right = right;
    rightLeft.Left = node;
    right.Parent = rightLeft;
    node.Parent = rightLeft;

    if (rightLeftLeft != null)
    {
        rightLeftLeft.Parent = node;
    }

    if (rightLeftRight != null)
    {
```

```
            rightLeftRight.Parent = right;
        }
        if (node == _root)
        {
            _root = rightLeft;
        }
        else if (parent.Right == node)
        {
            parent.Right = rightLeft;
        }
        else
        {
            parent.Left = rightLeft;
        }
        if (rightLeft.Balance == 1)
        {
            node.Balance = 0;
            right.Balance = -1;
        }
        else if (rightLeft.Balance == 0)
        {
            node.Balance = 0;
            right.Balance = 0;
        }
        else
        {
            node.Balance = 1;
            right.Balance = 0;
        }
        rightLeft.Balance = 0;

        return rightLeft;
    }
    private void InsertarBalance(AvlNodo node, int balance)
    {
        while (node != null)
        {
            balance = (node.Balance += balance);

            if (balance == 0)
            {
                return;
            }
            else if (balance == -2)
            {
                if (node.Right.Balance == -1)
```

```
            {
                RotateLeft(node);
            }
            else
            {
                RotateRightLeft(node);
            }
            return;
        }
        AvlNodo parent = node.Parent;

        if (parent != null)
        {
            balance = parent.Left == node ? 1 : -1;
        }
        node = parent;
    }
}
public void Insertar(TKey key, TValue value)
{
    if (_root == null)
    {
        _root = new AvlNodo { Key = key, Value = value };
    }
    else
    {
        AvlNodo node = _root;

        while (node != null)
        {
            int compare = _comparer.Compare(key, node.Key);

            if (compare < 0)
            {
                AvlNodo left = node.Left;
                if (left == null)
                {
                    node.Left = new AvlNodo { Key = key, Value = value, Parent = node };
                    InsertarBalance(node, 1);
                    return;
                }
                else
                {
                    node = left;
                }
            }
```

```
                else if (compare > 0)
                {
                    AvlNodo right = node.Right;

                    if (right == null)
                    {
                        node.Right = new AvlNodo { Key = key, Value = value, Parent = node };

                        InsertarBalance(node, -1);

                        return;
                    }
                    else
                    {
                        node = right;
                    }
                }
                else
                {
                    node.Value = value;

                    return;
                }
            }
        }
        public IEnumerator<TValue> GetEnumerator()
        {
            throw new NotImplementedException();
        }

        IEnumerator IEnumerable.GetEnumerator()
        {
            throw new NotImplementedException();
        }
    }
    static void Main(string[] args)
    {
    }
  }
}
```