

שרשרת דו-כיוונית

```
public class BinList
{
    //--- פעולה הבונה שרשרת דו כיוונית ---
    public static BinNode<Integer> build(int [] arr)
    {
        BinNode<Integer> lst = new BinNode<Integer>(arr[0]);
        BinNode<Integer> pos = lst;

        for(int i = 1 ; i < arr.length ; i++)
        {
            pos.setRight(new BinNode (pos, arr[i], null));
            pos = pos.getRight();
        }
        return lst;
    }

    //--- הצגת השרשרת מהתחלה לסוף ---
    public static void show(BinNode<Integer> lst)
    {
        System.out.print("[");

        while (lst != null)
        {
            System.out.print(lst.getValue());
            if (lst.hasRight())
                System.out.print(", ");
            lst = lst.getRight();
        }
        System.out.println("]");
    }

    //--- הצגת השרשרת מהסוף להתחלה ---
    public static void showBackWord(BinNode<Integer> lst)
    {
        System.out.print("[");
        BinNode<Integer> pos = getLast(lst);

        while (pos != null)
        {
            System.out.print(pos.getValue());
            if (pos.hasLeft())
                System.out.print(", ");
            pos = pos.getLeft();
        }
        System.out.println("]");
    }
}
```

```
//--- מחזירה הפניה לחוליה האחרונה בשרשרת ---
public static BinNode<Integer>getLast(BinNode<Integer> lst)
{
    BinNode<Integer> pos = null;
    while (lst != null)
    {
        pos = lst;
        lst = lst.getRight();
    }
    return pos;
}

//--- פעולה המוסיפה חוליה בתחילת השרשרת ---
public static BinNode<Integer> add (BinNode<Integer>lst, int x)
{
    BinNode<Integer> next = lst;
    lst = new BinNode<Integer>(null, x, next);
    next.setLeft(lst);
    return lst;
}

//--- פעולה המוסיפה חוליה לשרשרת במקום נתון ---
public static BinNode<Integer> insert (BinNode<Integer>lst, BinNode<Integer>pos,int x)
{
    if (pos == null)
        lst = add(lst, x);
    else
    {
        BinNode<Integer>next = pos.getRight();
        BinNode<Integer>temp = new BinNode<Integer>(pos, x, next);
        pos.setRight(temp);
        next.setLeft(temp);
    }
    return lst;
}

//--- פעולה המוסיפה חוליה לשרשרת מחוינת ---
public static BinNode<Integer> insert (BinNode<Integer>lst, int x)
{
    BinNode<Integer>pos = lst;
    while (pos != null && pos.getValue() < x)
    {
        pos = pos.getRight();
    }
    pos = pos.getLeft();

    BinNode<Integer>next = pos.getRight();
    BinNode<Integer>temp = new BinNode<Integer>(pos, x, next);
    pos.setRight(temp);
    next.setLeft(temp);

    return lst;
}

//--- פעולה המחזירה הפניה לחוליה לפי ערך ---
public static BinNode<Integer> getNode (BinNode<Integer>lst, int x)
{
    BinNode<Integer>pos = lst;
    while (pos != null && pos.getValue() != x)
        pos = pos.getRight();
    return pos;
}
}
```