

בניית שרשרת חוליות והדפסתה

(שרשרת חוליות = רשימה מקושרת דינאמית)

I בנייה מהסוף להתחלה

בסיום הפעולה הערך הראשון יהיה בסוף השרשרת והאחרון בתחילתה.

בניית שרשרת (באורך n)

- .1 $lst \leftarrow \text{null}$
- .2 בצע n פעמים:
 - 2.1 קלט x ויצירת חוליה עם x וערך $lst \leftarrow lst$
 - .3 החזר את lst

```
//~~~~~ פעולות לבניית שרשרת ~~~~~  
//--- בניית שרשרת "הפוכה" האיבר האחרון יהי הראשון ---  
public static Node<Integer> build (int n)  
{  
    int x;  
    Node<Integer> lst = null;  
    for (int i = 0 ; i < n ; i++)  
    {  
        x = rnd.nextInt(30)+1; // קלט x  
        lst = new Node<Integer> (x, lst);  
    }  
    return lst;  
}
```

II בנייה מהתחלה לסוף

בסיום הפעולה הערך הראשון יהיה בתחילת השרשרת והאחרון בסופה.

בניית שרשרת (באורך n)

- .1 $lst \leftarrow null$
- .2 $pos \leftarrow null$
- .3 בצע n פעמים:
 - 3.1 קלט x וצור חוליה עם x $temp \leftarrow x$
 - 3.2 אם $(lst == null)$ // זוהי חוליה ראשונה
 - 3.2.1 $lst \leftarrow null$ ועם x ועם null
 - 3.2.2 $pos \leftarrow lst$
 - 3.3 אחרת // זוהי חוליה שאינה ראשונה
 - 3.3.1 עדכן את next של pos לקבל חוליה חדשה עם x ועם null
 - 3.3.3 קדם את pos לחוליה החדשה
- .4 החזר את lst

```

//--- פעולה הבונה ומחזירה שרשרת באורך n ---
//--- סדר הכנסת האיברים לשרשרת הוא מהתחלה לסוף ---
public static Node<Integer> build (int n)
{
    int x;
    Node<Integer> lst = null;
    Node<Integer> pos = null;

    for (int i = 0 ; i < n ; i++)
    {
        x = rnd.nextInt(30)+1; // קלט x
        if (lst == null) // הכנסת חוליה ראשונה לשרשרת
        {
            lst = new Node<Integer> (x);
            pos = lst;
        }
        else // הכנסת חוליה שאינה ראשונה
        {
            pos.setNext(new Node<Integer> (x));
            pos = pos.getNext();
        }
    }
    return lst;
}

```

III הדפסת איברי השרשרת

כשסורקים שרשרת מותר לשנות את ערך ההפנייה לתחילת השרשרת, כי קיימת הפנייה לתחילת השרשרת מהפעולה המזמנת.

```
public static int showChain (Node<Integer> lst)
{
    :
    while (lst != null)
    {
        : // lst.getValue() בצע משהו עם/על
        lst = lst.getNext();
    }
    return ...
}
```

הדפסת איברי השרשרת:

```
//--- פעולה המציגה את השרשרת ---
public static void show (Node<Integer> chain)
{
    System.out.print ("[" );
    while (chain != null)
    {
        System.out.print (chain.getValue ().toString ());
        if (chain.hasNext ())
            System.out.print (", ");
        else
            System.out.println ("]");
        chain = chain.getNext ();
    }
}
```

פלט הפעולה יהיה השרשרת במבנה: $[x_1, x_2, \dots, x_n]$

כך ש- x_1 הוא האיבר שבראש השרשרת ו- x_n בסופה.