פעולה המקבלת עץ בינארי ומחזירה מבנה נתונים

תרגיל

1. כתוב פעולה המקבלת כפרמטר עץ בינארי ומחזירה **תור** שכל איבר בו ...
2. כתוב פעולה המקבלת כפרמטר עץ בינארי ומחזירה **מחסנית** שכל איבר בה ...
3. כתוב פעולה המקבלת כפרמטר עץ בינארי ומחזירה **רשימה** שכל איבר בה ...

כשמקבלים תרגילים מסוג זה, הפרמר היחיד הוא העץ הבינארי, והפעולה צריכה להחזיר את מבנה הנתונים.

עץ בינארי נסרק בדרך כלל בפעולה רקורסיבית, אם ניצור בכל זימון רקורסיבי את מבנה הנתונים החדש, נצטרך לעבוד קשה (עד כדי בלתי אפשרי) כדי לאחד את כל המבנים שנוצרו בכל זימון רקורסיבי למבנה אחד.

הפתרון - פעולה עוטפת

נכתוב פעולה המקבלת כפרמטר את העץ הבינארי, ומבצעת 3 פעולות:

(1) יצירת מבנה הנתונים להחזרה
(2) זימון פעולת עזר שתקבל את העץ ואת מבנה הנתונים ותבצע את הנדרש תוך מילוי מבנה הנתונים בערכים
(3) החזרת מבנה הנתונים

שימו לב: שם פעולת העזר יכול להיות זהה לשם הפעולה העוטפת (העמסת פעולות - אותו שם פרמטרים שונים).

נבחין בין מקרה שבו יש להחזיר מבנה נתונים **שקיימת עבורו מחלקה** המגדירה ומטפלת בו (תור או מחסנית) למקרה שבו יש להחזיר מבנה נתונים שאין עבורו מחלקה (רשימה, עץ בינארי אחר).

I **החזרת תור** (או מחסנית) - פעולת העזר תחזיר ערך void:

```
//--- הפעולה העוטפת ---
public static Queue<Integer> doSomthing (BinNode<Integer> bt)
{
	Queue<Integer> que = new Queue<Integer>();
	doSomthing (bt, que);		//	זימון פעולה המחזירה ערך void
	return que;
}

//--- פעולת העזר ---
public static void doSomthing (BinNode<Integer> bt, Queue<Integer> que)
{
	if (bt != null)
	{
		פעולת חישוב על העץ המוסיפה איבר לתור
		זימון הפעולה על בן שמאלי והתור
		זימון הפעולה על בן ימני והתור
	}
}
```

II **החזרת רשימה** - פעולת העזר מחזירה את הרשימה:

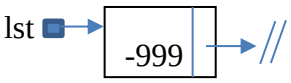
```
//--- הפעולה העוטפת ---
public static Node<Integer> doSomthing (BinNode<Integer> bt)
{
	Node<Integer> lst = null;
	lst = doSomthing (bt, lst);		//	זימון פעולה המחזירה רשימה
	return lst;
}

//--- פעולת העזר ---
public static Node<Integer>  doSomthing (BinNode<Integer> bt, Node<Integer> lst)
{
	if (bt != null)
	{
		int x = ערך מחושב על הצומת בעץ
		lst = new Node<Integer> (x, lst);	//	הכנסת הערך לתחילת הרשימה
		lst = זימון הפעולה על בן שמאלי והרשימה
		lst = זימון הפעולה על בן ימני והרשימה
	}
	return lst;
}
```

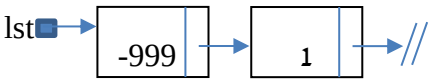
פתרון אפשרי אחר (דוגמת בגרות 2017):
ניצור בפעולה העוטפת חוליית דֶמֶה.
בתוך פעולת העזר נוסיף את האיבר החדש אחרי איבר הדמה
בסיום, בפעולה העוטפת נחזיר את lst.getNext()

מכיוון שלעולם איננו מעדכנים את האיבר הראשון ברשימה, פעולת העזר תחזיר ערך void כי ההוספה נעשית על הרשימה שהתקבלה כפרמטר.

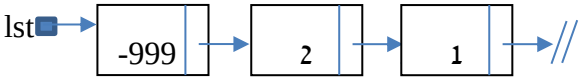
```
//---  הפעולה העוטפת ---
public static Node<Integer> doSomthing (BinNode<Integer> bt)
{
        Node<Integer> lst = new Node<Integer> (-999) ;     //        יצירת חוליית הדמה
        doSomthing (bt, lst);                              //  זימון פעולה המחזירה רשימה
        return lst.getNext();
}
//--- פעולת העזר ---
public static void  doSomthing (BinNode<Integer> bt, Node<Integer> lst)
{
        if (bt != null)
        {
                int x = ערך מחושב על הצומת בעץ
                lst.setNext(new Node<Integer> (x, lst));        //  הכנסת הערך אחרי חולית הדמה
                זימון הפעולה על בן שמאלי / בן ימני והרשימה
        }
}
```



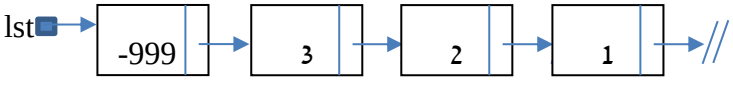
הרשימה שהתקבלה מהפעולה העוטפת



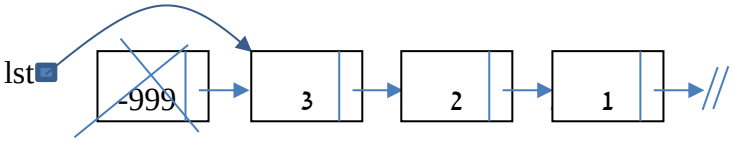
הרשימה אחרי הוספת הערך 1



הרשימה אחרי הוספת הערך 2



הרשימה אחרי הוספת הערך 3



הרשימה המוחזרת