

רקורסיה ברשימות

רשימות מקושרות מתאימות מאד לשימוש ברקורסיה.
ההגדרה של צומת ברשימה היא רקורסיבית:

```
public class IntNode
{
    private int _value;
    private IntNode _next;
}
```

כל צומת מורכב מנתונים ועוד שדה שהוא בעצמו צומת = רקורסיה.

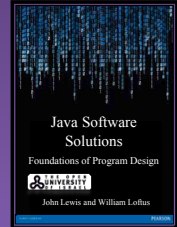
2 רקורסיה ברשימות

מבוא למדעי המחשב ושפת Java

11.4: רקורסיה ברשימות

מרצה: תמר וילנר

האוניברסיטה הפתוחה



הדפסת הרשימה ברקורסיה

```
public void printRec () {
    System.out.println ("The list is: ");
    printRec(_head);
}

private void printRec (IntNode p) {
    if (p == null) {
        System.out.println (" null ");
        return;
    }
    else {
        System.out.print (p.getValue() + " --> ");
        printRec (p.getNext());
    }
}
```

4 רקורסיה ברשימות

אורך הרשימה - ברקורסיה

```
public int lengthRec ()
{
    return lengthRec(_head);
}

private int lengthRec (IntNode p)
{
    if (p == null)
        return 0;
    return 1 + lengthRec (p.getNext());
}
```

Diagram illustrating the recursive calculation of list length:

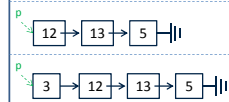
- Initial list: 3 → 2 → 8 → -5 → null
- Step 1: 1 + (length of 2 → 8 → -5 → null)
- Step 2: 1 + 1 + (length of 8 → -5 → null)
- Step 3: 1 + 1 + 1 + (length of -5 → null)
- Step 4: 1 + 1 + 1 + 1 + 0 = 4

3 רקורסיה ברשימות

הדפסת רשימה ברקורסיה – סימולציה

```
private void printRec (IntNode p) {
    if (p == null) {
        System.out.println (" null ");
        return;
    }
    else {
        System.out.print (p.getValue() + " --> ");
        printRec (p.getNext());
    }
}
```

מחסנית הזיכרון:



הפלט:

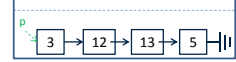
3 --> 12 -->

6 רקורסיה ברשימות

הדפסת רשימה ברקורסיה - סימולציה

```
private void printRec (IntNode p) {
    if (p == null) {
        System.out.println (" null ");
        return;
    }
    else {
        System.out.print (p.getValue() + " --> ");
        printRec (p.getNext());
    }
}
```

מחסנית הזיכרון:



הפלט:

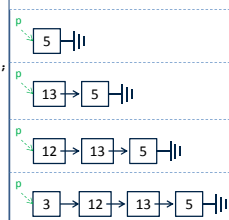
3 -->

5 רקורסיה ברשימות

הדפסת רשימה ברקורסיה – סימולציה

```
private void printRec (IntNode p) {
    if (p == null) {
        System.out.println (" null ");
        return;
    }
    else {
        System.out.print (p.getValue() + " --> ");
        printRec (p.getNext());
    }
}
```

מחסנית הזיכרון:



הפלט:

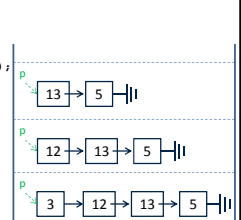
3 --> 12 --> 13 --> 5 -->

רקורסיה ברשימות 8

הדפסת רשימה ברקורסיה – סימולציה

```
private void printRec (IntNode p) {
    if (p == null) {
        System.out.println (" null ");
        return;
    }
    else {
        System.out.print (p.getValue() + " --> ");
        printRec (p.getNext());
    }
}
```

מחסנית הזיכרון:



הפלט:

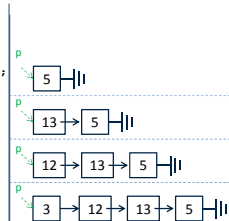
3 --> 12 --> 13 -->

רקורסיה ברשימות 7

הדפסת רשימה ברקורסיה – סימולציה

```
private void printRec (IntNode p) {
    if (p == null) {
        System.out.println (" null ");
        return;
    }
    else {
        System.out.print (p.getValue() + " --> ");
        printRec (p.getNext());
    }
}
```

מחסנית הזיכרון:



הפלט:

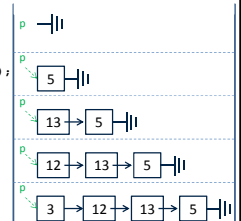
3 --> 12 --> 13 --> 5 --> null

רקורסיה ברשימות 10

הדפסת רשימה ברקורסיה – סימולציה

```
private void printRec (IntNode p) {
    if (p == null) {
        System.out.println (" null ");
        return;
    }
    else {
        System.out.print (p.getValue() + " --> ");
        printRec (p.getNext());
    }
}
```

מחסנית הזיכרון:



הפלט:

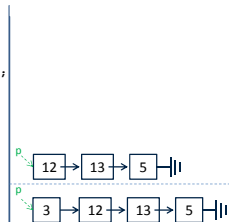
3 --> 12 --> 13 --> 5 --> null

רקורסיה ברשימות 9

הדפסת רשימה ברקורסיה – סימולציה

```
private void printRec (IntNode p) {
    if (p == null) {
        System.out.println (" null ");
        return;
    }
    else {
        System.out.print (p.getValue() + " --> ");
        printRec (p.getNext());
    }
}
```

מחסנית הזיכרון:



הפלט:

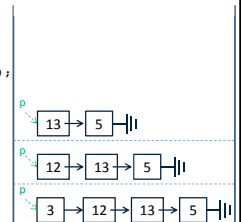
3 --> 12 --> 13 --> 5 --> null

רקורסיה ברשימות 12

הדפסת רשימה ברקורסיה – סימולציה

```
private void printRec (IntNode p) {
    if (p == null) {
        System.out.println (" null ");
        return;
    }
    else {
        System.out.print (p.getValue() + " --> ");
        printRec (p.getNext());
    }
}
```

מחסנית הזיכרון:



הפלט:

3 --> 12 --> 13 --> 5 --> null

רקורסיה ברשימות 11

הוספת איבר לסוף הרשימה

```
public void add(int num) {
    _head = add(_head, num);
}

private IntNode add(IntNode node, int num) {
    if (node == null) {
        return new IntNode(num, null);
    }
    else {
        node.setNext(add(node.getNext(), num));
        return node;
    }
}
```

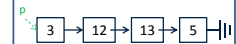
14 רקורסיה ברשימות

הדפסת רשימה ברקורסיה – סימולציה

```
private void printRec (IntNode p) {
    if (p == null) {
        System.out.println (" null ");
        return;
    }
    else {
        System.out.print (p.getValue() + " --> ");
        printRec (p.getNext());
    }
}
```

מחסנית הזיכרון:

הפלט:
3 --> 12 --> 13 --> 5 --> null



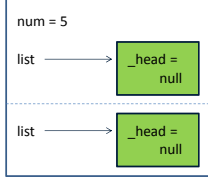
13 רקורסיה ברשימות

הוספת איבר לסוף הרשימה - סימולציה

```
public class Driver
{
    public static void main(String args)
    {
        IntList list = new IntList();
        list.add(5);
        list.add(8);
        list.add(7);
    }
}

:IntList במחלקה
public void add(int num) {
    _head = add(_head, num);
}
```

מחסנית הזיכרון:



16 רקורסיה ברשימות

הוספת איבר לסוף הרשימה - סימולציה

```
public class Driver
{
    public static void main(String args)
    {
        IntList list = new IntList();
        list.add(5);
        list.add(8);
        list.add(7);
    }
}

:IntList במחלקה
public void add(int num) {
    _head = add(_head, num);
}
```

מחסנית הזיכרון:



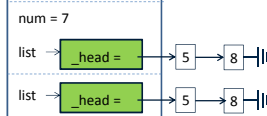
15 רקורסיה ברשימות

הוספת איבר לסוף הרשימה - סימולציה

```
public class Driver
{
    public static void main(String args)
    {
        IntList list = new IntList();
        list.add(5);
        list.add(8);
        list.add(7);
    }
}

:IntList במחלקה
public void add(int num) {
    _head = add(_head, num);
}
```

מחסנית הזיכרון:

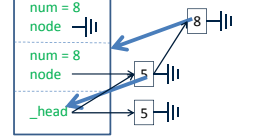


18 רקורסיה ברשימות

הוספת איבר לסוף הרשימה - סימולציה

```
private IntNode add(IntNode node, int num)
{
    if (node == null) {
        return new IntNode(num, null);
    }
    else {
        node.setNext(add(node.getNext(), num));
        return node;
    }
}
```

מחסנית הזיכרון:



_head = add(_head, 8);

17 רקורסיה ברשימות

הוספה למקום הנכון – רשימה ממוינת

```
public IntNode addInOrder (int num)
{
    return addInOrder (_head, num);
}

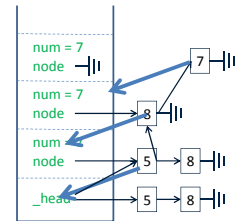
private IntNode addInOrder (IntNode p, int num)
{
    if (p == null)
        return new IntNode (num);
    if (p.getValue() > num)
        return new IntNode(num, p);
    p.setNext(addInOrder (p.getNext(), num));
    return p;
}
```

20 רקורסיה ברשימות

הוספת איבר לסוף הרשימה - סימולציה

```
private IntNode add(IntNode node, int num)
{
    if (node == null) {
        return new IntNode(num, null);
    }
    else {
        node.setNext(add(node.getNext(), num));
        return node;
    }
}

_head = add (_head, 7);
```

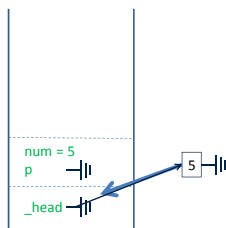


19 רקורסיה ברשימות

הוספת איבר לרשימה ממוינת - סימולציה

```
private IntNode addInOrder (IntNode p, int num)
{
    if (p == null)
        return new IntNode (num);
    if (p.getValue() > num)
        return new IntNode(num, p);
    p.setNext(addInOrder (p.getNext(), num));
    return p;
}

return addInOrder (_head, 5);
```

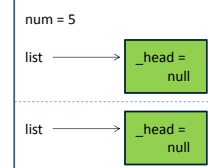


22 רקורסיה ברשימות

הוספת איבר לרשימה ממוינת - סימולציה

```
public static void main(String args)
{
    IntList list = new IntList();
    list._head = list.addInOrder(5);
    list._head = list.addInOrder(3);
    list._head = list.addInOrder(7);
}

public IntNode addInOrder(int num) {
    return addInOrder(_head, num);
}
```

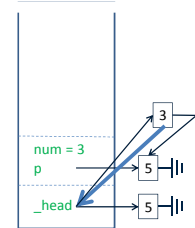


21 רקורסיה ברשימות

הוספת איבר לרשימה ממוינת - סימולציה

```
private IntNode addInOrder (IntNode p, int num)
{
    if (p == null)
        return new IntNode (num);
    if (p.getValue() > num)
        return new IntNode(num, p);
    p.setNext(addInOrder (p.getNext(), num));
    return p;
}

return addInOrder (_head, 3);
```

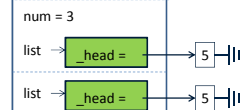


24 רקורסיה ברשימות

הוספת איבר לסוף הרשימה - סימולציה

```
public static void main(String args)
{
    IntList list = new IntList();
    list._head = list.addInOrder(5);
    list._head = list.addInOrder(3);
    list._head = list.addInOrder(7);
}

return addInOrder(_head, num);
```



23 רקורסיה ברשימות

הוספת איבר לרשימה ממוינת - סימולציה

ערימה:

```
private IntNode addInOrder (IntNode p, int num)
{
    if (p == null)
        return new IntNode (num);
    if (p.getValue() > num)
        return new IntNode (num, p);
    p.setNext (addInOrder (p.getNext(), num));
    return p;
}
```

מחסנית הזיכרון:

26 רקורסיה ברשימות

הוספת איבר לסוף הרשימה - סימולציה

ערימה:

```
public static void main (String args)
{
    IntList list = new IntList ();
    list._head = list.addInOrder (5);
    list._head = list.addInOrder (3);
    list._head = list.addInOrder (7);
}
```

מחסנית הזיכרון:

25 רקורסיה ברשימות

ה- main יראה כך:

```
public static void main (String args)
{
    IntList list = new IntList ();
    list.addInOrder (5);
    list.addInOrder (3);
    list.addInOrder (7);
}
```

28 רקורסיה ברשימות

השיטה הציבורית addInOrder

השיטה שכתבנו במחלקה IntList:

```
public IntNode addInOrder (int num)
{
    return addInOrder (_head, num);
}
```

כדאי להשתמש בשיטה שלא מסגירה את המימוש של הרשימה:

```
public void addInOrder (int num)
{
    _head = addInOrder (_head, num);
}
```

27 רקורסיה ברשימות