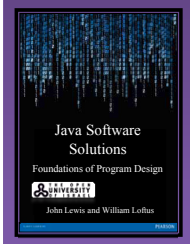
מבוא למדעי המחשב ושפת Java

11.2: רשימות מקושרות

מרצה: תמר וילנר



מגבלה

החסרון העיקרי של ייצוג רשימה באמצעות מערך הוא מגבלת האחסון.

אי אפשר לשנות את גודל המערך במהלך ריצת התכנית.

מה יקרה אם יאזלו התאים הפנויים במערך?

או אם בזמן ריצת התכנית יאוחסנו במערך רק רשימות מאד קטנות והמקום שהוקצה למערך הוא הרבה יותר גדול? זה בזבוז רב של זיכרון.

הפתרון - הקצאה דינמית של זיכרון

במהלך התכנית אנחנו יכולים להקצות אובייקטים חדשים. בכל פעם ליצור אובייקט אחד, לפי הצורך.

אובייקט יכיל התייחסות (הצבעה) לאובייקט אחר.

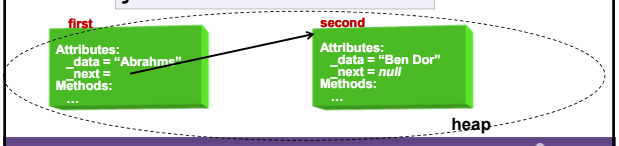
ההתייחסות הזו היא הכתובת של האובייקט בערימה (heap).

כל איבר כזה יהיה **חוליה** (link) ברשימה ויוגדר כך:

```
class LinkObject {
        String   _data;
        LinkObject  _next;
}
```


LinkObject

```
class LinkObject {
      String       _data;
      LinkObject _next;
      …
}
```



נניח שהרשימה היא של מספרים שלמים

המחלקה שתייצג צומת (node) אחד ברשימה תהיה IntNode:

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

המחלקה IntNode

```
public IntNode(int val)
{
     _value = val;
     _next = null;
}

public IntNode(int val, IntNode next)
{
     _value = val;
     _next = next;
}
```

שיטות המחלקה IntNode

```
    public int getValue()  {
        return _value;
    }
    public void setValue(int v)  {
        _value = v;
    }

    public IntNode getNext( )  {
        return _next;
    }
    public void setNext(IntNode node)  {
       _next = node;
    }
}  //end of class IntNode
```


המחלקה IntList

```
public class IntList
{
    private IntNode _head;

    public IntList( ) {
        _head = null;
    }

    public IntList(IntNode node) {
        _head = node;
    }
```

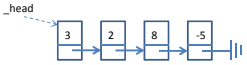

המחלקה IntList - המשך

```
public boolean empty( ) {
        return _head == null;
}

public IntNode nextElement (IntNode node)  {
        return node.getNext();
}

public int getValueOfNode (IntNode node)  {
        return node.getValue();
}
```


הוספת צומת לרשימה



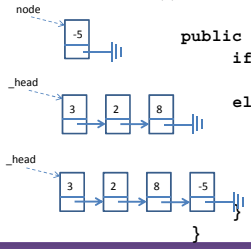
דרכים שונות:

1. להוסיף לתחילת הרשימה
2. להוסיף לסוף הרשימה
3. להוסיף אחרי צומת מסוים
4. להוסיף למקום מסוים (מספרי)

החלטה נוספת – האם הפרמטר המתקבל לשיטה הוא המספר שצריך להוסיף או הצומת שמאחסן את המספר?

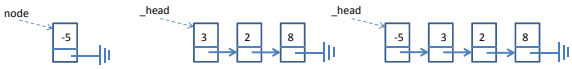
שיטת הוספת צומת לסוף הרשימה

```
// add a link at the end of the list
public void addToEnd(IntNode node) {
    if (empty( ))
        _head = node;
    else {
        IntNode ptr = _head;
        while (ptr.getNext( ) != null)
            ptr = ptr.getNext( );
        ptr.setNext(node);
    }
}
```



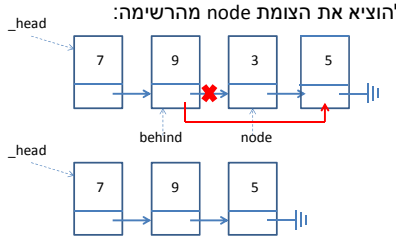
שיטת הוספת צומת לתחילת הרשימה

```
// add a link to the beginning of the list
public void addToHead (IntNode node)
{
    IntNode temp = _head;
    _head = node;
    node.setNext(temp);
}
```



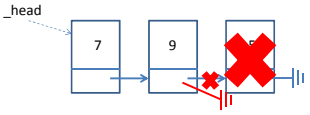
הוצאת צומת מרשימה

רוצים להוציא את הצומת node מהרשימה:



הוצאת צומת מרשימה – הקוד:

```
public void remove (int num)
{
    IntNode behind = _head;
    while (behind.getNext().getValue() != num)
       behind = behind.getNext();
    behind.setNext(behind.getNext().getNext());
}
```



הוצאת איבר מרשימה – יותר טוב...

```
public void remove (int num)
{
   if (_head != null)
   {
       IntNode behind = _head;
       while (behind.getNext().getValue() != num)
          behind = behind.getNext();
       behind.setNext(behind.getNext().getNext());
   }
}
```


הוצאת איבר מרשימה – עוד יותר טוב

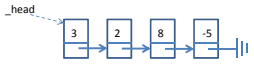
```
public void remove (int num) {
   if (_head != null)  {
       if (_head.getValue() == num)
          _head = _head.getNext();
       else  {
          IntNode behind = _head;
          while (behind.getNext().getValue() != num)
              behind = behind.getNext();
          behind.setNext(behind.getNext().getNext());
       }
   }
}
```


הוצאת צומת מרשימה - סופי

```
public void remove (int num) {
    if (_head != null)  {
       if (_head.getValue() == num)
          _head = _head.getNext();
          else  {
             IntNode behind = _head;
             while (behind.getNext() != null)  {
                 if (behind.getNext().getValue() == num) {
                    behind.setNext(behind.getNext().getNext());
                    return;
                 }
                 else
                    behind = behind.getNext();
             }  //of while
          } //of else (if num is not in _head)
    }  // of if (the list is not empty)
}  //of the method
```


הדפסת הרשימה

```
public void printList()
{
   IntNode temp = _head;
   while (temp != null)
   {
       System.out.print (temp.getValue() + " --> ");
       temp = temp.getNext();
   }
   System.out.println (" null");
}
```



3 --> 2 --> 8 --> -5 --> null

אורך הרשימה

```
public int length()
{
    IntNode temp = _head;
    int count = 0;
    while (temp != null)
    {
        count++;
        temp = temp.getNext();
    }
    return count;
}
```


מציאת איבר קודם לאיבר נתון

```
public IntNode predecessor (IntNode node)
{
    if (_head == null || _head == node)
        return null;
    IntNode behind = _head;
    while (behind.getNext() != null)
    {
        if (behind.getNext() == node)
            return behind;
        else
            behind = behind.getNext();
    }
    return null;
}
```