

# מטלת מנחה (ממ"ן) 18

הקורס: 20441 - מבוא למדעי המחשב ושפת Java

חומר הלימוד למטלה: יחידות 19 - 20 נושא המטלה: מחסניות, תורים, עצים

בינריים וחישוביות

משקל המטלה: 4 נקודות

מספר השאלות: 3

מועד אחרון להגשה: 3.7.2009

סמסטר: 2009ב

(ת)

שאלה 1 – לא להרצה (20%)

המחלקות IntNode ו- IntList שלהלן מממשות רשימה מקושרת. כל מחלקה בקובץ נפרד, כמובן.

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

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

    public IntNode getNext()      { return _next; }
    public void setNext(IntNode node) { _next = node; }
    public int getValue()         { return _value; }
    public void setValue(int v)   { _value = v; }
}

//-----
public class IntList
{
    private IntNode _head;

    public IntList() { _head = null; }

    /**
     * Adds an item to the beginning of the list
     * @param val the value of the node to insert to the list
     */
    public void addHead(int val)
    {
        _head = new IntNode(val, _head);
    }
}
```

```

/**
 * Returns the string representation of the linked list
 * @return String representing the Linked List
 */
public String toString()
{
    String str = "";
    for (IntNode tmp=_head; tmp!=null; tmp=tmp.getNext())
        str += tmp.getValue() + "\t";
    return str;
}
}

```

המחלקה Node שלהלן מממשת צומת בעץ בינרי.

```

public class Node
{
    private int _number;
    private Node _leftSon, _rightSon;

    public Node (int number) {
        _number = number;
        _leftSon = null;
        _rightSon = null;
    }

    public int  getNumber()      {return _number; }
    public Node getLeftSon()     {return _leftSon; }
    public Node getRightSon()    {return _rightSon; }
}

```

המחלקה BinaryTree מאגדת בתוכה שיטות סטטיות לטיפול בעץ בינרי. אחת השיטות במחלקה היא השיטה what שלהלן (שמשתמשת בשיטה הפרטית what שכתובה אחריה) :

```

public static IntList what (Node root, int i)
{
    IntList level = new IntList();
    if (root != null)
        what (level, i, root);
    return level;
}

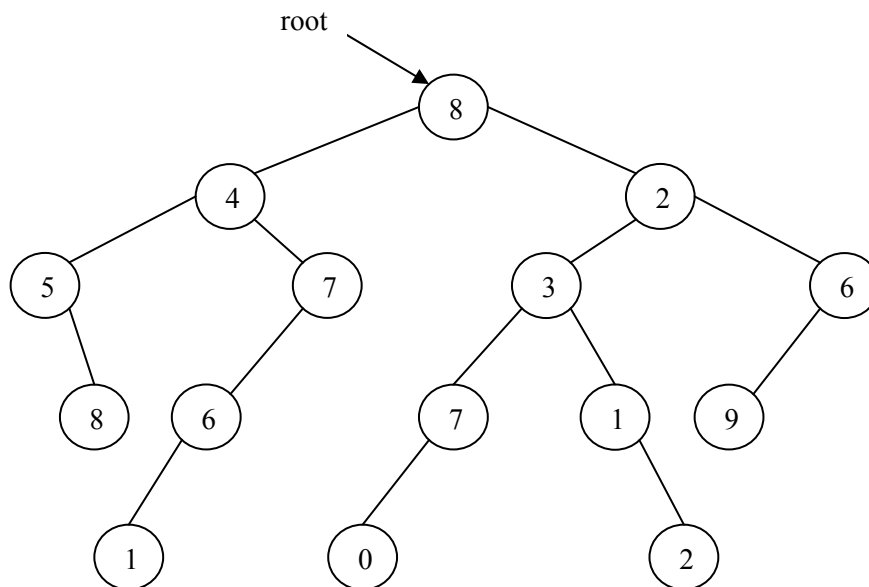
```

```

private static void what (IntList level, int i, Node1 t)
{
    if (t != null)
    {
        if (i>0)
        {
            what (level, i-1, t.getRightSon());
            what (level, i-1, t.getLeftSon());
        }
        else
            level.addHead(t.getNumber());
    }
}

```

(1) בהינתן העץ הבינרי הבא:



מה יודפס כתוצאה מהפקודות הבאות:

```

IntList list = BinaryTree.what(root, 3);
System.out.println (list);

```

(2) מה מבצעת השיטה what באופן כללי, כאשר היא מקבלת שורש לעץ בינרי, ומספר שלם כלשהו? שימו לב למקרה קצה (5 נקודות).

## שאלה 2 - להרצה (30%)

הוסיפו למחלקה BinaryTree שיטה הסוכמת (מלשון סכום) את כל האיברים בעץ בינרי, שערכם המספרי גדול (ממש) ממספר נתון. חתימת השיטה היא:

```
public static int sumOfBiggerThan(Node node, int n)
```

השיטה מקבלת קודקוד node שהוא שורש העץ (אובייקט זה הוא מהמחלקה Node שהוגדרה בשאלה 1), ומספר נתון n, ומחזירה את סכום כל האיברים בעץ שערכם גדול (ממש מ-n). הסבירו את המימוש שלכם.

## שאלה 3 - להרצה (50%)

בשאלה זו עליכם לממש את המחלקה Stack ולהשתמש בה.

המחלקה Stack מממשת **מחסנית** לאחסון מספרים שלמים.

במחסנית מתבצעים התהליכים של הכנסת איברים והוצאתם בקצה אחד בלבד הנקרא ראש המחסנית. הטיפול באיברים דרך פתח יחיד זה יוצר מצב שבו איבר ש נכנס אחרון יוצא ראשון (Last In First Out).

הפעולות המוגדרות על המחסנית הן:

- הבנאי (constructor), יוצר מחסנית ריקה.  
`public Stack()`
- פעולת הדחיפה - push, המקבלת איבר x ומכניסה את האיבר x לראש המחסנית. למחסנית אין הגבלה על הגודל ולכן אי אפשר לגרום לגלישה (overflow) בשל הכנסה של יותר מדי איברים.  
`public void push(int x)`
- פעולת השליפה - pop, המוציאה את האיבר שבראש המחסנית ומחזירה את ערכו. הפעולה לא מוגדרת אם המחסנית ריקה.  
`public int pop()`
- פעולת ההצצה לראש המחסנית - top, המחזירה את ערכו של האיבר שבראש המחסנית, בלי להוציאו משם. גם פעולה זו לא מוגדרת אם המחסנית ריקה.  
`public int top()`
- פעולה הבודקת אם המחסנית ריקה - empty, המחזירה true אם היא ריקה ו- false אחרת.  
`public boolean empty()`

מימוש המחלקה ייעשה בדומה לנעשה בהרצאותיו של פרופ' רוזנשיין. המימוש ישתמש במחלקה IntNode המתוארת בשאלה 1.

שימו לב שאסור להשתמש במערכים!

בנוסף עליכם לכתוב מחלקה בשם StackSorter המשתמשת במחלקה Stack. במחלקה זו עליכם לכתוב שיטה סטטית אחת המקבלת כפרמטר מחסנית וממיינת את המספרים הנמצאים בה. בסיום השיטה, המחסנית s צריכה להיות מלאה באותם מספרים אבל ממוינים בסדר עולה (האיבר שבראש המחסנית הוא הקטן ביותר). חתימת השיטה תהיה:

```
public static void sortStack(Stack s)
```

**בשיטה זו אתם יכולים להשתמש בשתי מחסניות עזר בלבד. אין צורך לכתוב מיון בסיבוכיות של  $O(n \log n)$ .**

**שימו לב שהשיטה יכולה להשתמש אך ורק במחלקה Stack ובשיטות שמוגדרות בה.**

אתם יכולים להוסיף למחלקה שיטות פרטיות כרצונכם (כמובן).

מה סיבוכיות הזמן ומה סיבוכיות המקום של השיטה שכתבתם? (5 נקודות)

#### הגשה

1. הגשת הממ"ן נעשית בצורה אלקטרונית בלבד, דרך מערכת שליחת המטלות.
2. את התשובה לשאלה 1 יש להגיש בקובץ Word יחיד בשם ex18.doc.
3. את התשובה לשאלה 2 יש להגיש בקובץ BinaryTree.java.
4. את התשובה לשאלה 3 יש להגיש בקובץ Stack.java, StackSorter.java.
5. יש לארוז את הקבצים בתוך קובץ zip יחיד. אין לשלוח קבצים נוספים.

#### בהצלחה