

פרק 8 - ממשקים

I : הממשק Comparable

```
public interface Comparable
{
    public static int maxItem = 10;

    public boolean isLessThan (Object o);
    public boolean isEqualTo (Object o);
}
```

הממשק מכיל רק כותרות הפעולות.
כותרת הפעולה מסתיימת ב- ;

מחלקה שתממש את הממשק, תכיר גם את הקבועים המוגדרים בממשק.

II : המחלקה Date מממשת את הממשק:

```
/**
 *      Class Date - by Hila Kadman
 *      this class implements Comparable
 */
public class Date implements Comparable
{
    private int dd;           // day
    private int mm;           // month
    private int yy;           // year

    public Date (int day, int month, int year){
        this.dd = day;
        this.mm = month;
        this.yy = year;
    }

    public int getDay ()      { return this.dd; }
    public int getMonth ()   { return this.mm; }
    public int getYear ()    { return this.yy; }
    public void setDay (int date) { this.dd = date; }
    public void setMonth (int month) { this.mm = month; }
    public void setYear (int year) { this.yy = year; }
    public String toString(){
        return this.dd+"/"+this.mm+"/"+this.yy;
    }
    //--- isEqualTo : מיושם פעולת המחשק ---
    public boolean isEqualTo (Object o)
    {
        Date d2 = (Date)o; // המרה כלפי מטה
        return (d2.dd == this.dd) && (d2.mm == this.mm) && (d2.yy == this.yy);
    }
    //--- isLessThan : מיושם פעולת המחשק ---
    public boolean isLessThan (Object o)
    {
        Date d2 = (Date)o; // המרה כלפי מטה
        if (this.yy > d2.yy)
            return false ;
        if (this.yy == d2.yy)
            if (this.mm > d2.mm || this.mm == d2.mm && this.dd >= d2.dd)
                return false;
        return true ;
    }
}
```

המחלקה Date מתחייבת לממש את כל פעולות הממשק

הפעולה מקבלת אובייקט מסוג Object ומבצעת עליו המרה כלפי מטה

III המחלקה Sorter המבצעת מיון בין שני טיפוסים הניתנים להשוואה, ובתנאי שהמחלקה שלהם מממשת את הממשק Comparable :

המיון הוא מיון בחירה (מציאת הערך הקטן ביותר במערך והחלפתו בערך שבמקום ה-i)

```
public class Sorter
{
    //--- sorts an array of integers from top to bottom ---
    public static void sort(Comparable[] arr)
    {
        int minIndex;

        for (int i = 0; i < arr.length-1; i++)
        {
            minIndex = findMinIndex (arr, i);
            swap (arr, minIndex, i);
        }

        //--- swaps the arr[x] with arr[y] ---
        private static void swap (Comparable[] arr, int x, int y)
        {
            Comparable temp = arr[x];
            arr[x] = arr[y];
            arr[y] = temp;
        }

        //search for smallest number in the unsorted part of array:
        private static int findMinIndex (Comparable[] arr, int from)
        {
            Comparable min = arr[from];
            int minIndex = from;
            for (int i = from; i < arr.length; i++)
            {
                if (arr[i].isLessThan(min))
                {
                    min = arr[i];
                    minIndex = i;
                }
            }
            return minIndex;
        }
    }
}
```

המחלקה Sorter ממיינת מערך
מטיפוס בר-השוואה Comparable

תכנית המגדירה מערך של תאריכים וממיינת אותם: IV

```
public class ChkInterface
{
    public static void main (String[] args)
    {
        Date [] dateArr = new Date [5];

        dateArr[0] = new Date (6,7,1991);
        dateArr[1] = new Date (12,12,1991);
        dateArr[2] = new Date (4,1,1991);
        dateArr[3] = new Date (1,11,1990);
        dateArr[4] = new Date (31,3,1991);

        System.out.println(" מערך התאריכים לפני המיון ");
        for (int i = 0; i < dateArr.length ; i++)
        {
            System.out.println (dateArr[i].toString()); ;
        }
        System.out.println();

        Sorter.sort (dateArr);

        System.out.println(" מערך התאריכים אחרי המיון ");
        for (int i = 0; i < dateArr.length ; i++)
        {
            System.out.println (dateArr[i].toString()); ;
        }
    }
}

/*
המיון לפני התאריכים מערך
6/7/1991
12/12/1991
4/1/1991
1/11/1990
31/3/1991

המיון אחרי התאריכים מערך
1/11/1990
4/1/1991
31/3/1991
6/7/1991
12/12/1991
*/
```