

#### תרגיל 4

את הקוד יש להגיש ב- GitHub. את הקישור יש להעתיק ל- moodle.  
תשובות מילוליות יש להגיש בקובץ Word או pdf ולהעלות ל- moodle.

1.

כתבי פונקציה גנרית SortTwo שמקבלת reference לשני משתנים מטיפוס T ומחליפה ביניהם אם הראשון גדול מהשני.

הדרכה :

כדי לממש את הפונקציה צריך להגדיר כיצד משווים בין שני אובייקטים מטיפוס T.  
ניתן לעשות זאת בשתי דרכים :

- א. להוסיף פרמטר לפונקציה שהוא delegate שמבצע את ההשוואה.  
הממשק יהיה דומה לפונקציה Sort שמקבלת פרמטר מסוג **Comparison<T>**
- ב. להגדיר אילוץ על הטיפוס T באופן הבא :

where T : IComparable<T>

ואז בתוך הפונקציה להשתמש בקריאה :

a.CompareTo(b)

2.

בשיעור כתבנו מחלקה `OperationTable` שיוצרת "לוח כפל" (כאשר הפעולה היא לא בהכרח כפל). כתבי מחלקה גנרית, `GenericOperationTable<T>`, שיוצרת לוח של פעולה אבל עכשיו הערכים בלוח הם לא דווקא `int` אלא למשל `double`. (והפעולה גם כן אינה בין שני `int`-ים אלא בין שני `double`-ים)

### שימי לב:

במחלקה `OperationTable` המקורית האופרנדים (המספרים ב"חותרות" של השורות והעמודות) היו מספרים שלמים מטיפוס `int`, והם נוצרו בעזרת לולאה פשוטה.

במחלקה `GenericOperationTable<T>` גנרית, הערכים שביניהם מבצעים את הפעולות אינם בהכרח `int`.

למשל יכולה להיות דרישה ללוח החיבור בין שברים:

	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{4}$
$\frac{1}{2}$			
$\frac{1}{3}$			
$\frac{1}{4}$			

ולכן, יש לשנות את הממשק של המחלקה `OperationTable` כך שחובה להעביר אליה את רשימת הערכים לעמודות ולשורות – באופן הבא:

```
class OperationTable<T>
{
    // the following line defines a _type_ op_func
    public delegate T OpFunc(T x, T y);

    // the following line defines a variable of type op_func
    public OpFunc op;

    public OperationTable(List<T> _row_values, List<T> _col_values,
        OpFunc _op)
    {
        // calculate the table
    }
}
```

כפי שניתן לראות, ה- `ctor` מקבל רשימה של ערכים מטיפוס `T` לשורות ומערך של ערכים מטיפוס `T` לעמודות, וכן את האופרטור שיש להפעיל.

למשל בדוגמא לעיל המערכים row\_values ו col\_values יהיו זהים :

values:

$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{4}$
---------------	---------------	---------------

יצירת המחלקה נעשית באופן הבא :

```
static void Main(string[] args)
{
    List<double> row_values = new List<double>();
    for(int i=; i<=4; i++)
    {
        row_values.Add(1/i);
    }

    List<double> col_values = new List<double>();
    for (int i = 1; i < 5; i++)
    {
        col_values.Add(1/i);
    }

    OperationTable<double> t1 =
        new OperationTable<double>(row_values, col_values, (x,y) => x+y );

    t1.Print();
}
```

שימי לב שהקוד יוצר שני מערכים : row\_values ו col\_values ומעביר אותם ל- ctor של המחלקה. מומלץ לשמור אותם בתוך המחלקה.

במהלך החישוב יש לבצע לולאה בתוך לולאה כאשר הערכים נלקחים מהמערכים row\_values ו col\_values

בהצלחה