מבוא לתכנות

הרצאה 7 – מערכים 1 סמסטר

מערך

מערך הוא *סידרה* של ערכים: •

'a' 'b' 'c' 'd' 'e' 'f' 'g' 'h'

- כל ערך במערך נקרא *איבר* •
- האיברים יכולים להיות מכל סוג. אבל כל האיברים במערך אחד חייבים להיות מאותו הסוג.

יצירת מערך

י כדי להגדיר מערך, אנחנו פשוט מוסיפים [] לסוג המשתנה:

```
int[] grades;
string[] names;
```

- grades ו grades הם שני מערכים ריקים ללא איברים כלל
 - new כדי להגדיר איברים למערך נשתמש במילת המפתח num וב [num] כאשר num הוא מספר האיברים שנרצה במערך

```
grades = new int[3];
names = new string[4];
```

- ערכם 0, grades יש 3 איברים מסוג grades כעת ל grades איברים שערכם "" students
- תכל סוג יוגדר עם ערך ברירת המחדל: int ←0,double, כאשר נדפיס למסך נקבל """), של string ←<u>null</u> ←0.0... bool ←false...

גישה לאיברים

- איברים במערך מסודרים ומאונדקסים כמו מחרוזות
 - 0 האינדקס של האיבר הראשון הינו –
 - n האינדקס של האיבר האחרון במערך בגודל (n-1)
 - ב: כדי לעדכן איבר במקום ספציפי, נשתמש ב:

```
names[1] = "Bob"; //write to the second element
names[0] = "Alice"; //write to the first element
names[3] = "Dora"; //write to the last element
```

: או, שנוכל להגדיר ולעדכן את כל האיברים בבת אחת כך

גישה לאיברים

: כדי לקרוא איבר נשתמש באותו סינטקס

```
names = new string[]{"Alice", "Bob", "Charlie",
"Dora" };
string brown = names[2];
Console.WriteLine(brown); //prints Charlie
```

מעבר על המערך

: כדי לשלוף איברים מהמערך ניעזר באינדקס

```
names = new string[]{"Alice", "Bob", "Charlie",
"Dora" };
for (int i = 0; i < names.Length; i++)
{
    names[i] = "*" + names[i] + "*";
    Console.WriteLine(names[i]);
}</pre>
*Alice*
*Bob*
*Charlie*
*Dora*
```

זוהי תכונה שמחזירה את כמות האיברים במערך Length – הנתון

תרגיל 1

- int Max(int[] values). כתבו את הפונקציה ישל מספרים, הפונקציה תחזיר את המספר הגדול ביותר
 - ראשית, בואו נגדיר את האלגוריתם במילים.

תרגיל 2

- יש לנו 2 רשימות:
- symbols רשימת סמלי Symbols הברות. symbols דוגמה : ["MSFT", "INTC", "CSCO", "GOOG"]
- monthly_gain רשימת רווח/הפסד של החודש שעבר, באחוז, של כל monthly_gain חברה דוגמה : [-0.72, 1.05, 0.76, -1.49]
 - אנו נרצה למצוא את הסמל של החברה עם הרווח החודשי הגבוה
 ביותר

סוגי נתונים

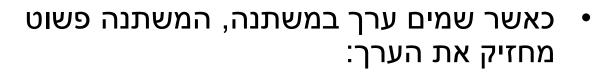
יש שני סוגי של נתונים:

ערך	הפניה
int	string
float\double	array
boolean	
char	

- . סוגי "ערך" מכילים ערך יחיד ופשוט.
- סוגי "הפניה" מחזיקים נתונים מורכבים:
 - מחרוזת: רצף תווים
 - מערך: רצף של ערכים –

סוגי נתונים

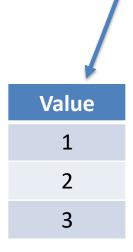
Variable	Value
X	17
numbers	



$$x = 17$$

אבל כאשר שמים הפניה למשתנה, המשתנה מכיל הפניה לאובייקט:

int[] numbers = new int[] {1,2,3};



הפניות מרובות

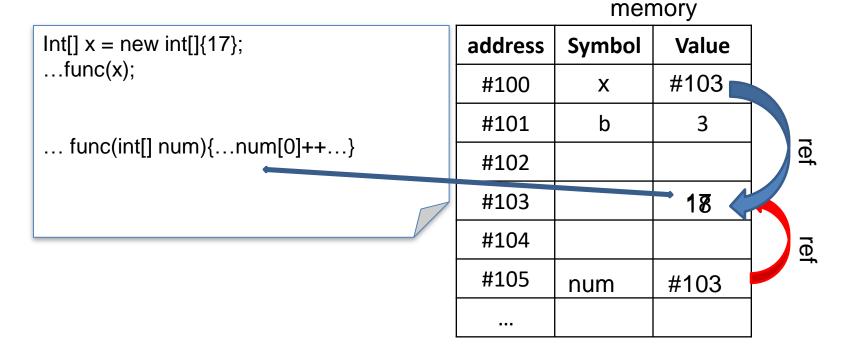
שימו לב לקוד הבא:

```
int[] a = new int[]{1, 2, 3, 4, 5}
int[] b = a
b[2] = 7
```

- ?aב מהם הערכים ב
- − מהם הערכים בd?
- אך הם לא. bi a זה נראה כאילו יזה נראה כאילו
 - !הם <u>הפניות</u> לאותו המערך bi a אמעשה, -
 - a אם נשנה משהו בd, נשנה גם את b.מדוע? בגלל שb וd מפנים לאותו מערך.
 - .a ישנו את bכמו כן גם הפוך, שינויים ב
- <u>הפניות מרובות</u> היא טכניקת תכנות חשובה שנשתמש בה בהמשך הקורס

סוג הפניה

- כשמשתמשים בסוג הפניה (למעט מחרוזת) כארגומנט, אנו
 שולחים את ההפניה ולכן הערכים המקוריים של המערך ישתנו
 מתוך הפונקציה
 - ... מאחורי הקלעים...



תרגיל 3

- int[] FilterEvens(int[] numbers). כתבו את הפונקציה. בהינתן מערך מספרים, הפונקציה מחזירה מערך חדש שמכיל רק את המספרים הזוגיים
 - $\{2,4,6\}$ תחזיר FilterEvens (new int[] $\{1,2,3,4,5,6\}$) דוגמה
 - ראשית, בואו נגדיר את האלגוריתם במילים.

Bubble Sort

- אחד האלגוריתמים המפורסמים (והפשוטים) למיין מערך
 נקרא "מיון בועות". בואו נממש אותו.
 - "repeatedly stepping through the list to be sorted, comparing each pair of adjacent items and swapping them if they are in the wrong order. The pass through the list is repeated until no swaps are needed, which indicates that the list is sorted."

 (wikipedia)

העתקת מערך

- ?מה אם אנחנו רוצים ליצור עותק של המערך
 - אופציה 1: להעתיק איבר איבר •

```
a = new int[] {70,80,90,100,95};
int[] b = new int[a.Length+2];
for (int i = 0; i < a.Length; i++)
{
    b[i] = a[i];
}
a[0] = 100;</pre>
```

- מה הערכים בa?

?ba מה הערכים ב

- !כעת a וd מצביעים על bi a כעת
 - a א ישפיע על b כל שינוי ב
 - b וגם הפוך, כל שינוי בa לא ישפיע על

70

80

90

100

95

העתקת מערך

CopyTo(array, index) נשתמש בפונקציית • אופציה -

```
int[] b2 = new int[a.Length];
a.CopyTo(b2, 0);
                                          הוא מערך היעד Array
   70
                   זהו האינדקס שממנו מתחילים את ההעתקה Index -
   80
   90
   100
                      אופציה 3: נשתמש בפונקציית
Array.Copy(src, dest, length)
   95
int[] b3 = new int[a.Length];
Array.Copy(a,b3, a.Length);
                                            הוא מערך המקור src
           70
           80
                                            הוא מערך היעד dest –
           90
                           זהו מספר האיברים שיש להעתיק Length –
           100
           95
```

העתקת מערך

. אופציה 4: נשתמש בפונקציית - Clone() הפתרון הטוב ביותר +

```
int[] b4 = (int[]) a.Clone();

70
80
90
100
95

(int[]) – נצטרך להמיר את מה שמוחזר ל
```

• מדוע?? נדע זאת בעתיד

מיון מערך

Array.Sort ():שיטה שימושית אחרת למערך היא

```
a = new int[] { 1, 2, 3, 4, 5, 8, 9, 30, 27, 17, 3 };
Array.Sort(a);
```

- השיטה ממיינת את האיברים במערך בהתאם לסוג שלהם ולסדר
 - היא יכולה למיין מחרוזות גם כן:

```
names = new string[] { "Bob", "Dora", "Alice",
    "Charlie" };
Array.Sort(names);
```

ניקיון מערך

- שמה ערכים Clear (array, index, length) פונקציית דיפולטיביים למספר ערכים
 - וחיכן להתחיל Index − −
 - Length how many elements to clear

```
a = new int[] { 1, 2, 3, 4, 5, 8, 9, 30, 27, 17, 3 };
Array.Clear(a,3,3);
```

1,2,3,0,0,0,9,30,27,17,3,

Foreach

foreach(type varName in ArrName)

```
Use varName here as the current value
                          <u>לא ניתן</u> לשנות את תוכן המערך בלולאה זו
                             לא אפשרי לדעת את האינדקס הנוכחי
                                     3. אבל הרבה יותר נוח לכתיבה
int[] a = new int[] { 1, 2, 3, 4, 5, 8, 9, 30, 27, 17, 3};
foreach (int number in a)
    Console.WriteLine(number + 2);
```