

השלמות - לולאת foreach הירארכיה של השפה C#, ToString

C# היא שפה מונחת עצמים מיסודה, ולכן היא בנויה בצורה הירארכית של מחלקות שיורשות ממחלקות אחרות.

מחלקת ה'על' שממנה כל עצם בשפה יורש תכונות ופעולות ניקראת: **Object** - כשדיברנו על הורשה, אמרנו שעל ידי הורשה, המחלקה היורשת (derived class) מקבלת את כל המשתנים (מאפיינים) וכל הפעולות (מתודות) של המחלקה ממנה יורשים. בצורה זו המחלקה היורשת יכולה גם להשתמש בכל הפעולות של המחלקה ממנה ירשה, וגם להגדיר פעולות משל עצמה שאינן נימצאות במחלקת הבסיס (base class)

פעולות וירטואליות (virtual methods)

עוד נושא שעליו דיברנו בעניין ההורשה, היה 'פעולות וירטואליות'. אלה הן פעולות שמוגדרות במחלקת הבסיס, עם מילת המפתח: virtual, וניתן להגדירן מחדש במחלקה היורשת, על ידי ציון במילת המפתח: **override**. למי שלא זוכר, הייתה לנו דוגמה בסיכום שיעור:

http://www.lamed-oti.com/school/tmy/yud_alef/20160121.pdf

המחלקה Object, מכילה מספר פעולות וירטואליות, כגון: ToString(), Equals(), GetType() ועוד מספר כאלה. כעת נדבר על: ToString()

ToString()

פעולה וירטואלית זו, מיועדת על מנת ליצג עצם בצורה של מחרוזת. הרי עצם זה דבר מורכב שיכול להיות בנוי מעשרות מאפיינים, עשרות פעולות, ולכן לעיתים קשה 'לתפוס', או להבין, מהי מהות העצם. הפעולה ToString() מיועדת לפשט עבורנו את ה'מראה' של העצם. אנחנו בוחרים כיצד יראה עצם מסוג מסוים שהגדרנו.

ה'חתימה' של הפעולה ToString במחלקה Object ניראית כך:

```
public virtual string ToString() { }
```

לדוגמה, אם הגדרנו מחלקה שהיא תלמיד (Student), והמאפיינים שלה הם: שם, תאריך לידה, שיכבה, כיתה, מספר טלפון, מערך ציונים, ציון ממוצע ועוד. בנוסף יש בה פעולות שונות, כגון: הוספת איחורים, שינוי ציון ועוד. נניח שהחלטנו שכדי להציג אובייקט (עצם) מסוג תלמיד, נירצה לראות אותו בצורה כזו:

Student name: nnnnnnnn DOB: dd/mm/yyyy Average grade: kk

כדי שזה יתבצע, נגדיר את: ToString() בתוך המחלקה: Student בצורה הבאה:

```
public override string ToString()
{
    return "Student: " + Name + " DOB:" + dob + " Average Grade: " + avg;
}
```

זה בהנחה ששמות המשתנים במחלקה Student הם: name עבור שם התלמיד, dob עבור תאריך לידה, avg עבור ציון ממוצע.

ובכן, הפעולה ToString() בסך הכל מחזירה מחרוזת. השימוש של הפעולה הזו יראה כך (אין תיאור מלא של המחלקה עם הבנאי):

```
Student s1 = new Student("Jenny", "10-12-2000", 0, 88, 90, 75, 80);
Student s2 = new Student("Aaron", "09-10-1999", 0, 70, 92, 78, 67);
Console.WriteLine(s1, "\n", s2);
```

Result:

```
Student: Jenny DOB:10-12-2000 Average Grade: 83.25
Student: Aaron DOB:09-10-1999 Average Grade: 76.75
```

אם לא מגדירים את הפעולה ToString() במחלקה מסוימת ומנסים להשתמש בה, כלומר, או להדפיס את העצם (נניח s1) או להדפיס: s1.ToString() נקבל את שם המחלקה (בד"כ שם ה-namespace נקודה, שם המחלקה) - אפשר לנסות זאת.

foreach

זוהי דרך לעבור על סידרה של עצמים אחד אחד לדוגמה:

```
string[] pets = { "dog", "cat", "bird" };
```

```
// ... Loop with the foreach keyword.
```

```
foreach (string value in pets)
{
    Console.WriteLine(value);
}
```

result:

dog

cat

bird

המילה value היא מילה שאנחנו 'המצאנו' ואינה מוגדרת בשום מקום אחר, כלומר צורת הלולאה זה מחוץ לסוגריים לכתוב: foreach (כמילה אחת), ובסוגריים את סוג העצם, משתנה שמוגדר כאן ואחר כך מילת המפתח in, ואחרי זה, שם העצם שמכיל סידרת ערכים.

עוד דוגמה לשימוש:

```
List<int> list = new List<int>();
list.Add(1);
list.Add(2);
list.Add(3);
// Loop over list elements using foreach-loop.
foreach (int element in list)
{
    Console.WriteLine(element);
}
```

כאן יודפס:

1
2
3

הפעם קראנו למשתנה שמקבל את ערכי הרשימה: element

```
foreach (char c in "example")  
{  
    Console.WriteLine(c);  
}
```

הפעם קראנו למשתנה: c ומה שיודפס זה:

e
x
a
m
p
l
e