

המשך תכנות מונחה עצמים

היום דיברנו על איך בעצם יוצרים עצמים (או אובייקטים), עבור תכנות מונחה עצמים. הזכרנו שוב שאחת הסיבות להתפתחות צורת התכנות הזו הייתה לדמות בתכנות את העולם המציאותי, כך שיהיה קל יותר ופחות מסובך לכתוב תוכניות בעלות משמעות, כלומר תוכניות שיעזרו לנו לפתור בעיות מחיי היומיום.

כאשר דיברנו על מחלקה (class) אמרנו שלמעשה זה כמו מן אב-טיפוס, או סוג של 'דבר', כמו למשל הגדרה של: 'תלמיד', או הגדרה של 'כתה'. אמרנו גם שמחלקה מכילה תכונות או מאפיינים (קראנו להם משתנים תחילה) וגם פעולות. למשל מחלקה של תלמיד יכולה להכיל בתור מאפיינים: שם תלמיד, תאריך לידה, מספר איחורים, מספר חיסורים וכד'. הפעולות שניתן להגדיר עבור תלמיד יכולות להיות דברים כגון: לאחר, להבריז משיעור, לקחת בחינה ועוד.

בינתיים לא ראינו כיצד להגדיר את המחלקה, אבל אמרנו שאם המחלקה כבר מוגדרת, אפשר להשתמש בה כדי ליצור אובייקטים. במילים אחרות, המחלקה היא כמו מן תבנית של הגדרות שבעזרתה נוכל ליצור אובייקטים שונים, ולאחר יצירת האובייקטים, נוכל להשתמש בהם בצורות שונות בתוכנית.

למשל:

```
class Student; //We assume that a class called Student has been defined
```

```
public static void Main(){
    Student s1= new Student("Barak Obama", "10/12/1961", 0);
    Student s2= new Student("Hillary Clinton", "05/06/1946",2);
    s2.Late();
```

הסבר: במחלקה: Student ישנם 3 שדות (מאפיינים) שהם: שם, תאריך לידה ומספר איחורים. שני הראשונים הם מסוג מחרוזת (string) והאחרון מסוג שלם (int) במחלקה גם מוגדרת פעולה שניקראת: Late (מה שהיא תעשה זה תוסיף 1 למספר האיחורים של התלמיד.)

יצרנו כאן שני אובייקטים, s1 ו- s2 (כאשר סיפקנו שמות, תאריכי לידה ומספר התחלתי של איחורים עבור כל אחד מהסטודנטים). לאחר מכן הפעלנו את פעולת האיחור על האובייקט s2. כתוצאה מכך, אם כעת נדפיס את תוכן השדות של האובייקטים, עבור: Hillary Clinton יודפסו 3 איחורים ועבור: Barak Obama יודפסו 0 איחורים. זה מכיוון שהפעלנו את פעולת ה- Late רק על האובייקט s2 (ששייך ל- Hillary Clinton).

השתמשנו במילת מפתח חדשה שניקראת: new על מנת ליצור אובייקט חדש. הערכים שבתוך הסוגריים הם הערכים ההתחלתיים של האובייקט וניקראים בעגת המתכנתים: פרמטרים.

באופן כללי, כאשר מתכננים את התוכנית, נצטרך להחליט מה הן המחלקות שאותן נגדיר, ונגדיר אותן כאוסף של תכונות ופעולות שדרושות לנו כדי לבצע את מטרת התוכנית. האמת שבתוכניות פשוטות קשה לראות את היתרון בתכנות מונחה עצמים, אבל נקווה להגיע לתוכניות מורכבות יותר, שם נוכל להיווכח ביתרונות של תכנות מסוג זה.

01.12.2014

עוד דבר קטן שעשינו היה שיפור של הדפסת לוח הכפל. הדגמנו זאת בעזרתו האדיבה של יואב בן אזר. בהדפסה של המכפלה (IJ) כתבנו את הדבר הבא:

```
Console.Write("{0}", IJ.ToString().PadLeft(4));
```

הפעלנו כאן 2 פעולות על המכפלה. הפעולה: ToString() הופכת את המספר למחרוזת, כלומר אם למשל המספר היה: 50, זה יהפוך ל: "50" (למרות שאת הגרשים לא ניראה בהדפסה). והפעולה השנייה הייתה ל "דפן" את המחרוזת עם רווחים משמאל כאשר 4 תווים מוקצים להדפסתה ולכן כל מספר יודפס ב-4 תווים כאשר הוא מוצמד לימין ומשמאלו ישנם מספיק רווחים כדי לראות בבירור את כל המספרים.

זו למעשה הייתה הדגמה של מעט מהכוח של השפה בהדפסות עם פורמט כרצוננו.

שיעורי בית: לקרוא בספר עמודים 12 עד 18.