

חזרה והרחבה של תכנות מונחה עצמים

היום ניסינו לחשוב על עצמים שונים ולרשום מספר תכונות שלהם וכן מספר פעולות שהם יהיו מסוגלים לבצע. כדי ליישם את העצמים שעליהם חשבנו בתוכנית מחשב, עברנו על השלבים הבאים:

1. הגדרה (עם או בלי טבלה) של התכונות הרצויות (למשל אם העצם היה כלב, היה לו סוג, משקל וכד'), וכן את הפעולות שידע לעשות (לרוץ, לנבוח וכו')
2. נתנו שמות וסוג לכל תכונה (הגדרנו כמשתנה), למשל משקל הכלב יהיה מספר שלם (int) סוג הכלב תהיה מחרוזת, וגם שמות מתאימים לפעולות (מתודות), כמו למשל פעולת נביחה תיקרא: bark. אמרנו גם שאת התכונות נגדיר כמשתנים מסוג: private (פרטיים) ואת הפעולות נגדיר כ- public (ציבורי) - אז התכונות יהיו משהו כמו: private int dog_weight וכן: private string dog_type
3. נתעכב לרגע על הגדרת הפעולה (method) שהיא מעט מורכבת. בהגדרת ה- method ישנם האלמנטים הבאים: public (סוג הגישה ציבורי כך שניתן להפעיל את פעולת ה- bark מכל מקום בתוכנית. לאחר מכן בא סוג הערך שתחזיר הפעולה, ומילת המפתח: void מציינת שפעולת ה- bark אינה מחזירה כל ערך. לאחר מכן סוגריים ריקים (אבל חייבים להיות שם) - ריקים כיוון שאיננו מספקים שום פרמטר לפעולת הנביחה. לאחר מכן פתיחת הבלוק של הפעולה {}, פעולת הדפסה, וסגירת הבלוק } שמסמל את סופה של הפעולה. הנה זה לפניכם:

```
public void bark()
{
    Console.WriteLine("Woof woof, woof woof");
}
```

עוד דבר חשוב שהזכרנו בקשר לעצמים, זה שדרושה לנו "פעולה בונה". זו פעולה מיוחדת שניקראת: constructor והיא למעשה יוצרת את העצם הראשוני, כאשר נשתמש במילת המפתח new כפי שניראה בהמשך התוכנית.

הפעולה הבונה שמה תמיד יהיה כשם המחלקה (כך המחשב יזהה אותה) ולא יהיה לה שום ערך מוחזר, אפילו לא void. יתכן שיהיו לה פרמטרים. אז אם נכתוב את הפעולה הבונה עבור מחלקת הכלב שלנו, היא תראה בערך כך:

```
public dog(int initial_weight, string initial_type)
{
    dog_weight = iniital_weight;
    dog_type = iniital_type;
}
```

שימו לב שכל מה שעושה כאן הפעולה הבונה זה לשים את ערכי הפרמטרים שלה (initial_weight (initial_type לתוך המשתנים שמייצגים את התכונות של הכלב במחלקה (dog_weight, dog_type)

4. יישום בתוכנית. כללית תוכנית בשפת C# היא אוסף של מחלקות (classes) ועד היום ראינו רק מחלקה אחת ובתוכה רק מתודה אחת שניקראת: Main. ולכן כדי לעשות סדר בבלאגן כעת נראה מבנה של תוכנית שבה מוגדרת מחלקה (class) נוספת שניקראת: dog (עבור הכלב שלנו)

```
using System;
using System.Collections.Generic;
namespace practice_makes_perfect
{
    class dog
    {
        private int dog_weight;
        private string dog_type;
        //constructor
        public dog(int inital_weight, string iniital_type)
        {
            dog_weight = iniital_weight;
            dog_type = iniital_type;
        }

        public void bark()
        {
            Console.WriteLine("Woooooooooooooooooooooffffff");
        }
    } //end of class dog.

    class MainClass
    {
        public static void Main(string[] args)
        {
            dog d1= new dog(20, "puddle"); //instantiating a dog object
            dog d2= new dog(3, "Chiuwhawha); //instantiating another dog object
            d1.bark();
        }
    }
}
```

שימו לב שבתוך המתודה Main עדיין אנחנו מתחילים את הלוגיקה של התוכנית, ולכן שם גם אנחנו יוצרים את העצמים מהמחלקה (זיכרו מחלקה מוגדרת פעם אחת, אבל אפשר להשתמש בה כדי ליצור עצמים שונים (בדוגמה הנ"ל יוצרים 2 עצמים, ועל הראשון מפעילים את פעולת ה-bark, על ידי קישור העצם (d1) עם הפעולה בעזרת נקודה. d1.bark;

שימו לב שעל ידי פקודה כמו: dog d1= new dog(20, "puddle"); מה שיקרה זה פעולת הבונה של מחלקה dog תיכנס לפעולה ותיקח את הפרמטרים שנתנו (20 ו-"puddle") ותשים אותם במשתנים של התכנות על פי הסדר (לעיתים ניקראים גם member variables). כלומר במשתנה dog_weight של d1 יהיה 20, ובמשתנה dog_type - יהיה הערך "puddle"

בואו ונוסיף את סוג הכלב בנביחה שנידע איזה כלב נבח, בצורה הבאה:

```
Console.WriteLine("Woooooooooooooooooooooffffff says: {0} ", dog_type);
```

הערה: לעיתים לתכנות קוראים בשמות כמו: class variables או member variables (ע"מ להבדילם ממשתנים אחרים שאפשר להגדיר בתוך המחלקה - נראה זאת בהמשך).

כעת תוכלו לשאול: מדוע לטרוח כל כך הרבה כדי לעשות דבר כה שטותי כמו להדפיס: woof כנביחת כלב (מחלקה חדשה, עם תכונות, פעולות, פעולה בונה וכד').

התשובה היא שכדי להבין את העיקרון לקחנו דוגמה (או דוגמאות) פשוטה, אבל בהמשך נראה מדוע זה יכול להיות רעיון טוב לתכנת בעזרת סוג תכנות זה שהוא "מונחה עצמים".