

מבוא לתכנות II

הרצאה 4 – תכונות, סטטי, קבוע ו- Enum

סמסטר 2

תכונות (Properties)

- תכונות הן דרך שימושית לכפות כימוס (encapsulation)
 - מסתיר את שדות המחלקה
 - מתן גישה לשדות דרך סביבה מבוקרת = פונקציה.
 - נוכל לאמת את הנתונים הנכנסים.
 - נוכל להחזיר ערך מחושב ולא רק את המקורי.
 - מחליף את השימוש בפונקציות `GetField()` ו `SetFiled()`
- מגדירים את השדה כפרטי (private) ואת התכונה כציבורית (public)
- קונבנציות הכתיבה עבור האות הראשונה:
שם השדה יתחיל באות קטנה ושם התכונה יתחיל באות גדולה

```
class Student
{
    private string name;
    public string Name
    {
        get { return "*** " + name + " ***"; }
        set
        {
            if (value.Length > 2)
                name = value;
            else
                Console.WriteLine("name is too short!");
        }
    }
}

...
Student s1 = new Student();
//s1.name = "avi"; //Error - private field!
s1.Name = "ga";
Console.WriteLine(s1.Name);
s1.Name = "gad";
3 Console.WriteLine(s1.Name);
```

name is too short!
*** ***
*** gad ***

וריאציות

```
class Student
{
public int Grade { get; set; }
public int Grade1 { get { return Grade; } set { Grade = value; } }
//the same as Grade

//public int Grade2 { get { return Grade * 2; } set; } ERROR- if only
//one is implemented then don't write the other
public int Grade2 { get { return Grade * 2; } } //read only!!
public int Grade3 { set { Grade = value + 5; } } //write only!!!
...
class Program
{
s1.Grade = 77;
//s1.Grade2 = 88; //Error - read only!
s1.Grade3 = 80;
//Console.WriteLine(s1.Grade3); //Error - write only
Console.WriteLine(s1.Grade); //85
```

פרטי / פומבי (Private\ Public)

```
class Student
{
...
public int Grade4 {private get { return Grade; } set { Grade = value; } }
public int Grade5 {get { return Grade; } private set { Grade = value; } }
...
class Program
{
...
s1.Grade4 = 44;
//Console.WriteLine(s1.Grade4); //Error - private
//s1.Grade5 = 55; //Error - private
Console.WriteLine(s1.Grade5);
```

סטטיק (Static)

- אלמנטים סטטיים (שדות או פונקציות) מוגדרים עבור כלל המחלקה, ולא עבור אובייקט ספציפי (מופע) של המחלקה. רק אחד נוצר עבור כל האובייקטים.
- ההגדרה הדיפולטיבית היא לא סטטית אלה מופע (למשל משתנה מופע), יש להוסיף לפני שם השדה את מילת המפתח *static*

```
public static int counter;  
public static void ClearCounter() {...}
```

- אנו קוראים לאלמנטים הסטטיים דרך שם המחלקה ולא דרך שם האובייקט שלה.

```
Car c1 = new Car();  
//c1.counter = 7; //ERROR - static can not be accessed through  
the object!  
Car.ClearCounter();
```

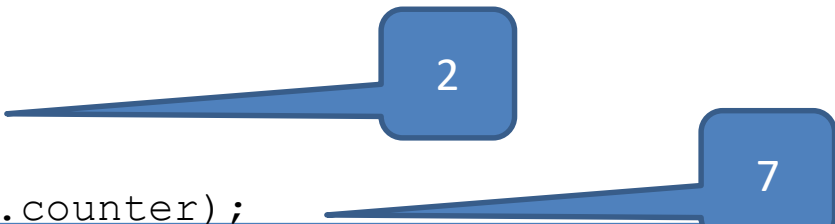
• דוגמה לשימוש באלמנט סטטי

```
class Car
{
    public int Number { get; set; }
    public string Brand { get; set; }
    public static int counter;

    public Car(){counter++;}

    static public void ClearCounter(){counter = 0;}

    public void ShowCounter(){Console.WriteLine(counter);}
}
...
Car c1 = new Car();
Car c2 = new Car();
c2.ShowCounter();
Car.counter = 7;
Console.WriteLine(Car.counter);
```



אלמנטים סטטיים

- אלמנטים סטטיים נגישים בתוך המחלקה עצמה מכל פונקציה, סטטית או מופע.
- אלמנטים סטטיים נגישים מחוץ למחלקה רק דרך שם המחלקה ולא דרך אובייקט שלה.

פונקציה סטטית

- פונקציה סטטית יכולה לגשת רק לשדות ופונקציות סטטיות, ולא לשדות ופונקציות מופע.
- פונקציות מופע יכולות לגשת לשדות ופונקציות מופע וסטטיים.

סטטיק

- השתמשו בסטטיק כאשר המשתנה שייך לכל אובייקטי המחלקה (ולא שייך למופע אחד)
- השתמשו בפונקציות סטטיות כאשר הן עושות שימוש רק בשדות סטטיים או כאשר החישוב לא תלוי אובייקט ספציפי
- בדרך כלל פונקציות סטטיות יקבלו את המידע הנחוץ להן לתפקוד באמצעות פרמטרים, כמו מחלקת Math והפונקציות שלה, שבמקרה שלה אין צורך באובייקט של Math אלא רק בחישובים שהיא מספקת. הרי שני מופעים של המחלקה Math היו נותנים את אותם ערכים תמיד.

קבוע (Const)

- משתנה קבוע (*Const*) הוא משתנה סטטי אשר מאותחל בעת יצירתו ולא יכול להשתנות לאחר מכן, כמו הערך π

```
class Car
{...
    //public const string trademark; //ERROR - must be assigned!
    public const string trademark = "Ruppin";
...
    //public void changeConst() { trademark = "aaa"; } //can not be
                                                //changed!

    public void ShowTradeMark(){ Console.WriteLine(trademark); }
}
...
Car c1 = new Car();
Car c2 = new Car();
//c2.trademark="nir" ; //ERROR - const is static variable by default!
Console.WriteLine(Car.trademark);
c2.ShowTradeMark();
```

Ruppin
Ruppin

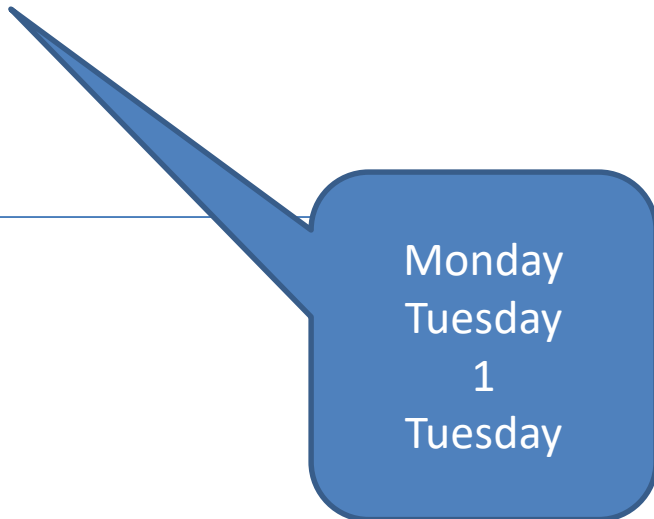
Enum

- יצירת רשימה של קבועים שמהווה סוג חדש בשפה, כאשר כל קבוע בה מיוצג על ידי מספר
- המטרה היא ליצור קוד קריא ובהיר שחוסך שימוש במספרים וקודים אשר מייצגים סיטואציה בעלת משמעות.

```
namespace lecture_11
{
    enum DayOfWeek
    {
        Sunday = 0,
        Monday = 1,
        Tuesday = 2,
        Wednesday = 3,
        Thursday = 4,
        Friday = 5,
        Saturday = 6,
    }
}
```

Enum

```
DayOfWeek day = DayOfWeek.Monday;  
Console.WriteLine(day);  
DayOfWeek day2 = (DayOfWeek)2;  
Console.WriteLine(day2);  
int num = (int)day;  
Console.WriteLine(num);  
string str = day2.ToString();  
Console.WriteLine(str);
```



Monday
Tuesday
1
Tuesday

Enum

- ברירת המחדל של enum היא int אבל ניתן להשתמש בכל סוג של מספר שלם: int, byte...
- הרשימה מתחילה ב-0 וגדלה באחד בכל צעד (המספרים בדוגמה הקודמת מיותרים)
- נוכל לקבוע ידנית את הערך הקבוע והרשימה תמשיך לספור ממנו הלאה (+1)
- ניתן לקבוע שפריט ברשימה יהיה שווה לפריט אחר ברשימה, באמצעות שימוש באופרטור " = "
- ניתן להשתמש על המשתנים הללו באמצעות פונקציות אריתמטיות
- כאשר מדפיסים, הקבוע הוא זה שיודפס, חוץ מהמקרים שבהם לא הוגדרו קבועים כלל

Enum

```
enum YesNo : byte{Yes, No}  
enum Grade {VeryGood, Good, OK=100, Fail, Wors=Fail }
```

```
YesNo answer = YesNo.Yes;  
Console.WriteLine(answer);  
Grade myGrade = Grade.Good;  
Console.WriteLine((int)myGrade);  
myGrade = Grade.Wors;  
Console.WriteLine((int)myGrade);  
Console.WriteLine(myGrade);  
myGrade = Grade.Wors+2;  
Console.WriteLine((int)myGrade);  
Console.WriteLine(myGrade);
```



Yes
1
101
Fail
103
103

פונקציות עזר של Enum

```
//get all the constant names
string[] arrstr= Enum.GetNames(typeof(Grade));
//get all the constant numbers\values
int[] arrint = (int[])Enum.GetValues(typeof(Grade));
//casting from string to enum. Bool=true, means ignore case
Grade g = (Grade)Enum.Parse(typeof(Grade), "Good", true);
//checking for a string if it is defined as enum
Console.WriteLine(Enum.IsDefined(typeof(Grade), "BAD"));
Console.WriteLine(Enum.IsDefined(typeof(Grade), "Good"));
```

False
True

- צרו enum שמייצג רשימת צבעים
- הדפיסו את כל שמות הצבעים וערכיהם המספריים
- קלטו מהמשתמש שם של צבע והדפיסו את השם של הצבע הבא בתור ברשימה
- אם המשתמש הזין שם לא חוקי אז הדפיסו הודעת שגיאה