



جامعة حماه  
المعهد التقني للحاسوب  
السنة الأولى

محاضرة ٦

برمجة ٢

عملي

قسم البرمجيات

إعداد:

م. أريج فياض

م. رفا البنات

### • الخصائص (properties):

الحقول التي تكون بداخل ال class تكون محمية ومخفية على المستخدم ولا يستطيع الوصول إليها ولكن ماذا لو كنا نريد تمرير قيم لهذه الحقول من خارج ال class بدون أن نتعامل مع الحقول بشكل مباشر أو بدون دوال ؟

في هذه الحالة نستخدم get و set وهما كلمتان محجوزتان في اللغة حيث تمكنك set من وضع قيم لهذه الحقول أي الكتابة فقط، و get تمكنك من قراءة البيانات التي تكون بداخل أي حقل أي تكون للقراءة فقط. الموصلات تعامل معاملة الدوال ، فنحن نتحكم بمن يستطيع تمرير قيم للحقول ومن لا يستطيع ، من يمكنه رؤيتها ومن لا يمكنه.

لا يشترط كتابة set و get معاً في نفس الخاصية، بل يمكننا جعل هذا الحقل للقراءة فقط وعندها نستخدم get فقط، أو للكتابة فقط فنستخدم set أو نستخدم كليهما.

نمرر القيمة للحقل من خلال كلمة value المحجوزة في اللغة وهي تأخذ القيمة التي تأتي من المستخدم وتمررها للحقل المشار إليه ويجب أن تكون القيمة من نفس نوع الحقل أو المتغير.

وفي get نسترجع القيمة التي بداخل الحقل من خلال الكلمة المحجوزة return .

مثال ٥:

ليكن لدينا موظف له اسم وراتب بحيث يكون الراتب لا يقل عن ٣٠٠٠٠ يطلب تمثيل هذه المعلومات باستخدام الصف وباستعمال الخصائص get&set

```
class Employee
{
    private string EmployeeName;
    private int EmployeeSalary;

    public string name
    {
        set
        { EmployeeName = value; }

        get
        { return EmployeeName; }
    } //end of name

    public int salary
    {
        set
        {
            if (value < 30000)
                Console.WriteLine("invalid data");
            else
                EmployeeSalary = value;
        }
    }
}
```

```

        get
        { return EmployeeSalary; }

    } //end of salary
}

```

في خاصية ال salary أنشأنا شرطاً لكي نتأكد من القيمة التي تأتي من المستخدم هل هي في المدى المطلوب وفي حالة كانت في المدى المطلوب تُخزن بداخل الحقل اما اذا كانت خارج المدى فلا تُخزن.

```

class Program
{
    static void Main(string[] args)
    {
        Employee e1 = new Employee();
        e1.name = "karem";
        e1.salary = 90000;
        Console.WriteLine("employee name:" + e1.name);
        Console.WriteLine("employee salary:" + e1.salary);
        Console.ReadKey();
    }
}

```

مثال:

أنشئ صف يحوي خصائص سيارة (الموديل، اللون، الرقم) واعطائها قيم باستخدام `get&set` وبدون استخدام البواني.

```

class Car
{
    private string Car_Model;
    private string Car_Color;
    private int Car_id;

    public string model_property
    {
        get { return Car_Model; }
        set { Car_Model = value; }
    }
    public string color_property
    {
        get { return Car_Color; }
        set { Car_Color = value; }
    }
    public int id_property
    {
        get { return Car_id; }
        set { Car_id = value; }
    }
}

class Program
{
    static void Main(string[] args)

```

```

{
    Car myCar1 = new Car();
    myCar1.model_property = "kia";
    myCar1.color_property = "white";
    myCar1.id_property = 87569;
    Console.WriteLine("car model: "+myCar1.model_property);
    Console.WriteLine("car color: "+myCar1.color_property);
    Console.WriteLine("car id: "+myCar1.id_property);
    Console.WriteLine("-----");
    Car myCar2 = new Car();
    myCar2.model_property = "mercedes";
    myCar2.color_property = "black";
    myCar2.id_property = 4235;
    Console.WriteLine("car model: " + myCar2.model_property);
    Console.WriteLine("car color: " + myCar2.color_property);
    Console.WriteLine("car id: " + myCar2.id_property);
    Console.WriteLine("-----");
    Console.ReadKey();
}

```

```

car model: kia
car color: white
car id: 87569
-----
car model: mercedes
car color: black
car id: 4235
-----

```

مثال:

اكتب برنامج لحساب مساحة ومحيط مستطيل باستخدام get&set

```

class rectangle {
    private int length;
    private int width;
    public int rect_length
    {
        get{return length;}
        set{length=value;}
    }
    public int rect_width
    {
        get{return width;}
        set{width=value;}
    }
    public int area()
    {
        int a = length * width;
        return a;
    }
    public int perimeter()
    {
        int p = (length + width) * 2;
        return p;
    }
}
class Program
{

```

```

length=4
width=5
area=20
perimeter=18

```

```
static void Main(string[] args)
{
    rectangle R1 = new rectangle();
    R1.rect_length = 4;
    R1.rect_width = 5;
    Console.WriteLine("length=" + R1.rect_length);
    Console.WriteLine("width=" + R1.rect_width);
    Console.WriteLine("area="+R1.area());
    Console.WriteLine("perimeter="+R1.perimeter());
    Console.ReadKey();
}
```

### • الكلمة المحجوزة this :

تستخدم بداخل ال class وتستطيع رؤية كل ما بداخل ال class تعمل وكأنها كائن من هذا ال class فنستخدمها بداخل الدوال التي بداخل ال class لكن غير مسموح استخدامها بداخل دوال من نوع static ولا تستطيع رؤية المتغيرات أو الحقول التي تكون من نوع static .

بإعادة كتابة المثال السابق:

```
class rectangle
{
    public int length;
    public int width;

    public rectangle(int l, int w)
    {
        this.length = l;
        this.width = w;
    }

    public int area()
    {
        int a = length * width;
        return a;
    }
    public int perimeter()
    {
        int p = (length + width) * 2;
        return p;
    }
}
class Program
{
    static void Main(string[] args)
    {
        rectangle R1 = new rectangle(4,5);
        Console.WriteLine("length=" + R1.length);
        Console.WriteLine("width=" + R1.width);
        Console.WriteLine("area=" + R1.area());
        Console.WriteLine("perimeter=" + R1.perimeter());
        Console.ReadKey();
    }
}
```

## • الكلمة المحجوزة Static:

المتغيرات والدوال التي بداخل ال class ويكون نوعها static تكون ثابتة أي انه لا يمكن التعامل معها إلا من خلال اسم ال class نفسه.

نستخدم static عندما نريد التعامل مع الدالة أو الحقل خارج ال class بدون الحاجة إلى أخذ نسخة منه object فمثلاً الدالة Writeline () هي دالة static داخل ال console وال console هو ال class الخاص بدوال الادخال والاخراج . فعند استدعاء هذه الدوال نستخدم اسم ال class مباشرة بدون ان نأخذ منه نسخة.

والدالة الرئيسية يجب أن تكون من نوع static وأي دالة نريد استدعاؤها مباشرة في دالة من النوع static يجب أن تكون من النوع static ، لكن يمكن التعامل مع المتغيرات من نوع static من خلال الدوال، حتى لو لم تكن الدالة static لكن العكس خطأ ، أي أننا لا نستطيع استخدام متغير ليس static مع دالة نوعها static .

```
class Program
```

```
{
    static int y;
    static void name()
    { y = 4; }
```

في هذه الحالة يجوز اسناد قيمة للمتغير والتعامل معه بشكل طبيعي لان نوعه static والدالة نوعها static .

مثال:

اذا رغبتنا بإدخال رقم تسلسلي للمستطيلات التي يتم ادخالها يمكننا استخدام متحول ستاتيكي يعمل كعداد كالتالي:

```
public int length;
public int width;
public static int number = 0; //متحول ستاتيكي
public rectangle()
{
    length = 4;
    width = 5;
    number++;
}
public int area()
{
    int a = length * width;
    return a;
}
public int perimeter()
{
    int p = (length + width) * 2;
    return p;
}
}
class Program
{
    static void Main(string[] args)
    {
```

```

    rectangle R1 = new rectangle();
    Console.WriteLine("ractangle number: " + rectangle.number); //static number1
    R1.length = 6;
    R1.width = 2;
    Console.WriteLine("length=" + R1.length);
    Console.WriteLine("width=" + R1.width);
    Console.WriteLine("area=" + R1.area());
    Console.WriteLine("perimeter=" + R1.perimeter());

    rectangle R2 = new rectangle();
    Console.WriteLine("ractangle number: " + rectangle.number); //static number2
    R2.length = 8;
    R2.width = 5;
    Console.WriteLine("length=" + R2.length);
    Console.WriteLine("width=" + R2.width);
    Console.WriteLine("area=" + R2.area());
    Console.WriteLine("perimeter=" + R2.perimeter());

```

```

ractangle number: 1
length=6
width=2
area=12
perimeter=16
ractangle number: 2
length=8
width=5
area=40
perimeter=26

```

مثال:

ليكن لدينا موظف له المعلومات التالية: (اسم الموظف من نوع private ورقم تسلسلي زوجي يبدأ من 2 وله عنوان وعدد ساعات عمل واجرة الساعة جميعهم من نوع public).

المطلوب: باني بدون وسطاء ، وتابع لحساب الراتب باليوم الواحد ، وتابع لطباعة اسم الشخص وراتبه.

```

class Employee
{
    private string name; //الاسم
    public static int number; //الرقم التسلسلي الزوجي
    public int number_hour; //عدد ساعات العمل
    public string address; //العنوان
    public int price; //أجرة الساعة الواحدة
    public Employee() //باني بدون وسطاء
    {
        name = "kinda";
        address = "hama";
        number += 2;
        price = 1000;
    }
    public int salary_day() //تابع حساب أجرة اليوم الواحد
    { return number_hour * price ; }
}

```

```

public void print()//تابع طباعة الاسم واجرة اليوم
{ Console.WriteLine("employee name:" + name +
    " & salary in one day:" + salary_day()); }
// private استخدام توابع get&set لادخال الاسم لأنه من النوع
public string getname()
{ return name; }

public void setname(string name)
{ this.name = name; }
}
class Program
{
    static void Main(string[] args)
    {
        Employee e1 = new Employee();
        Console.WriteLine("employee id:"+Employee.number);
        e1.setname("baraa");
        e1.number_hour = 5;
        e1.price = 2000;
        e1.print();

        Employee e2 = new Employee();
        Console.WriteLine("employee id:"+Employee.number);
        e2.setname("samer");
        e2.number_hour = 8;
        e2.price = 1500;
        e2.print();
    }
}

```

```

employee id:2
employee name:baraa & salary in one day:10000
employee id:4
employee name:samer & salary in one day:12000

```