

مراجع لكافة طلاب الجامعات السورية

محاضرات - سلايدات - اختبارات

مناهج عربية



arabeducationsite.com

البرمجة غرضية التوجه Object Oriented Programming (OOP)

البنية الأساسية لأي برنامج غرضي التوجه هو الصف **Class**

تهدف البرمجة غرضية التوجه oop الى تضمين المعطيات والتوابع ضمن مكونات تدعى الصفوف class نسمي أي بناء من صف بالغرض object أو كائن.

يتألف الصف من حقول **Fields** وخصائص **Properties** ومناهج **Methods**.

*الحقول: هي المعطيات حيث نخزن ضمنها المعلومات

*الخصائص: هي توابع تمكننا من التعامل مباشرة مع الحقول

*المناهج: هي توابع تقوم بإجراء عمليات على المعطيات (وظائف الكائن)

التركيب النحوي للـ **Class**
الجزء الأول:

DataMember جزء البيانات الأعضاء البيانية / البطاقة الشخصية للكائن الذي يحمل مواصفات الصف.
الجزء الثاني:

هو جزء خاص بالدوال (وظائف الكائن) **Functions**

```
Class      (اسم اختياري) اسم الصف
}
DataMember
اسم العضو (اختياري) نمط المعطيات محدد الوصول
Functions
(وسطاء) اسم الدالة نمط الاعداد محدد وصول ثانوي محدد وصول أساسي
}
جسم الدالة التي ستقوم بالدالة بتنفيذه
} }
```

```
namespace ConsoleApplication1
```

```
{
    class time
    {
        int second, minute, hour;
        public int Second
        {
            set { second = ((value >= 0 && value < 60) ? value : 0); }
            get { return second; }
        }
        public int Minute
        {
            set { minute = ((value >= 0 && value < 60) ? value : 0); }
            get { return minute; }
        }
    }
}
```

مثال

```

}
public int Hour
{
    set { hour = ((value >= 0 && value < 24) ? value : 0); }
    get { return hour; }
}

```

Program Class

```

{
Static void Main (string [] args)
{

```

في الدالة Main: - إنشاء كائنات

```

Time t = new Time ();
Time t1 = new Time ();

```

- استدعاء الدوال

```

t1.Hours = 9;
t1.Minutes = 35;
Console.WriteLine (t1.Hours + ":" + t1.Minutes);
}
}

```



الخصائص Properties

لدينا الـ **Class Circle** يمثل دائرة، ويحتوي الحقول التالية:
x.v من نوع **Int** تمثل مركز الدائرة، **r** من نوع **double** تمثل نصف القطر

والمطلوب:

- 1- كتابة الخصائص **Properties**
- 2- الدالة **Area** يعيد مساحة الدائرة.
- 3- الدالة **Around** يعيد محيط الدائرة.
- 4- الدالة **Draw** يقوم بطباعة معلومات الدائرة على الشاشة .

```
namespace
{
    class circle
    {
        int x, y;
        double r;
        public int X
        {
            set { x = value; }
            get { return x; }
        }
        public int Y
        {
```

الخاصية هي عضو دالي :
 1. اسمها نفس اسم العضو البياني
 2. ويبدأ اسم الخاصية بحرف كبير
 3. مكونة من جزأين :
 الدالة **set** : لاسناد قيمة للعضو البياني
 الدالة **get** : لاعادة قيمة للعضو

```

    set { y = value; }
    get { return y; }
}
public double R
{
    set { r = Math.Abs(value); }
    get { return r; }
}
public double Area()
{
    return 3.14 * r * r;
}
public double Around()
{
    return 2 * 3.14 * r;
}
public void Draw()
{
    Console.WriteLine("{0},{1}{2}", x, y, r);
}
}
class Program
{
    static void Main(string[] args)
    {
        circle c = new circle();
        c.R = 7.5;
        c.X = 2;
        c.Y = 5;
        Console.WriteLine(c.Area());
        Console.WriteLine(c.Around());
        c.Draw();
    }
}
}

```

قيمة ثابتة او Value
يتم قراءتها من لوحة
المفاتيح بالتحويل

ملاحظة: الفضاء namespace يحوي صف واحد على الأقل وهو الصف Program (صف الدالة main)

وظيفة :

اكتب برنامج بلغة # c لتمثل الصف Employee
يملك الدوال التالية:

1. دالة input لإسناد قيم البيانات
2. الخصائص
3. دالة print لطباعة قيم البيانات

اكتب الدالة main وأنشئ كائنات Employee واستدعي الدوال لاختبار البرنامج .