

الصفوف المجردة

(Abstract classes)

مقرر برمجة متقدمة - ١ (عملي)

المحاضرة الثالثة

أ.كنانة الحلواني

مفاهيم الصفوف المجردة

(Abstract classes)

➤ تساعد الصفوف المجردة على البرمجة بطريقة عمومية قابلة لإعادة الاستخدام.

➤ يتم تعريف الصفوف المجردة في C# كما يلي:

```
abstract class myAbs
{
}

class myclass : myAbs
{
}
```

➤ يمكن للصف المجرد أن يتضمن طرائق مجردة، و طرائق كاملة.

➤ يمكن للصف المجرد أن يتضمن خاصيات مجردة، و خاصيات كاملة.

مفاهيم الصفوف المجردة

(Abstract classes)

- يمكن التصريح عن المتحولات (الحقول) ضمن الصف المجرد.
- يمكن تحديد محددات الوصول في الصف المجرد.
- لا يمكن انشاء أغراض من الصف المجرد.
- يمكن للصف العادي أن يرث من صف مجرد واحد فقط، و أكثر من واجهة.
- يمكن للصف المجرد أن يرث من صف اخر، و يمكن أن يرث من واجهات أخرى، و في هذه الحالة يجب أن يطبق الصف المجرد جميع الخصائص و التوابع الموجودة في الواجهة التي ورث منها.
- لا يمكن أبداً أن يكون الصف محدد بكلا الكلمتين abstract و sealed بنفس الوقت، لأنهما مفهومان متناقضان تماماً، حيث الصف المجرد لا يمكن اشتقاق غرض منه، و الصف العقيم لا يمكن الوراثة منه ، و بالتالي لو كان الصف مجرد و عقيم بنفس الوقت فليس له أي فائدة.

مفاهيم الطرق المجردة

(Abstract Method)

- يتم التصريح عن الطريقة المجردة ضمن الصف المجرد بكتابة كلمة `abstract` قبل نمط المعطيات (قبل معرف الوصول أو بعده) .
- الطريقة المجردة تعتبر تلقائياً `virtual` لذلك عند تطبيقها (Implementation) في الصف الابن لا بد من تحديدها كطريقة `override`.
- يمكن تحديد معرف الوصول للطريقة المجردة، و اذا لم يتم تحديده فإنه افتراضياً سيكون `private`.
- عند التصريح عن الطريقة المجردة لا يتم كتابة جسم لها.
- يجب على الصف المشتق تطبيق جميع الطرق المجردة الموجودة في الصف المجرد.
- يتم تعريف الطرق المجردة في `c#` كما يلي:

```
abstract class myAbs
{
    abstract public int myfun(int x,int y);
}

class myclass : myAbs
{
    public override int myfun(int a, int b)
    {
        return a + b;
    }
}
```

مفاهيم الخصائص المجردة

(Abstract properties)

- يتم التصريح عن الخاصية المجردة ضمن الصف المجرد بكتابة كلمة `abstract` قبل نمط المعطيات (قبل معرف الوصول أو بعده) .
- الخاصية المجردة تعتبر تلقائياً `virtual` لذلك عند تطبيقها (Implementation) في الصف الابن لا بد من تحديدها كخاصية `override`.
- يمكن تحديد معرف الوصول للخاصية المجردة، و اذا لم يتم تحديده فإنه افتراضياً سيكون `private`.
- عند التصريح عن الخاصية المجردة لا يتم كتابة التعليمات الخاصة بالـ `get` , `set`.
- يجب على الصف المشتق تطبيق جميع الخصائص المجردة الموجودة في الصف المجرد.
- يتم تعريف الخاصية المجردة في `C#` كما يلي:

```
abstract class myAbs
{
    public abstract string Name { set; get; }
}

class myclass : myAbs
{
    string name;
    public override string Name
    {
        get { return name; } set { name = value; }
    }
}
```

مثال عملي (1)

- ليكن لدينا صف مجرد باسم `math` يتضمن خاصية مجردة لتمثيل متحولين صحيحين الأولى `X` بحيث تكون القيمة المسندة أكبر تماما من الصفر، و الثانية باسم `y`.
- و يتضمن تابع لطباعة الجملة التالية : `Hello, This is abstract class math`. و تابعين مجردين احدهما لإيجاد ناتج طرح عددين و الاخر لإيجاد ناتج مربع عدد.
- و لدينا صف اخر باسم `mycalc` يرث من الصف السابق.
- قم بإنشاء البنية البرمجية السابقة، ثم انشاء غرض من الصف المناسب في التابع الرئيسي `main` مع اسناد قيم و استدعاء التوابع.

حل مثال عملي (1)

```
abstract class math //الصف المجرد
{
    abstract public int X { set; get; }
    abstract public int Y { set; get; }
    public void display()
    {
        Console.WriteLine("Hello, This is abstract class math");
    }
    public abstract int sub(int x, int y);
    public abstract int squ(int x);
}
```

```
class mycalc : math
{
    int x, y;
    public override int X
    {
        set { if (value > 0) x = value; }
        get { return x; }
    }
    public override int Y
    {
        get { return y; }
        set { y = value; }
    }
    public override int sub(int x, int y)
    {
        return x - y;
    }
    public override int squ(int x)
    {
        return x * x;
    }
}
```

حل مثال عملي (1)

```
class Program
{
    static void Main(string[] args)
    {
        mycalc c = new mycalc();
        c.X = 7; c.Y = 2;
        Console.WriteLine("X-Y= " + c.sub(c.X, c.Y)+ "X^2= "+c.squ(c.X));
    }
}
```


مثال عملي (2)

➤ ليكن لدينا صف مجرد باسم shape يعبر عن الاشكال الهندسية ، يحوي على حقل يمثل طول الضلع اسمه x ، و خاصية مجردة Side و طريقة مجردة لحساب مساحة الشكل الهندسي Area و طريقة display تطبع الجملة : shape class is abstract .

➤ و صف آخر يعبر عن المربع square يرث من الصف السابق و يتضمن تابع بانني ذو وسطاء، و يتضمن تابع باسم display يطبع الجملة التالية: square class .

➤ قم بإنشاء البنية البرمجية السابقة، ثم انشاء غرض من الصف المناسب في التابع الرئيسي main مع اسناد قيم و استدعاء التوابع.

حل مثال عملي (2)

```
abstract class shape
{
    int x;
    public abstract int Side { set;
        get;}

    public abstract int Area ();
    virtual public void display()
    { Console.WriteLine("shape class is abstract"); }
}

class squer:shape
{ int side=5;
    public override int Side { set { side = value; } get { return side; } }
    public override int Area()
    {
        return side * side;
    }

    public squer(int side)
    { this.side = side; }

    override public void display() { Console.WriteLine("squer class"); }
}
```

حل مثال عملي (2)

```
class Program
{
    static void Main(string[] args)
    {
        squer s = new squer(5);
        Console.WriteLine( s.Area());
        s.display();
    }
}
```

😊 انتهت المحاضرة 😊

😊 مساحة لاستفساراتكم 😊