جامعة حماه

المعهد التقاني للحاسوب

السنة الأولى

برمجة ٢

محاضرة٧

قسم البرمجيات

عملي

إعداد:

م. رفا البنات

م. أريج فياض

- **التحميل الزائد(overloading):**

هي عبارة عن انشاء أكثر من دالة بداخل نفس ال class بنفس الاسم ، لكن بشروط معينة.

لنفترض أنه توجد دالة اسمها print ولا تأخذ مدخلات ونريد انشاء دالة أخرى اسمها print في هذه الحالة يجب أن نضيف لها مدخل واحد على الأقل مثل print(int x) حتى نميزهاعن الأخرى في وقت الاستدعاء ولا يحدث تشتت للمترجم ، الدالتان بنفس الاسم فكيف يفرق بينهما ؟

فهو يعتبر الدالة التي تستدعيها مجهولة لأنه لم يعرف أي منهما تقصد ، لكن عند اضافة مدخل فحينها سيذهب إلى التي قصدتها أهي التي بمدخل أم لا.

لكن لن نضيف مدخل لن نستخدمه بغرض اضافة دالة جديدة بنفس الاسم فهذا غير صحيح ، نضيف فقط ما تحتاج إليه فعلياً في الكود وما تستخدمه فقط.

أما في حالة أن الدالتين يأخذان مُدخل فيجب أن نغير نوع أحدهما فمثلاً إذا كانت الدالة sum(int x) فيجب أن يكون نوع المدخل في الدالة الجديدة أي نوع غير int .

باختصار لايجب ان يكون هناك دالتان متماثلتان في نفس ال class ، يجب أن يوجد اختلاف في مُدخلات أحدهما على الأقل.

مثال:

لدينا دالة للطباعة تحمل نفس الاسم لكن تحتوي على متحولات مختلفة

```
static void print()
{ Console.WriteLine("hello"); }

static void print(int x)
{ Console.WriteLine(x); }

static void print(double x)
{ Console.WriteLine(x); }

static void print(string x)
{ Console.WriteLine(x); }
```

عند كتابة اسم الدالة للاحظ ظهور تحميل زائد في البرنامج وعددهم ثلاثة، حيث لا تحدث عملية التحميل الزائد لدالة لا تأخذ مدخلات.
عند استدعاء الدالة التي تأخذ مدخلات يجب ان نمرر مدخل من نفس النوع الذي تم تحديده في الدالة

```
        static void Main(string[] args)
         {
             print();
             print(5);
             print(3.14);
             print("sami");
             Console.ReadKey();
         }
```

مثال:

```
class calculator
    {
        public void sum(int x, int y)
        { Console.WriteLine(x + y); }

        public void sum(int x, float y)
        { Console.WriteLine(x + y); }


    }
    class Program
    {


        static void Main(string[] args)
        {
            calculator c1 = new calculator();
            c1.sum(5,6);
            c1.sum(5,6.2f);
            Console.ReadKey();
        }
```

هنا تم استدعاء الدالة المناسبة بحسب نوع المدخلات.

- الوراثة(Inheritance) :

الوراثة بالمعنى العام هو أن يرث الأشخاص بص الصفات والأشياء الخاصة بأبيه او أقاربه، وكذلك الأمر في البرمجة، عندما يرث class من class آخر فإنه يرث منه كل الدوال والمتغيرات والصفات الخاصة بهذا ال class ويصبح ال class الوارث يسمى ابن ، وال class الموروث يسمى أب.

والوراثة هي من أهم مبادئ السي شارب وهي تسهل وتختصر الوقت ، فعندما نرث class ما فإننا نستخدم كل شيء بداخل هذا ال class بدون الحاجة لكتابتها مرة اخرى في ال class الجديد .

لغة ال c# لا تدعم الوراثة المتعددة ، الوراثة المتعددة هي أن يرث ال class الابن من أكثر من class آخر.

وتتم وراثة class ل class آخرعن طريق النقطتين (:) كالتالي

```
Class Base

{  }

Dlass Derived:Base

{ }
```

يمكن لل class الابن رؤية الخواص التي بداخل ال class الأب التي يكون الوصول إليها public او protected أو internal أو protected internal ماعدا ال private فلا يمكن للأعضاء الوصول إليها.

protected: (وصول محمي) أي دوال الصف نفسه ودوال الصف الابن (أي الصف المشتق) تستطيع الوصول إلى البيانات ويرث الأعضاء الدالية.

مثال :

بناء الشجرة الهرمية ل point ونشتق منها صف الدائرة circle حيث ال point له إحداثيات (x && y)

وال circle له إحداثيات (x&&y&&r) حيث r هي نصف القطر .

حيث ترث الدائرة إحداثيات الأب وتبني لنفسها نصف القطر .

المطلوب: وراثة دائرة ل point وبناء كلاس الدائرة مع خصائص get&&set

وحساب مساحة الدائرة ومحيطها.

```
namespace ConsoleApplication1
{    class point
         {   private int x;
             private int y;
     public point()
  { x = 0;     y = 0;  }
```

```
public point(int x, int y)
{       this.x = x;
        this.y = y;
}
public int X//get&&set خصائص
{
get { return this.x; }
set { this.x = value; }
}
public int Y//get&&set خصائص
{
get { return this.y; }
set { this.y = value; }
}
public void printe()
{   Console.WriteLine("(" + x + "," + y + ")" ); }
}//end point
class circle :point
{
public int r;
public circle()
{
this. r = 0;
}//constroctor
public circle( int x, int y,int r):base(x,y)
{     this.r = r;
}
public int R//get&&set خصائص
{
get{return this.r;}
set{this.r=value;}
}
public void print()
{
Console.WriteLine("(" +base.X + "," + base.Y + ")" + "r=" + r);// طباعة احداثيات الدائرة
}
public double mohet()//المحيط
{return 2*3.14*r;}
public double s()//المساحة
{return 3.14*r*r;} }
class Program
{
static void Main(string[] args)
{
point b = new point(3,2);
b.printe();
circle m = new circle();
Console.WriteLine("enter x");
int a = Int32.Parse(Console.ReadLine());
Console.WriteLine("enter y");
int b = Int32.Parse(Console.ReadLine());
Console.WriteLine("enter r");
int c = Int32.Parse(Console.ReadLine());
circle s = new circle(a, b,c);
```

```
(3,2)
enter x
5
enter y
6
enter r
2
(5,6)r=2
mhet=12.56
s=12.56
```

```
s.print();
Console.WriteLine("mhet="+s.mohet());
Console.WriteLine("s=" + s.s());
Console.ReadLine();
```

مثال:

اكتب كلاس باسم person له (اسم ، عمر ، تاريخ ميلاد) والمطلوب:

باني بوسطاء وباني بلا وسطاء مع تعليمة طباعة بكل باني.

انشاء كلاس جديد employee يرث من كلاس person (اسم ، عمر ، تاريخ ميلاد) بالإضافة إلى عدد ساعات العمل ،باستخدام باني بوسطاء وباني بلا وسطاء وتعليمة طباعة بكل باني وطريقة لحساب راتب الموظف.

```
class person
{
   protected int length;
    protected int age;
    protected string name;
    protected int date;

    1 reference
    public person()
    {
        length = 9;
        age = 30;
        name = "aa";
        date = 1996;
        Console.WriteLine("start the father person 1");
    }
    1 reference
    public person(int l,int a, string n,int d)
    {
        length = l;
        age = a;
        name = n;
        date = d;
       Console.WriteLine("start the body  person 2");
    }
}
```

```
6 references
class employee:person
{
    public int hour;

    1 reference
    public employee()
    { hour = 200;
        Console.WriteLine("start of employee1");
    }

    1 reference
    public employee(int h)
    {
        hour = h;
        Console.WriteLine("start the hour");
    }
    // تابع لحساب راتب الموظف
    2 references
    public int salary()
    {
        int s = hour * 1000;
        return s;
    }
}
0 references
class Program
{
    0 references
    static void Main(string[] args)
    {
        person p = new person();
        person p1 = new person(8, 9, "bb", 1999);

        employee e = new employee();
        Console.WriteLine(e.salary());
        employee e1 = new employee(6);
        Console.WriteLine(e1.salary ());
    }
}
}
```

```
C:\Windows\system32\cmd.exe
start the father person 1
start the body  person 2
start the father person 1
start of employee1
200000
start the father person 1
start the hour
6000
Press any key to continue . . .
```