# الفئات ( الأصناف ) Classes:

يعتبر مفهوم الفئات واحد من أفضل ميزات لغة C#،

الفئة أو الكلاس هي عبارة عن مفهوم مجرد يُستخدم لتعريف نمط معطيات جديد، ويمكن أن يضم مجموعة من البيانات تسمى حقول (fields) ومجموعة من الدوال (methods) التي تعمل على هذه البيانات، بالإضافة إلى مجموعة من الخصائص (Properties) التي تطبق على حقول الفئة.

يبدأ تعريف الكلاس بالكلمة المحجوزة class يليها اسم اختياري للصف (يخضع لقاعدة تعريف الأسماء ). يتم تحديد جسم الكلاس بواسطة قوسين { }، نعرّف ضمن هذين القوسين جميع أعضاء الكلاس.

**التصريح عن الكلاس ( الفئة ):** يتم التصريح عن الكلاس بالشكل التالي:

```
class class_name
{
    access_modifier  dataType   var1 ;
    access_modifier  dataType   var2 ;
   ..............
   access_modifier  dataType   Property1
   { get { return var ; }
     set {  var = value; }
    }
   ..............
   access_modifier  return_value  method_name  (parameters )
   {
     ........
   }

}
```

حيث:

class_name : اسم الكلاس .

access_modifier : محدد الوصول لمتحولات وطرائق الكلاس.

dataType var1 : نوع واسم الحقل ( المتحول ) الأول من متحولات الكلاس.

dataType var2 : نوع واسم الحقل ( المتحول ) الثاني من متحولات الكلاس.

.....

تجدر الإشارة إلى أن حقول الكلاس يمكن أن تكون من أي نوع ويمكن أن تكون أغراض مشتقة من صفوف أخرى أيضاً.

## محدد الوصول ( Access Modifier ) :

يحدد محدد الوصول طريقة التعامل مع أي عضو من أعضاء الكلاس، ومن أهم محددات الوصول:

| | |
|---|---|
| **Private** | أعضاء الكلاس التي نحدد محدد الوصول إليها كـ Private يتم التعامل معها فقط في الكلاس المُعرّفة ضمنه ولا يمكن الوصول إليها أبدا خارج هذا الكلاس، وهذا المحدد هو المحدد الافتراضي. |
| **Public** | أعضاء الكلاس التي نحدد محدد الوصول إليها كـ Public يمكن التعامل معها في الكلاس المعرّفة ضمنه وكذلك يمكن التعامل معها خارج هذا الكلاس. |

كما ذُكر سابقاً فإن الكلاس هو نمط معطيات، لذلك نحتاج لتعريف متحولات نوعها هو هذا النمط، هذه المتحولات يُطلق عليها اسم أغراض ( **objects** )،

نُعرّف الغرض( object ) بالشكل التالي:

*class_name object_name* = **new** *class_name*( ) ;

حيث:

*class_name* هو اسم الكلاس.

*object_name* هو اسم الغرض *المُشتق* من الكلاس.

## الخصائص (Properties):

نستخدمها للوصول إلى الحقول المُعرّفة كـ Private، وتُعرّف بالشكل العام التالي:

```
اسم الخاصة  نوع الخاصة  محدد الوصول
{
 get
  { .....  }
 set
  { .....  }
}
```

Get تقوم بإعادة القيمة الحالية لعضو الكلاس وذلك عن طريق تعليمة return.

Set تقوم بإسناد قيمة جديدة لعضو الكلاس وذلك عن طريق الكلمة المحجوزة value.

**التابع الباني (Constructor):** هو نوع خاص من الطرق ( methods ) يتميز بأنه:

- له نفس اسم الكلاس.
- لا يعيد أي قيمة
- نستخدمه لإعطاء قيم ابتدائية لحقول الكلاس.
- يتم استدعاؤه ضمنياً وبشكل تلقائي عند انشاء غرض من هذا الكلاس.

**التابع الهادم (Destructor):** عكس التابع الباني يتميز بأنه:

- له نفس اسم الصف مسبوقاً بـ ~.
- لا يعيد أي قيمة.
- لا يأخذ أي وسيط.
- يتم استدعاؤه ضمنياً وبشكل تلقائي عند الانتهاء من استخدام الغرض.
- من النادر استخدامه لأنه في C# يتم تحرير الذاكرة تلقائياً عن طريق ما يسمى بـ جامع النفايات ( Garbage Collector ).

## مفهوم التحميل الزائد ( OverLoading):

المقصود به إمكانية تعريف أكثر من طريقة أو دالة بنفس الاسم ضمن المجال نفسه ويكون الاختلاف **بعدد أو نوع** البارمترات (الوسائط) الممررة لهذه الدالة.

تطبيقاً لهذا المفهوم يمكن تعريف أكثر من تابع باني في نفس الكلاس.

## الأعضاء الساكنة Static:

يتم الوصول إليها عن طريق اسم الكلاس مباشرة وليس عن طريق أي غرض من الأغراض المعرّفة من هذا الكلاس، لأن هذه الأعضاء تكون مشتركة بين جميع أغراض الكلاس.

**مثال :**

```
namespace  classes_example
{
    class  person
    {
        public  string  fname;
        public  string  lname;
        private  int      age;
        public  string  address;

        public   int      Age
        {
          get  { return  age  ; }
```

```
            set {
              if  (value > 0 && value < 120)
               { age = value; }
                else age = 0;
                }
        }

        //constructors
        // باني بدون بارمترات
        public  person()
        {
            fname  = "XX";
            lname  = "YY";
            age  =  20;
            address  =  "Hama";
            Console.WriteLine ( "Constructor Person" ) ;
        }
        // باني ببارامترين
        public  person ( string  fname ,  string  lname )
        {
            this.fname  =  fname;
            this.lname  =  lname;
            this.age  =  20;
            address  =  "Hama";
        }
        // باني بثلاثة بارمترات
        public person( string  fname, string lname, int Age )
        {
            this.fname  =  fname;
            this.lname  =  lname;
            this.Age  =  Age;
            address  = "Hama";
        }

       public  void  printinfo( )
        {
          Console.WriteLine(" Hi, I am Person >>>");
          Console.WriteLine(" Name is {0} {1},\n Age is
            {2},\n Address is {3} ", fname,lname, Age,
                                    address);
          Console.WriteLine ( "-----------------------" );
        }
    }

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

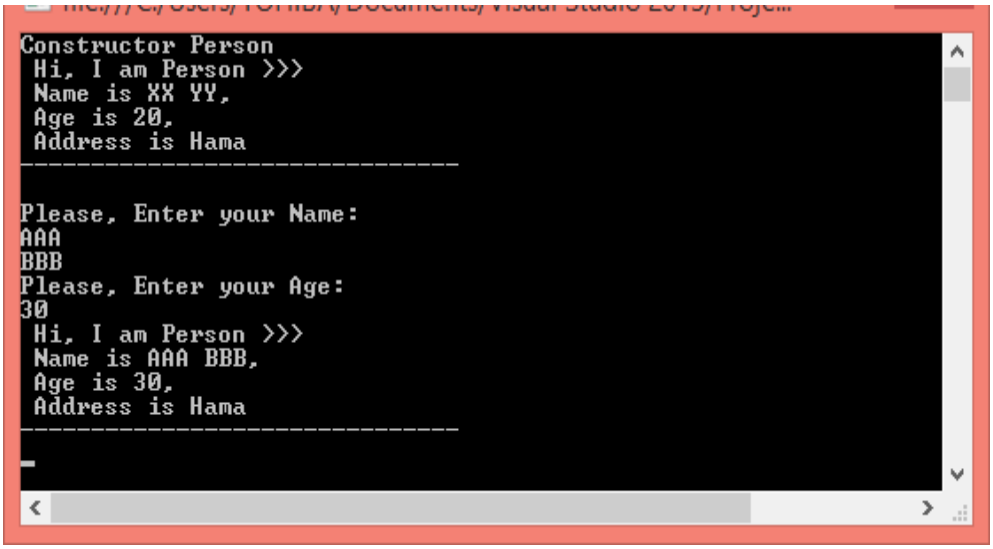
```
        {
            person p1 =  new person( );
            p1.printinfo( );

            Console.WriteLine( );
            Console.WriteLine( "Please, Enter your Name:" );
            string  p_first_name  = Console.ReadLine( );
            string  p_last_name  =  Console.ReadLine( );
            Console.WriteLine( "Please, Enter your Age:" );
            int  p_age  =  int.Parse (Console.ReadLine( ) );

            person p2  = new person (p_first_name,
                        p_last_name, p_age);
            p2.printinfo( );
            Console.ReadKey();

        }
    }
}
```

نتيجة التنفيذ:



**مثال:**

لدينا الكلاس CheckingAccount يقوم بتدقيق حساب مصرفي:

١- حقول هذا الكلاس: هي رقم الحساب account_num، اسم صاحب الحساب account_name، الرصيد الحالي balance .

٢- طرق الكلاس هي: طريقة لمعرفة الرصيد الحالي getBalance().

طريقة لسحب مبلغ get_money(amount).

طريقة لإيداع مبلغ deposit(amount).

```
class CheckingAccount
    {
      public string account_num;
      public string account_name;
      public double balance;

      //باني بدون بارامترات
      public CheckingAccount()
       {
         balance = 5000;
       }

      //باني بثلاثة بارامترات
      public CheckingAccount(string account_num, string
            account_name, double balance)
       {
         this.account_num = account_num;
         this.account_name = account_name;
         this.balance = balance;
       }

       //طريقة لمعرفة الرصيد الحالي
       public double getBalance()
       {
           return balance;
       }

       //طريقة لإيداع مبلغ
       public void deposit(double amount)
       {
           balance = balance + amount;
       }

       //طريقة لسحب مبلغ
```

```
   public void get_money (double amount)
    {
     //تتم عملية السحب بعد التأكد أنها أقل من المبلغ الموجود
       if (balance > amount)
            balance = balance - amount;
    }

// طريقة لطباعة معلومات الرصيد
public void display()
 {
   Console.WriteLine("**********************");
   console.WriteLine("Your name is {0},\n Your account number
     is {1},\n Your balance is {2}." , account_name,
                 account_num , balance);

   Console.WriteLine("**********************");

 }
}

class Checking {
  public static void Main(string[] a)
   {
    CheckingAccount c1 = new CheckingAccount("123","Sara",100000);
      c1.display();  // طباعة معلومات الرصيد
      c1.deposit(20000);//إيداع مبلغ
      c1.get_money(15000); //سحب مبلغ
      c1.getBalance();//معرفة الرصيد الحالي
      c1.display();// طباعة معلومات الرصيد

    CheckingAccount c2 = new CheckingAccount();
      Console.Write(" Please, Enter your account number:");
      c2.account_num = Console.ReadLine();
      Console.Write(" Please, Enter your account name:");
      c2.account_name  = Console.ReadLine();
      c2.display();
      Console.Write(" How much money you want to deposit?");
      double amount1 = double.Parse(Console .ReadLine());
      c2.deposit(amount1);//مبلغ إيداع
      Console.WriteLine("Your balance is: {0} ",
                 c2.getBalance());

      Console.Write(" How much money you want to get?");
      double amount2 = double.Parse(Console.ReadLine());
      c2.get_money(amount2);

      c2.display();
```

```
        Console.ReadKey();
    }

}
```

نتيجة التنفيذ ستكون بالشكل التالي:

```
**********************
Your name is Sara,
Your account number is 123,
Your balance is 100000.
**********************
**********************
Your name is Sara,
Your account number is 123,
Your balance is 105000.
**********************
 Please, Enter your account number:45678
 Please, Enter your account name:Ahmad
**********************
Your name is Ahmad,
Your account number is 45678,
Your balance is 5000.
**********************
 How much money you want to deposite?60000
Your balance is:65000
 How much money you want to get?10000
**********************
Your name is Ahmad,
Your account number is 45678,
Your balance is 55000.
**********************
```