

## C# في Record و Struct و Class بين الفرق

### 1. Class (الكلاس)

- ✓ Heap الـ في يتخزن → (Reference Type) مرجعي نوع
- ✓ جديدة نسخة وليس (العنوان) المرجع تمرير يتم كلاس من كائن تمرير عند
- ✓ (Polymorphism) الأشكال وتعدد (Inheritance) الوراثة يدعم
- ✓ والمعقدة الكبيرة الكائنات مع يُستخدم

مثال:

```
class Person
{
    public string Name { get; set; }
    public int Age { get; set; }
}
```

### 2. Struct (الاستراكت)

- ✓ Stack الـ في يتخزن → (Value Type) قيمي نوع
- ✓ المرجع وليس البيانات نسخ يتم تمريره عند
- ✓ (Interfaces) يطبق ممكن لكن الوراثة يدعم لا
- ✓ الصغيرة البيانات مع الأداء في أسرع
- ✓ (الإحداثيات) مثل البسيطة للبيانات يُستخدم

مثال:

```
struct Point
{
    public int X { get; set; }
    public int Y { get; set; }
}
```

### 3. Record (الريكورد)

- ✓ أيضًا record struct ويدعم الكلاس مثل (Reference Type) مرجعي نوع

✓ (Immutable) الثابتة البيانات مع للعمل مخصص ✓

افتراضياً (Value-based Equality) بالقيمة تكون المقارنة ✓

البيانات ونقل DTOs الـ في غالباً يُستخدم ✓

الكلاس مثل الوراثة يدعم ✓

مثال:

```
record Person(string Name, int Age);
```

```
var p1 = new Person("Ali", 30);
```

```
var p2 = new Person("Ali", 30);
```

```
Console.WriteLine(p1 == p2); // True (بالقيمة مقارنة)
```

### مقارنة جدول

الميزة	Class	Struct	Record
التخزين نوع	مرجعي (Heap)	قيمي (Stack)	مرجعي (Heap)
المساواة طريقة	مرجعية (Reference)	بالقيمة (Value)	بالقيمة (Value)
الوراثة	نعم	(انترفيس إلا) لا	نعم
الاستخدام	ومعقدة كبيرة كائنات	صغيرة بيانات	DTOs / Immutable
التغيير إمكانية	Mutable	Mutable	افتراضياً Immutable

### الفرق يوضح عملي مثال

```
class C { public int X { get; set; } }
```

```
struct S { public int X { get; set; } }
```

```
record R(int X);
```

```
var c1 = new C { X = 5 };
```

```
var c2 = new C { X = 5 };
```

```
Console.WriteLine(c1 == c2); // False (مرجع مقارنة)
```

```
var s1 = new S { X = 5 };
```

```
var s2 = new S { X = 5 };
```

```
Console.WriteLine(s1.Equals(s2)); // True (قيمة مقارنة)
```

```
var r1 = new R(5);  
var r2 = new R(5);  
Console.WriteLine(r1 == r2); // True (قيمة مقارنة)
```