

مقایسه خواص عمومی يك كلاس در C++ و C#

قبل از بررسی تفاوت بین کلاسها بین C++ و C# ابتدا خاصیت های عمومی يك كلاس را بررسی کرده سپس به بررسی تفاوت بین این دو زبان قدرتمند خواهیم پرداخت.

تعریف. کلاس در حالت ساده مجموعه ای از خواص(attributes) و روش ها (methodes) است که در رابطه با هم هدف مشترکی را دنبال می کنند و خدماتی را ارائه می کنند. البته این تعریف در حد يك تعریف علمی می باشد ولي ذکر این نکته ضروري است که اکثر تعاریف در شی گرایی در طبیعت وجود دارد و در ساده ترین حالت برای تعریف يك كلاس می توان از تعریف انسان استفاده کرد که این بحث در این مقاله نمی گنجد. در هر حال ما به تعریف فوق اکتفا کرده و بحث را ادامه می دهیم.

خواص ها و روش ها که در تعریف کلاس ارائه شد می تواند نحوه دسترسی متفاوتی مانند public و private و protected داشته باشد که برای برنامه نویسان C++ نیازی به توضیح نیست.

تفاوت در تعریف يك كلاس:

در C# يك كلاس در يك برنامه در داخل يك لایه تعریف خواهد شد(اجباری نیست). تعریف يك كلاس در C# تفاوتی با C++ دارد که سعی خواهیم کرد این تفاوت ها را با چند مثال توضیح دهیم.

فرض کنید کلاس زیر در C++ تعریف شده است که این کلاس را به يك كلاس C# تبدیل خواهیم کرد.

```
// Test.h
```

```
class CTest{  
  
    private:  
  
        int m_at1;  
  
        int f1();  
  
    public:  
  
        float m_at2;  
  
        int f2();  
  
        CTest();  
  
        ~CTest();  
  
};
```

```

// Test.cpp

#include "test.h"

CTest::CTest(){

    m_at1 = 0;

    m_at2 = 0.0;

}

CTest::~~CTest(){

    // cout << "destructor call";

}

int CTest: f1(){

    return m_at1;

}

int CTest: f2(){

    return m_at1+(int)m_at2;

}

```

تفاوت اول:

بر خلاف C++ در C# تعریف و نحوه عملکرد يك تابع يا Method در خود تعریف کلاس قرار داده مي شود. در C++ غالبا تعریف کلاس در فایل h. و بدنه توابع در فایل cpp. قرار مي گرفت. البته ذکر اين نکته ضروري است که مي توان همين عمل را در h. نیز قرار داد ولي در يك برنامه بزرگ اين عمل کنترل برنامه را از دست برنامه نويس خارج کرده و همچنين مشکلات ديگري را براي برنامه نويسان ايجاد مي کند که برنامه نويسان C++ با اين مشکلات آشنا هستند.

تفاوت دوم:

بر خلاف C++ در C# يك دسته از متغيرها را نمي توان بصورت public يا private و ... تعريف کرد و براي هر متغير بايد نحوه دسترسي به آن نیز مشخص شود.

تفاوت سوم:

بر خلاف C++ در C# تابع destructor وجود ندارد و خود C# مسئول از بین بردن يك شي مي باشد كه اين از بين بردن با توجه به محدوده تعريف اين شي انجام مي شود. البته تعريف destructor وجود دارد ولي نمي توان destructor را فراخواني كرد. در C++ فراخواني Destructor با استفاده از عملگر delete قابل انجام بود ولي چون C# مدیریت object ها را خود بعهده مي گيرد شي ايجاد شده نميتواند توسط برنامه خراب شود.

با توجه به تفاوتهاي ذكر شده مي توان اين كلاس را به صورت زير براي يك كلاس C# ارائه كرد.

```
// Test.cs

using System;
namespace NS
{
    publicclass CTest
    {
        privateint m_at1;

        public float m_at2;

        privateint f1()
        {
            return m_at1;
        }

        public int f2()
        {
            return m_at1+(int)m_at2;
        }

        public CTest()
```

```
{  
    m_at1 = 0;  
    m_at2 = 0.0f;  
}  
~Test()  
{  
    // Console.WriteLine("destructor call");  
}  
}  
}
```