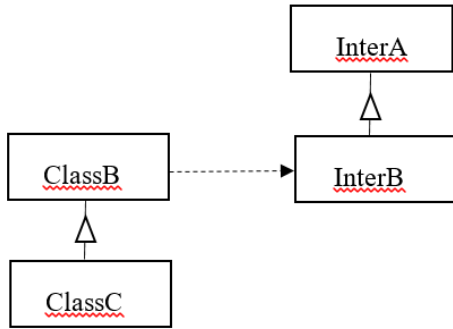


## משקים והורשה



אם ממשק A יורש ממשק B ומחלקה B מממשת את ממשק B אז המחלקה B צריכה לממש את כל הפעולות המוגדרות בשני הממשקים.

אם מחלקה C יורשת ממחלקה B (מחלקה C לא התחייבה לממש את הממשקים)

אז מחלקה C יכולה להשתמש בפעולות הממשק שמומשו במחלקה B ממנה היא ירשה.

```

class ClassA : InterA
{
    public ClassA(string st)
    {
        Console.WriteLine(st + " object of ClassA");
    }

    public void PrintA(string st)
    {
        Console.WriteLine(st + " PrintA() from CalssA");
    }
}
    
```

```

interface InterA
{
    void PrintA(string st);
}

interface InterB : InterA
{
    void PrintB(string st);
}
    
```

```

class ClassB : InterB
{
    public ClassB(string st)
    {
        Console.WriteLine(st + " object of ClassB");
    }

    public void PrintA(string st)
    {
        Console.WriteLine(st + " PrintA() from CalssB");
    }

    public void PrintB(string st)
    {
        Console.WriteLine(st + " PrintB() from CalssB");
    }
}

class ClassC : ClassB
{
    public ClassC(string st): base(st)
    {
        Console.WriteLine(st + " object of ClassC");
    }
}
    
```

```

static void Main(string[] args)
{
    ClassA ca = new ClassA("ca"); Console.WriteLine();
    ClassB cb = new ClassB("cb"); Console.WriteLine();
    ClassC cc = new ClassC("cc"); Console.WriteLine();

    Console.WriteLine();           ca object of ClassA
    ca.PrintA("ca");               cb object of ClassB
    cb.PrintA("cb");               cc object of ClassB
    cb.PrintB("cb");               cc object of ClassC
    cc.PrintA("cc");
    cc.PrintB("cc");               ca PrintA() from CalssA
                                   cb PrintA() from CalssB
                                   cb PrintB() from CalssB
                                   cc PrintA() from CalssB
                                   cc PrintB() from CalssB
}
    
```

כלומר - קיים יחס הורשה בין ממשקים. המשמעות: ממשק-A מגדיר את הפעולות שלו. ממשק-B מרחיב (יורש) את ממשק-A. ממשק-C יורש מממשק-B. בכל אחד מהממשקים כתובות רק כותרות הפעולות שהם מגדירים. מחלקה המממשת את ממשק אחד תצטרך לממש רק את הפעולות של ממשק זה. מחלקה המממשת את ממשק שמרחיב ממשק אחר תצטרך לממש את כל הפעולות של ממשק-העל ושל תת-הממשק.

```

interface InterC : InterB
{
    void PrintC(string st);
}

class ClassD : InterC
{
    public void PrintC(string st)
    {
        Console.WriteLine(st + " PrintC() from CalssD");
    }
}
    
```

אבל - עלולה להיווצר פה בעיה: מחלקה-D (מחלקה עצמאית, ללא הורשה) המתחייבת לממש את ממשק-C, רואה בממשק-C רק את הפעולות שהוגדרו בממשק זה, איך היא אמורה לדעת מהן הפעולות בממשק-B ושל ממשק-A?

- 1 'InheritanceInterfaces.ClassD' does not implement interface member 'InheritanceInterfaces.InterA.PrintA(string)'
- 2 'InheritanceInterfaces.ClassD' does not implement interface member 'InheritanceInterfaces.InterB.PrintB(string)'