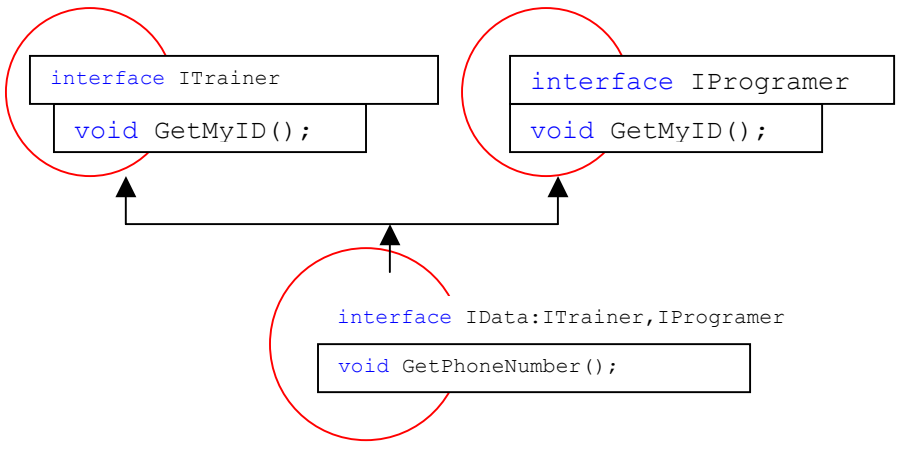


ירושה מרובה בין ממשקים

Interface : מטרתה להגדיר התנהגות עבור המחלקות הנגזרות, לא ניתן להגדיר משתנים או לממש פונקציות ב interface כל הפונקציות המוגדרות בו הן אבסטרקטיות בשונה ממחלקה אבסטרקטית שיכולה להכיל משתנים וגם פונקציות שאינן אבסטרקטיות

Interface : ניתן לרשת ממספר Interfaces ובכך ליישם ירושה מרובה (ב C# לא ניתן לרשת באופן ישיר ממספר מחלקות אפשר לרשת באופן ישיר רק מחלקה אחת) ניתן לגזור Interface ממספר interfaces (לא ניתן ב COM וגם אין צורך בקובץ IDL)



Employee: Student, ITrainer, IProgrammer

```

Employee empObj=new Employee();
empObj.GetMyID();

ITrainer empIT=empObj;
empIT.GetMyID();

IProgrammer empIP=(IProgrammer)empObj;
empIP.GetMyID();
  
```

יצירת משתנה מיוחס מסוג המחלקה יפיל את המתודה אשר הוגדרה במחלקה יצירת משתנה מיוחס מסוג הממשק הנבחר יפעיל את מימוש המתודה לפי סוג הממשק

Worker: Student, IData

```

Worker wrkObj=new Worker();
wrkObj.GetMyID();

ITrainer ITwrk = wrkObj;
ITwrk.GetMyID();

IProgrammer IPwrk= (IProgrammer)ITwrk;
IPwrk.GetMyID();

IData IDwrk = wrkObj;
IDwrk.GetPhoneNumber();
  
```

```
((ITrainer) IDwrk).GetMyID();
((IProgramer) IDwrk).GetMyID();
```

ממשק IDATA יורש שני ממשקים
ניתן להמיר טיפוס מסוג IDATA לאחד משני
הממשקים הנורשים ועל ידי כך הפעיל את
המימוש המותאם לממשק הנבחר

```
namespace ThreeInterfaces
{
```

```
using System;
abstract class Student
{
    abstract public void GetMyID();
}
```

הגדרת מחלקה אבסטרקטית
+ הגדרת מתודה אבסטרקטית
(חובה לדרוס אותה)

```
interface ITrainer
{
    void GetMyID();
```

הגדרת Interface

```
interface IProgramer
{
    void GetMyID();
```

הגדרת Interface

```
interface IData:ITrainer,IProgramer
{
    void GetPhoneNumber();
```

הגדרת Interface הנגזר משני
Interfaces קודמים.

```
class Employee: Student,ITrainer,IProgramer
{
```

```
void ITrainer.GetMyID()
{
    Console.WriteLine("Employee-Trainer ID=E-1-11");
}
void IProgramer.GetMyID()
{
    Console.WriteLine("Employee-Programer ID=E-2-22");
}
public override void GetMyID()
{
    Console.WriteLine("Employee-Student ID=E-3-33");
}
}
```

מחלקה היורשת את
המחלקה הבסיסית
וממשת את המתודה
שלה כולל מימוש של
הממשקים כאשר על
ממשק הוא בעל מימוש
שונה לאותה מתודה

```
class Worker:Student,IData
{
```

```
void ITrainer.GetMyID()
{
    Console.WriteLine("Worker-Trainer ID=W-1-11");
}
void IProgramer.GetMyID()
{
    Console.WriteLine("Worker-Programer ID=W-2-22");
}
public override void GetMyID()
{
    Console.WriteLine("Worker-Student ID=W-3-33");
}
}
```

מחלקה היורשת את
המחלקה הבסיסית
וממשת את המתודה
שלה כולל מימוש של
הממשקים כאשר על
ממשק הוא בעל מימוש
שונה לאותה מתודה

```
void IData.GetPhoneNumber()
{
```

```
Console.WriteLine("Phone Number worker:000-000000");
```

```
    }  
}  
  
public class MainClass  
{  
  
    public static void Main(string[] args)  
    {  
        Console.WriteLine("Employee: Student, ITrainer, IProgramer");  
        Console.WriteLine("-----\n");  
  
        Employee empObj=new Employee();  
  
        empObj.GetMyID();  
  
        ITrainer empIT=empObj;  
        empIT.GetMyID();  
  
        IProgramer empIP=(IProgramer)empObj;  
        empIP.GetMyID();  
  
        Console.WriteLine("\n\nWorker: Student, IData");  
        Console.WriteLine("-----\n");  
  
        Worker wrkObj=new Worker();  
        wrkObj.GetMyID();  
  
        ITrainer ITwrk = wrkObj;  
        ITwrk.GetMyID();  
  
        IProgramer IPwrk= (IProgramer)ITwrk;  
        IPwrk.GetMyID();  
  
        IData IDwrk = wrkObj;  
        IDwrk.GetPhoneNumber();  
  
        ((ITrainer) IDwrk).GetMyID();  
        ((IProgramer) IDwrk).GetMyID();  
  
        Console.WriteLine("\n\n-----");  
  
    }  
}
```

משתנה מטיפוס המחלקה
יפנה למימוש המתודה
שהוגדר המחלקה

טיפוס מסוג ממשק שגורש
יפעיל את המימוש שהוגדר
עבור אותו ממשק

ניתן לבצע המרה מפורשת (לא חובה במצב זה)

הממשק IDATA נורש משני
הממשקים האחרים לכן נוכל
לבצע המרה של הטיפוס
לטיפוס הממשק האחר ובכך
לפנות למימוש שונה של
אותה מתודה

יש לבצע המרה מפורשת (חובה במצב זה)

המרה של הטיפוס לסוג אחר

הפעלת מימוש המתודה לפי
סוג הממשק המצביע

```
Result:  
Employee: Student,ITrainer,IProgramer  
-----  
Employee-Student ID=E-3-33  
Employee-Trainer ID=E-1-11  
Employee-Programer ID=E-2-22  
Worker: Student,IData  
-----  
Worker-Student ID=W-3-33  
Worker-Trainer ID=W-1-11  
Worker-Programer ID=W-2-22  
Phone Number worker:000-000000  
Worker-Trainer ID=W-1-11  
Worker-Programer ID=W-2-22  
-----  
Press any key to continue
```