

## משקים

נכון / לא נכון

נתונים הממשקים והמחלקות המממשות אותם:

```
public interface InterA
{
    int Fa();
}

public interface InterB
{
    void Fb();
}

class A : InterA
{
    protected char ch;
    protected int x;

    public A(char ch, int x)
    {
        this.ch = ch;
        this.x = x;
    }

    public virtual void Print() { Console.WriteLine("Print() from A: " + Fa()); }
    public virtual int Fa() { return this.x; }
    public override string ToString() { return ch + ": x: " + this.x; }
}

class B : A, InterB
{
    protected int y;

    public B(char ch, int x, int y): base(ch, x)
    {
        this.y = y;
    }

    public override void Print() { Console.WriteLine("Print() from B: " + Fa()); }
    public override int Fa() { return base.Fa() + this.y; }
    public virtual void Fb() { Console.WriteLine("Fb() From B: " + (this.x * this.y)); }
    public override string ToString() { return base.ToString() + " y: " + this.y; }
}

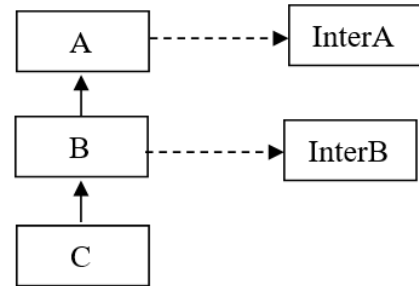
class C: B
{
    protected string str;

    public C(char ch, int x, int y, string str) : base(ch, x, y)
    {
        this.str = str;
    }

    public override void Print() { Console.WriteLine("Print() from C: " + Fa()); }
    public override void Fb()
    {
        string st = "";
        for (int i = 0 ; i < this.x-1 ; i++)
            st += this.str + "-";
        st += this.str;
        Console.WriteLine("Fb() From C: " + st);
    }

    public override string ToString() { return base.ToString() + " str: " + this.str; }
    public void G() { Console.WriteLine("G() of C"); }
}

```



להלן קטע תכנית עם הגדרות נכונות ושגויות:

```
static void Main(string[] args)
{
    A [] arr = new A [5];
    arr[0] = new A('A', 5);
    arr[1] = new B('B', 2, 6);
    arr[2] = new C('C', 3, 7, "Xo");
    arr[3] = new InterA();
    arr[4] = new InterB();

```

המחלקות B ו-C הן סוג של A ולכן ניתן ליצור הפניה מסוג A על עצמים ממחלקות אלו

שגיאה! לא ניתן ליצור עצמים מסוג ממשק

```
InterA [] arA = new InterA [4];
arA[0] = new A ('A', 10);
arA[1] = new B ('B', 12, 14);
arA[2] = new C ('C', 4, 15, "Yo");
arA[3] = new InterA();

```

ניתן ליצור הפנייה מטיפוס הממשק

המחלקה A מממשת את InterA, מכאן שפעולות הממשק מוגדרות אצל כל המחלקות. (חלקן יכולות להגדירן מחדש).

שגיאה! לא ניתן ליצור עצמים מסוג ממשק

```
InterB [] arB = new InterB [3];
arB[0] = new A ('A', 10);
arB[1] = new B ('B', 12, 14);
arB[2] = new C ('C', 4, 15, "Yo");

```

שגיאה! המחלקה A אינה מממשת את InterB

B מממשת את InterB. C קיבלה בירושה את פעולות הממשק

```
arB[2].G();
((C)arB[2]).G();

```

שגיאה! פעולות של העצם C אינן גלויות עבור הפניה מטיפוס הממשק

כדי להפעיל את הפעולות שמוגדרות במחלקה של העצם יש לבצע downcasting (המרה כלפי מטה)

מערך arr מטיפוס המחלקה A. כל העצמים יורשים מ-A ומכאן שהם מממשים את הממשק InterA:

```
for (int i = 0 ; i < arr.Length ; i++)
    if (arr[i] != null)
    {
        Console.WriteLine(arr[i].GetType() + " : arr[" + i + "] : ");
        arr[i].Print();
    }

```

לאחר הסרת שורות ההשמה השגויות נותרו חלק מתאי המערך ריקים, כלומר null.

```
InterfaceTrueFalse.A: arr[0] : Print() from A: 5
InterfaceTrueFalse.B: arr[1] : Print() from B: 8
InterfaceTrueFalse.C: arr[2] : Print() from C: 10
```

העצם ב-arr[0] הוא מטיפוס A שאינו מממש את המחלקה A ולכן לא יוצג. עבור העצמים שכן מממשים את הממשק, יש לבצע המרה לטיפוס הממשק (או לטיפוס המחלקה הגבוה ביותר בעץ ההורשה שמממש את InterB וכולם יורשים ממנו, כלומר לטיפוס B)

```
for (int i = 0 ; i < arr.Length ; i++)
    if (arr[i] is InterB)
    {
        Console.WriteLine(arr[i].GetType() + " : arr[" + i + "] : ");
        ((InterB)arr[i]).Fb();
    }

```

תא המכיל null ממילא אינו מסוג הממשק

```
InterfaceTrueFalse.B: arr[1] : Fb() From B: 12
InterfaceTrueFalse.C: arr[2] : Fb() From C: Xo-Xo-Xo
```