

## משקיים

נכון / לא נכון

נתונים הממשקים והמחלקות הממשות אותם:

```

public interface InterA
{
    public int fa();
}

public interface InterB
{
    public void fb();
}

public class A implements InterA
{
    protected char ch;
    protected int x;

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

    public void print () { System.out.println("print() from A: " + fa()); }
    public int fa()      { return this.x; }
    public String toString () { return ch + ": x: " + this.x; }
}

public class B extends A implements InterB
{
    protected int y;

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

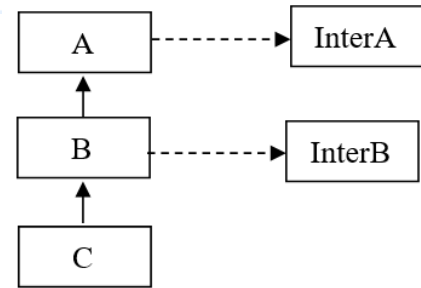
    public void print () { System.out.println("print() from B: " + fa()); }
    public int fa()      { return super.fa() + this.y; }
    public void fb()     { System.out.println("fb() From B: " + (this.x * this.y)); }
    public String toString () { return super.toString() + " y: " + this.y; }
}

public class C extends B
{
    protected String str;

    public C(char ch, int x, int y, String str)
    {
        super(ch, x, y);
        this.str = str;
    }

    public void print () { System.out.println("print() from C: " + fa()); }
    public void fb()
    {
        String st = "";
        for (int i = 0 ; i < this.x-1 ; i++)
            st += this.str + "-";
        st += this.str;
        System.out.println("fb() From C: " + st);
    }

    public String toString () { return super.toString() + " str: " + this.str; }
    public void g () { System.out.println("g() of C"); }
}
    
```



להלן קטע תכנית עם הגדרות נכונות ושגויות:

```
public 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[2] = new InterA();
```

ניתן ליצור הפניה 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)
{
    System.out.print(arr[i].getClass() + ": arr[" + i + "] : ");
    arr[i].print();
}
```

לאחר הסרת שורות ההשמה השגויות נותר חלק מותאי המערך ריקים, כלומר null.

```
class A: arr[0] : print() from A: 5
class B: arr[1] : print() from B: 8
class 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] instanceof InterB)
{
    System.out.print(arr[i].getClass() + ": arr[" + i + "] : ");
    ((InterB)arr[i]).fb();
}
```

תא המכיל null ממילא אינו מסוג הממשק

```
class B: arr[1] : fb() From B: 12
class C: arr[2] : fb() From C: Xo-Xo-Xo
```