A c'tor a → A .1

A Print Me

B c'tor

B Print Me

C c1 → classC [x=9, y=0, z=5] .2

"ClassC::x=9 y=0 z=5"

A arr[0] → classA [x=12]

A arr[1] → classB [x=19 y=3]

A arr[2] → [x=4 y=11 z=2]

A arr[3] → C [x=1 y=8 z=5]

A arr[3] → C [x=4 y=11 z=2]

count=0

"ClassA::x=12"

count=1

"ClassB::x=19 y=3"

count=2

"ClassC::x=4 y=11 z=2"

"ClassC::x=4 y=11 z=2"

"1# count is 2"

count=0

"Print ClassA!"

"Print ClassA!"

count=1

"Print ClassC!"

count=2

"Print ClassC!"

count=3

"2# count is 3"

CamScanner נסרק עם

3. אמנם כניסה - אובייקט שלפי מסוג One
אמנם יציאה - האובייקט עם תכונת num הקטנה יותר

4. אמנם כניסה - אובייקט מסוג One
אמנם יציאה - אם זה הוא מסוג Two, אז תחליף את ה - num
רק מבחינתם, אם לא אז תחליף את הדינמיקה הנוכחית.

3.

One arr[0] → Two: num=1, min=0

One arr[1] → One: num=5, min=0

One arr[2] → Two: num=1, min=0

One arr[3] → Two: num=1, min=0

One arr[4] → Two: num=1, min=0

4. קיים הבדל בין השורות:
ב-#1# לא כתוב לו GetSmaller של a אבל יחזיר this.
ב-#2# כתוב לו GetSmaller והתבצע השוואה בין a לאובייקטים.

5.
arr[0] - up casting
arr[1] - אין הערה
arr[2] - up casting
arr[3] - up casting
arr[4] - up casting

6. לא חוקית, מאחר הוא תכונה מסוג private.
אפשר לגשת בפועל SetNum פרשנב את ערך num בתוך המחלקה.

```
internal class B:A
{
  protected int c;

  public B(int a, int b, int c): base(a,b)
  {
    this.c=c;
  }

  public void SetC(int c)
  {
    this.c=c;
  }

  public int GetC()
  {
    return this.c;
  }

  public override void CalcMe()
  {
    sumIt=this.a+this.b+this.c;
  }

  public override void PrintMe()
  {
    Base.PrintMe();
    Console.WriteLine("PrintMe class B ({0},{1},{2})",this.a,this.b,this.c);
  }

}
```

A a1 → A: a=1 b=2 sumIt=3

B b1 → B: a=10 b=20 c=30 sumIt=30

A myArr[0] → a1

A myArr[1] → b1

A myArr[2] → B: a=-10 b=20 c=-30 sumIt=-20

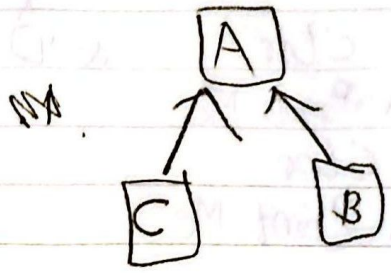
A myArr[3] → B: a=2 b=4 c=8 sumIt=6

"PrintMe class A (1,2)"
"myArr #0 Sum is: 3"
myArr[1] → a=10 b=20 c=30 sumIt=60
"PrintMe class A (10,20)"
"PrintMe class B (10,20,30)"
"myArr #1 Sum is: 60"
myArr[2] → a=-10 b=20 c=-30 sumIt=-20
"PrintMe class A(-10,20)"
"PrintMe class B(-10,20,-30)"
"myArr #2 Sum is: -20"

myArr[3] → a=2 b=4 c=8 sumIt=14
"PrintMe class A(2,4)"
"PrintMe class B(2,4,8)"
"myArr #3 Sum is: 14"

CamScanner נסרק עם

10.



1. Object, אין קשר בין D לשאר המחלקות האחרות
ולכן נשתמש בטיפוס היחיד שיכול לכיל עצמים
לא קשורים.

ב.

```
public static int SumArr(Object[] arr)
{
    int sum=0;
    for (int i=0; i<arr.length; i++)
    {
        if (arr[i] is A || arr[i] is D)
            sum += arr[i].F();
    }
    return sum;
}
```

נסרק עם CamScanner