

תכנות מונחה עצמים - ממשקים

לפני כן- חזרה קצרה. לפניכם שתי מחלקות :

הראשונה מחלקת בסיס

```
class ClassA
{
    protected int a;
    public ClassA(int a)
    {
        this.a = a;
    }
    public override string ToString()
    {
        Console.WriteLine("i am class A");

        return "my value is," + this.a;
    }
}
```

המחלקה השנייה יורשת ממחלקת הבסיס

```
class ClassB: ClassA
{
    int b;
    public ClassB(int b,int a): base(a)
    {
        this.b = b;
    }
    public override string ToString()
    {
        Console.WriteLine("i am not from class a "+base.ToString());
        return "my value is," + this.b;
    }
}
```

בתכנית הראשית

```
static void Main(string[] args)
{
    ClassA a = new ClassA(3);
    Console.WriteLine(a.ToString());
    Console.WriteLine("-----");

    ClassB b = new ClassB(4, 5);
    Console.WriteLine(b.ToString());
}
```

משימה ראשונה

1. הוסיפו מחלקה נוספת `ClassC` היורשת מ `ClassB`
2. צרו תרשים הורשה של כל המחלקות
3. צרו בתכנית הראשית מופע לכל אחת מהמחלקות בנפרד. הציגו את הפלט המתקבל
4. תארו את סדר הבניה של המחלקות עבור המופע של המחלקה החדשה שהוספתם.

משימה שניה

נחזור לתכנית שלנו מהתרגיל של גן החיות

זוכרים את גן החיות שלנו?

הנה הקוד שאיתו נעבוד- העתיקו אותו לסביבת העבודה כי איתו נמשיך הלאה בחקר הפלט שלנו

```
class Animal
{
    protected string name;
    protected char gender;
    public Animal(string name, char gender)
    {
        this.name = name;
        this.gender = gender;
    }
    public override string ToString()
    {
        return "i am an animal";
    }
}
class Dog : Animal
{
    private int speed;
    public Dog(int speed, string name, char gender) : base(name,
gender)
    {
        this.speed = speed;
    }
    public int GetSpeed()
    {
        return this.speed;
    }
    public string Getname()
    {
        return this.name;
    }

    public override string ToString()
    {
        base.ToString();
        return "i am a dog "+this.Getname();
    }
}
class Bird:Animal
{
    double hight;
    public Bird(double hight, string name, char gender) : base(name,
gender)
    {
```

```

        this.hight = hight;
    }
    public override string ToString()
    {
        base.ToString();
        return "i fligh hight "+this.Getname();
    }
    public string Getname()
    {
        return this.name;
    }
}
class fish:Animal
{
    string watertype;
    public fish(string watertype, string name, char gender) :
base(name, gender)
    {
        this.watertype=watertype;
    }
    public override string ToString()
    {
        return " i am a fish "+this.Getname();
    }
    public string Getname()
    {
        return this.name;
    }
}

class Program
{
    static void show(Animal[] arr)
    {
        foreach(Animal item in arr)
            Console.WriteLine(item.ToString());
    }
    static void Main(string[] args)
    {
        Animal[] arr = new Animal[4];

        arr[0] = new fish("salt", "karish", 'm');
        arr[1] = new Bird(3.5, "pashosh", 'f');
        arr[2] = new Dog(30, "suzy", 'f');
        arr[3] = new fish("salt", "karish", 'm');
        show(arr);

        Console.ReadLine();
    }
}

```

הוסיפו בתכנית הראשית עוד איבר למערך- איבר של המחלקה `Animal`. שנו את גודל המערך בהתאם ובחנו את הפלט. עד כה הכל תקין.

אבל....

האם זה נכון לעשות זאת? האם בעולם האמיתי יש לנו אובייקט שנקרא "חיה"? ברור שלא!!

למעשה מחלקת הבסיס- אמורה להיות רק מחלקה שמשמשת להורשה- כלומר- מחלקות נגזרות מקבלות ממנה את התכונות המשותפות לכל המחלקות היורשות – וכל מחלקה שיורשת מרחיבה את מחלקת הבסיס. מחלקת הבסיס נועדה רק להרחבת יכולות על ידי מחלקות שיורשות ממנה.

כדי לאפשר זאת- נגדיר את מחלקת הבסיס כמחלקה אבסטרקטית בכותרת של המחלקה

`abstract class Animal`

חלק שני- מחלקות אבסטרקטיות

מה קרה עכשיו בתכנית הראשית?

אין לנו אפשרות ליצור מופע של מחלקה אבסטרקטית. המחלקה האבסטרקטית מיועדת רק להורשה.

ישנם מספר כללים במחלקות אבסטרקטיות

- בכדי ליצור מחלקה אבסטרקטית יש לכתוב `abstract` לפני שם המחלקה.
- לא ניתן ליצור מופע של אובייקטים בתכנית הראשית, ממחלקה אבסטרקטית.
- ניתן ליצור **ייחוס** ממחלקה אבסטרקטית (בד"כ לצורכי פולימורפיזם) למשל במערך של עצמים.
- מחלקה אבסטרקטית יכולה להכיל כל דבר שמחלקה רגילה יכולה להכיל. גם פעולה בונה. למעשה בלי פעולה בונה במחלקת הבסיס- לא נוכל לבצע הרחבת יכולות של המחלקות הנגזרות.
- מחלקה אבסטרקטית יכולה (אבל לא חייבת) להכיל פונקציות אבסטרקטיות. **לא ניתן לכתוב פונקציה אבסטרקטית במחלקה שהיא לא אבסטרקטית.**
- פונקציה אבסטרקטית היא פונקציה שאין לה מימוש. היא נכתבת כהצהרה ללא גוף כלומר ללא בלוק של פקודות ועם הסימן ; בסוף ההצהרה.
- פונקציה אבסטרקטית היא תמיד פונקציה וירטואלית- אין צורך להגדיר אותה במילה `virtual`
- כאשר יורשים מחלקה אבסטרקטית חובה לממש את כל הפונקציות האבסטרקטיות שבה (אם יש). המימוש של פונקציה אבסטרקטית הוא באמצעות מילת המפתח `override`

נחקור את הכללים

עבור הקוד שהעתקתם במשימה שניה

1. הגדירו את המחלקה **חיה** כמחלקה אבסטרקטית.
2. מהי הבעיה בתכנית הראשית?- התאימו את הקוד בתכנית הראשית באופן שיפתור את הבעיה.
3. הריצו את הקוד ובחנו את הפלט
4. סמנו את הפעולה הבונה במחלקה חיה – כהערה בעזרת הסימן //. מהן השגיאות שהתקבלו במחלקות הנגזרות? סמנו גם אותן כהערה.

5. הריצו ובחנו את הפלט. השוו אותו מול הפלט שקיבלתם בסעיף 3.
6. הוסיפו במחלקה חיה פעולה וירטואלית בשם Food. הפעולה תחזיר מחרוזת "I am hungry"
7. הוסיפו בכל המחלקות היורשות מהמחלקה האבסטרקטית פעולה הדורסת את הפעולה הוירטואלית שבסעיף 6. העולה תזמן את הפעולה ממחלקת הבסיס באופן הבא

```
return this.name+base.Food();
```

- בחנו את הפלט.
8. הגדירו את הפעולה הוירטואלית שהגדרתם בסעיף 6 כפעולה אבסטרקטית. מהי השגיאה שהתקבלה בפעולה זו- שימו לב לשגיאות הנוספות שבמחלקות הנגזרות שמפעילות את הפעולה הזו. כיצד נתקן את השגיאה?
9. גן החיות החליט שלכל חיה יוגדר עובד שתפקידו לדאוג לתזונת החיות. תפקידו לדאוג שכל חיה תקבל את המזון הייחודי לה. ממשו את הפעולה `public abstract string Feeder();` בהתאם לכללים שצויינו לעיל- בהתאם להבנתכם

השוואה בין virtual לabstract - מתוך אתר וובמאסטר

virtual	abstract	
יש לממש את הפונקציה (בדומה לפונקציה רגילה). משמש לפולימורפיזם.	אין מימוש של הפונקציה - לא צריך להגדיר את גוף הפונקציה, רק כותרת.	מימוש הפונקציה במחלקה
לא חובה לממש.	חובה לממש במחלקות היורשות בצורה ישירה.	מימוש הפונקציה במחלקות היורשות
בכל מחלקה	במחלקה אבסטרקטית בלבד	מיקום הפונקציה
	לא private (משום שאז הפונקציה לא תיראה במחלקות הנגזרות ולא נוכל לממש אותה)	הרשאת גישה

הקוד המלא מופיע כאן

```
abstract class Animal
{
    protected string name;
    protected char gender;
    public Animal(string name, char gender)
    {
        this.name = name;
        this.gender = gender;
    }
    public override string ToString()
    {
        return "i am an animal";
    }

    public abstract string Food();
}
```

```

class Dog : Animal
{
    private int speed;
    public Dog(int speed, string name, char gender) : base(name,
gender)
    {
        this.speed = speed;
    }
    public int GetSpeed()
    {
        return this.speed;
    }
    public string Getname()
    {
        return this.name;
    }
    public override string Food()
    {
        return this.name;
    }

    public override string ToString()
    {
        base.ToString();
        return " i am a dog "+this.Getname();
    }
}
class Bird:Animal
{
    private double hight;
    public Bird(double hight, string name, char gender) : base(name,
gender)
    {
        this.hight = hight;
    }
    public override string ToString()
    {
        base.ToString();
        return " i fligh hight "+this.Getname();
    }
    public string Getname()
    {
        return this.name;
    }
    public override string Food()
    {
        return this.name;
    }
}
class Fish:Animal

```

```

    {
        private string watertype;
        public Fish(string watertype, string name, char gender) :
base(name, gender)
        {
            this.watertype=watertype;
        }
        public override string ToString()
        {
            return " i am a fish "+this.GetName();
        }
        public string GetName()
        {
            return this.name;
        }
        public override string Food()
        {
            return this.name;
        }
    }
}

```

```

class Program

```

```

{
    static void Show(Animal[] arr)
    {
        foreach (Animal item in arr)
        {
            Console.WriteLine(item.ToString());
            Console.WriteLine(item.Food());
        }
    }
    static void Main(string[] args)
    {
        Animal[] arr = new Animal[4];

        arr[0] = new Fish("sweet", "bob", 'm');
        arr[1] = new Bird(3.5, "pashosh", 'f');
        arr[2] = new Dog(30, "suzy", 'f');
        arr[3] = new Fish("salt", "karish", 'm');

        Show(arr);

        Console.ReadLine();
    }
}

```