

Άσκηση 1: Δημιουργία Κλάσης με Abstraction

Οδηγίες:

1. Δημιούργησε μια κλάση `Animal` που θα έχει μια μέθοδο `MakeSound()` χωρίς υλοποίηση.
2. Δημιούργησε δύο υποκλάσεις της κλάσης `Animal`: `Dog` και `Cat`.
3. Κάθε υποκλάση θα πρέπει να υλοποιεί τη μέθοδο `MakeSound()` με διαφορετικό ήχο για το σκύλο και τη γάτα.

Λύση:

```
using System;

// Κλάση με Abstraction
abstract class Animal
{
    // Αφηρημένη μέθοδος χωρίς υλοποίηση
    public abstract void MakeSound();
}

// Υποκλάση που αντιπροσωπεύει Σκύλο
class Dog : Animal
{
    // Υλοποίηση της αφηρημένης μεθόδου
    public override void MakeSound()
    {
        Console.WriteLine("Ο σκύλος λέει: Γαβ γαβ!");
    }
}

// Υποκλάση που αντιπροσωπεύει Γάτα
class Cat : Animal
{
    // Υλοποίηση της αφηρημένης μεθόδου
    public override void MakeSound()
    {
        Console.WriteLine("Η γάτα λέει: Μιάου μιάου!");
    }
}

// Κύρια κλάση για εκτέλεση
class Program
{
    static void Main()
    {
        Animal myDog = new Dog();
    }
}
```

```

        Animal myCat = new Cat();

        myDog.MakeSound(); // Εκτυπώνει: Ο σκύλος λέει: Γαβ γαβ!
        myCat.MakeSound(); // Εκτυπώνει: Η γάτα λέει: Μιάου
        miaou!
    }
}

```

Άσκηση 2: Δημιουργία Διεπαφής με Abstraction

Οδηγίες:

1. Δημιούργησε μια διεπαφή `IDriveable` με μια μέθοδο `Drive()`.
2. Δημιούργησε δύο κλάσεις: `Car` και `Bicycle`, που υλοποιούν τη διεπαφή `IDriveable`.
3. Κάθε κλάση θα πρέπει να έχει διαφορετική υλοποίηση της μεθόδου `Drive()`.

Λύση:

```

using System;

// Διεπαφή που καθορίζει τι πρέπει να κάνει κάτι για να είναι
// οδηγούμενο
interface IDriveable
{
    void Drive();
}

// Κλάση που αντιπροσωπεύει Αυτοκίνητο
class Car : IDriveable
{
    // Υλοποίηση της μεθόδου Drive
    public void Drive()
    {
        Console.WriteLine("Το αυτοκίνητο οδηγείται.");
    }
}

// Κλάση που αντιπροσωπεύει Ποδήλατο
class Bicycle : IDriveable
{
    // Υλοποίηση της μεθόδου Drive
    public void Drive()
    {
        Console.WriteLine("Το ποδήλατο ποδηλατεί.");
    }
}

```

```

    }
}

// Κύρια κλάση για εκτέλεση
class Program
{
    static void Main()
    {
        IDriveable myCar = new Car();
        IDriveable myBicycle = new Bicycle();

        myCar.Drive();          // Εκτυπώνει: Το αυτοκίνητο οδηγείται.
        myBicycle.Drive();      // Εκτυπώνει: Το ποδήλατο ποδηλατεί.
    }
}

```

Άσκηση 3: Σύνθετο Παράδειγμα με Abstraction

Οδηγίες:

1. Δημιούργησε μια αφηρημένη κλάση Shape με μια αφηρημένη μέθοδο Draw().
2. Δημιούργησε δύο υποκλάσεις της κλάσης Shape: Circle και Rectangle.
3. Κάθε υποκλάση θα πρέπει να υλοποιεί τη μέθοδο Draw() με διαφορετική αναπαράσταση.

Λύση:

```

using System;

// Αφηρημένη κλάση Shape
abstract class Shape
{
    // Αφηρημένη μέθοδος χωρίς υλοποίηση
    public abstract void Draw();
}

// Υποκλάση που αντιπροσωπεύει Κύκλος
class Circle : Shape
{
    // Υλοποίηση της αφηρημένης μεθόδου
    public override void Draw()
    {
        Console.WriteLine("Σχεδιάζω έναν κύκλο.");
    }
}

```

```
// Υποκλάση που αντιπροσωπεύει Ορθογώνιο
class Rectangle : Shape
{
    // Υλοποίηση της αφηρημένης μεθόδου
    public override void Draw()
    {
        Console.WriteLine("Σχεδιάζω ένα ορθογώνιο.");
    }
}

// Κύρια κλάση για εκτέλεση
class Program
{
    static void Main()
    {
        Shape myCircle = new Circle();
        Shape myRectangle = new Rectangle();

        myCircle.Draw();          // Εκτυπώνει: Σχεδιάζω έναν κύκλο.
        myRectangle.Draw();       // Εκτυπώνει: Σχεδιάζω ένα
ορθογώνιο.
    }
}
```

Άσκηση 4: Συνδυασμός Κλάσεων και Διεπαφών

Οδηγίες:

1. Δημιούργησε μια διεπαφή `IPlayable` με μια μέθοδο `Play()`.
2. Δημιούργησε μια αφηρημένη κλάση `Media` που υλοποιεί τη διεπαφή `IPlayable`.
3. Δημιούργησε δύο υποκλάσεις της κλάσης `Media`: `Music` και `Video`, που έχουν δικές τους υλοποιήσεις για τη μέθοδο `Play()`.

```
using System;

// Διεπαφή που καθορίζει την ικανότητα αναπαραγωγής
interface IPlayable
{
    void Play();
}

// Αφηρημένη κλάση Media
abstract class Media : IPlayable
{
}
```

```
// Η αφηρημένη κλάση υλοποιεί τη διεπαφή αλλά δεν παρέχει
υλοποίηση για τη μέθοδο Play
}

// Υποκλάση που αντιπροσωπεύει Μουσική
class Music : Media
{
    // Υλοποίηση της μεθόδου Play
    public override void Play()
    {
        Console.WriteLine("Παίζει μουσική.");
    }
}

// Υποκλάση που αντιπροσωπεύει Βίντεο
class Video : Media
{
    // Υλοποίηση της μεθόδου Play
    public override void Play()
    {
        Console.WriteLine("Παίζει βίντεο.");
    }
}

// Κύρια κλάση για εκτέλεση
class Program
{
    static void Main()
    {
        IPlayable myMusic = new Music();
        IPlayable myVideo = new Video();

        myMusic.Play(); // Εκτυπώνει: Παίζει μουσική.
        myVideo.Play(); // Εκτυπώνει: Παίζει βίντεο.
    }
}
```