

## BÀI 6. ĐA HÌNH (1/2)

### Bài 1. Biết và hiểu các bước xây dựng phương thức có tính đa hình

Có 3 loại xe hơi (**Car**): **LandRover**, **Ford**, **Honda** cần mô hình hóa trên máy tính:



Mỗi loại xe trên tương ứng với một lớp: Lớp **LandRover**, lớp **Ford**, lớp **Honda**

Mỗi loại xe khi di chuyển (**Move**) sẽ dùng những động cơ khác nhau:

- Xe **LandRover** : Dùng động cơ **2.0 L engine**
- Xe **Ford** : Dùng động cơ **V engine**
- Xe **Honda** : Dùng động cơ **i-vtec engine**

Như vậy, mỗi lớp có phương thức **Move()** riêng để xuất ra màn hình thông tin động cơ của xe đang dùng.

Hãy thiết kế các lớp và viết phương thức **Move()** của các lớp sao cho:

- Lúc đầu người dùng tạo ra **một danh sách** chứa các đối tượng trên (danh sách này vừa chứa xe **LandRover**, vừa chứa xe **Ford** và vừa chứa xe **Honda**, mỗi loại xe có thể được tạo nhiều lần) .
- Với danh sách đối tượng đã tạo, khi ra lệnh **Move** (Gọi phương thức **Move()** cho từng đối tượng trong danh sách, các đối tượng sẽ dùng các động cơ tương ứng để di chuyển (yêu cầu, chỉ cần xuất ra màn hình xe di chuyển bằng động cơ nào trong phương thức **Move()**). Người ta nói, phương thức **Move()** như vậy **có tính đa hình**.

## Bài giải Thực hành Lập trình Hướng đối tượng – Buổi 6

Cụ thể, hãy thiết kế các lớp và viết các phương thức **Move()** sao cho phương thức **Main()** hoạt động và cho kết quả như sau:

```
static void Main()
```

```
{
```

```
    Car[] cars = new Car[3];
```

```
    cars[0] = new LandRover();
```

```
    cars[1] = new Ford();
```

```
    cars[2] = new Honda();
```

```
    foreach (Car car in cars)
```

```
        car.Move();
```

```
}
```

**Phương thức Move() chạy sẽ cho kết quả:**

Move: **LandRover (2.0 L engine)**

Move: **Ford (V engine)**

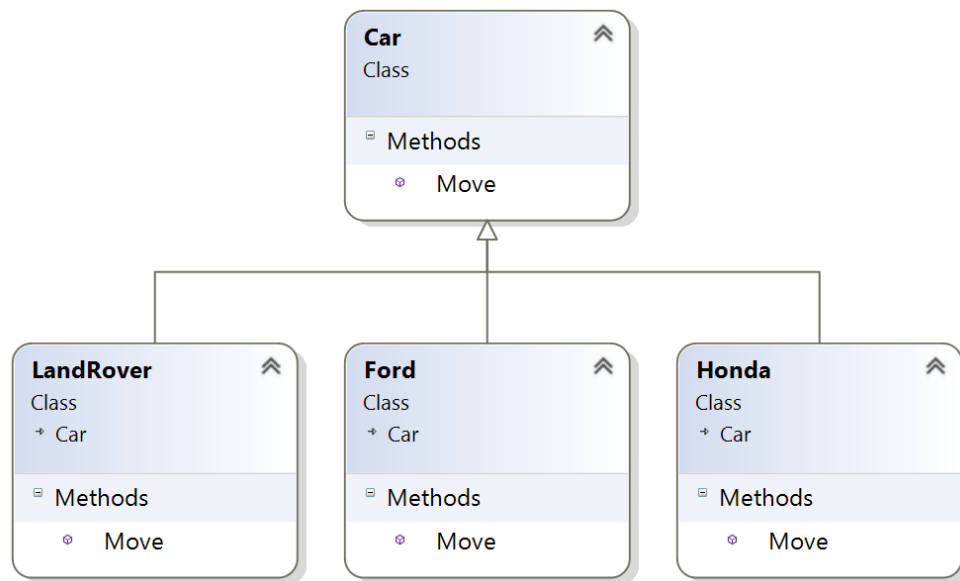
Move: **Honda (i-vtex engine)**

car.Move() gọi Move() của lớp  
LandRover

car.Move() gọi Move() của lớp  
Ford

car.Move() gọi Move() của lớp  
Honda

**Sơ đồ kế thừa:**



- Cài đặt sơ đồ thừa kế trên
- Cài đặt phương thức **Move()** sao cho phương thức này **có tính đa hình**
- Trong trong phương thức **Main()**, viết lệnh kiểm tra tính đa hình của phương thức **Move()** như đã mô tả

**Chú ý:** Phương thức **Main()** có thể viết khác đi (Dùng collection có tên **List**)

```
static void Main()
{
    List<Car> cars = new List<Car>();
    cars.Add(new LandRover());
    cars.Add(new Ford());
    cars.Add(new Honda());

    foreach (Car car in cars)
        car.Move();
}
```

**Kết quả khi thực hiện các lệnh trên vẫn là:**

Move: **LandRover (2.0 L engine)**

Move: **Ford (V engine)**

Move: **Honda (i-vtex engine)**

**File Car.cs**

```
namespace ThuchanhOOP
{
    class Car
    {
        public virtual string Move()
        {
            return "Moving by using an engine type.";
        }
    }
}
```

**File Ford.cs**

```
namespace ThuchanhOOP
{
    class Ford : Car
    {
        public override string Move()
        {
            return " Move: Ford (V Engine)";
        }
    }
}
```

**File Honda.cs**

```
namespace ThuchanhOOP
{
    class Honda:Car
    {
        public override string Move()
        {
            return " Move: Honda (i-vtec Engine)";
        }
    }
}
```

**File LandRover.cs**

```
namespace ThuchanhOOP
{
    class LandRover : Car
    {
        public override string Move()
        {
            return " Move: LandRover (2.0 L Engine)";
        }
    }
}
```

### File Program.cs

```
using System;
using System.Collections.Generic;

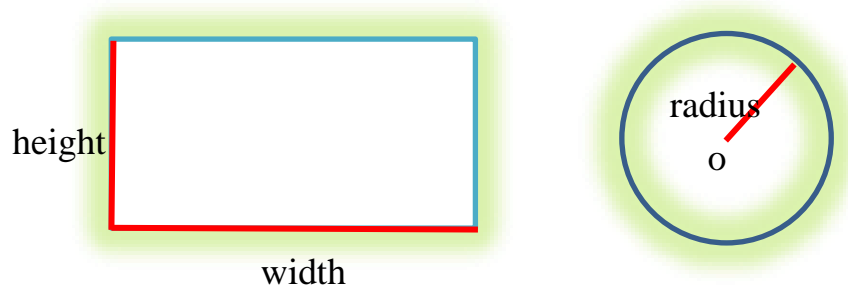
namespace ThuchanhOOP
{
    class Program
    {
        static void Main(string[] args)
        {
            List<Car> cars = new List<Car>();

            cars.Add(new LandRover());
            cars.Add(new Ford());
            cars.Add(new Honda());

            foreach (Car car in cars)
                Console.WriteLine(car.Move());

            Console.ReadKey();
        }
    }
}
```

**Bài 2. Tính đa hình của phương thức tính diện tích Area() của các hình**  
Có 2 loại hình khác nhau cần quản lý: hình chữ nhật (**Rectangle**), hình tròn (**Circle**).



- Hình chữ nhật: có chiều cao (height) và chiều rộng (width)
- Hình tròn: có bán kính radius

Diện tích các hình tính như sau:

$$S_{\text{chữ nhật}} = \text{height} \times \text{width}$$

$$S_{\text{hình tròn}} = 3.14 \times \text{radius}^2$$

### Yêu cầu:

- Hãy thiết kế các lớp (vẽ sơ đồ lớp) cho bài toán sao cho phương thức tính diện tích **Area()** của từng hình có tính đa hình (chú ý: lớp cơ sở không cần phải có fields)
- Hãy cài đặt sơ đồ lớp trên sao cho phương thức tính diện tích **Area()** của từng hình có tính đa hình
- Tạo một danh sách chứa một số đối tượng **Rectangle**, **Circle** (dùng phương thức constructor để tạo đối tượng)
- Tính diện tích của từng hình (xuất diện tích ra màn hình)
- Tính tổng diện tích các hình

### File Shape.cs

```
namespace ThuchanhOOP
{
    class Shape
    {
        public virtual double Area()
        {
            return 0.0;
        }
    }
}
```

### File Circle.cs

```
using System;

namespace ThuchanhOOP
{
    class Circle : Shape
    {
        private double radius;

        public double Radius
        {
            get => radius;
            set
            {
                if (value > 0)
                    radius = value;
            }
        }

        //constructors
        public Circle()
        {
            Radius = 1.0;
        }
    }
}
```

```
public Circle(double _radius)
{
    Radius = _radius;
}

//methods
public override double Area()
{
    return Math.PI * Radius * Radius;
}

public override string ToString()
{
    return $"Hình tron: Ban kinh = {Radius}";
}
}
```

### File Rectangle.cs

```
namespace ThuchanhOOP
{
    class Rectangle : Shape
    {
        double width, height;
        public double Width { get => width; set => width = value; }
        public double Height { get => height; set => height = value; }

        public Rectangle()
        {
            Width = 1.0;
            Height = 1.0;
        }
        public Rectangle(double h, double w)
        {
            Width = h;
            Height = w;
        }

        public override double Area()
        {
            return Height * Width;
        }

        public override string ToString()
        {
            return $"Hình chu nhat: Cao {Height} x Rong {Width}";
        }
    }
}
```

### File Program.cs

```
using System;
using System.Collections.Generic;

namespace ThuchanhOOP
{
    class Program
    {
        static void Main(string[] args)
        {
        }
    }
}
```

## Bài giải Thực hành Lập trình Hướng đối tượng – Buổi 6

```
List<Shape> shapes = new List<Shape>();

Circle c = new(3.6);
shapes.Add(c);

Rectangle r = new(2.3, 4.0);
shapes.Add(r);

c = new(1.7);
shapes.Add(c);

c = new(4.15);
shapes.Add(c);

r = new();
shapes.Add(r);

double TongDT = 0;
foreach (var a in shapes)
{
    Console.WriteLine(a.ToString());
    Console.WriteLine($"Dien tich = {a.Area():0.00}");
    TongDT += a.Area();
}
Console.WriteLine($"Tong dien tich = {TongDT:0.00}");

Console.ReadKey();
}
```