

## BÀI 2. LỚP VÀ ĐỐI TƯỢNG (2/3)

### Bài 1. Dùng property với lớp Account

Trong lớp **Account** thay thế các phương thức **Get/Set** thành các **properties**. Và kiểm tra tính hợp lệ trước khi thay đổi dữ liệu của field

- *Số tiền* phải là số dương

#### File Account.cs

```
using System;

namespace ThuchanhOOP
{
    class Account
    {
        //fields - data members
        private string soTK;
        private long soTien;

        //Properties
        public string SoTK { get => soTK; }

        //constructors
        public Account(string soTK)
        {
            this.soTK = soTK;
            soTien = 0;
        }
        public Account(string soTK, long soTien) : this(soTK)
        {
            napTien(soTien);
        }

        //methods
        public void rutTien(long stien)
        {
            if (stien > 0 && stien <= soTien)
                soTien = soTien - stien;
        }

        public void napTien(long stien)
        {
            if (stien > 0)
                soTien = soTien + stien;
        }

        //Hiển thị thông tin tài khoản, gồm mã số tài khoản và số tiền hiện có trong tài khoản
        public long KiemTraSoDu()
        {
            return soTien;
        }
    }
}
```

## Bài 2. Dùng properties với lớp Book

Trong lớp **Book** thay thế các phương thức **Get/Set** thành **properties**. Và kiểm tra tính hợp lệ trước khi thay đổi dữ liệu của các fields

- *Tên sách* không được là chuỗi rỗng
- *Giá sách* phải là số dương
- *Giảm giá* không được là số âm

### File Book.cs

```
using System;

namespace ThuchanhOOP
{
    class Book
    {
        //fields - data members
        private string maSach;
        private string tenSach;
        private long giaSach;
        private long giamGia;

        //properties
        public string MaSach { get => maSach; }
        public string TenSach
        {
            get => tenSach;
            set
            {
                if (string.IsNullOrEmpty(value) == false)
                    tenSach = value;
            }
        }
        public long GiaSach
        {
            get => giaSach;
            set
            {
                if (value > 0)
                    giaSach = value;
            }
        }
        public long GiamGia
        {
            get => giamGia;
            set
            {
                if (value >= 0 && value < GiaSach)
                    giamGia = value;
            }
        }

        //constructors
        public Book(string maSach, string tenSach)
        {
            this.maSach = maSach;
            this.tenSach = string.IsNullOrEmpty(tenSach) ? "Noname" : tenSach;
            GiaSach = 0;
        }
    }
}
```

## Bài giải Thực hành Lập trình Hướng đối tượng – Buổi 2

```
GiamGia = 0;
}
public Book(string maSach, string tenSach, long giaSach, long giamGia) :
    this(maSach, tenSach)
{
    GiaSach = giaSach;
    GiamGia = giamGia;
}

//methods
public long GiaBan()
{
    return (GiaSach - GiamGia);
}
}
```

### Bài 3, 4

Thực hiện tương tự các Bài 1, 2

### Bài 5. Sử dụng mảng đối tượng của lớp Circle

Tạo project mới, sau đó thêm lớp **Circle** đã tạo vào project này.

Viết phương thức **Main()**

- Khai báo và nhập n hình tròn (n là số nguyên được nhập từ bàn phím)
- Xuất thông tin n hình tròn đã nhập ra màn hình
- Tính và xuất diện tích của từng hình tròn trên
- Tìm diện tích lớn nhất trong n hình tròn trên

Ví dụ:

Nhap so luong duong tron n: 3

Nhap duong tron thu 1:

Ban kinh r: 5

Nhap duong tron thu 2:

Ban kinh r: 8

Nhap duong tron thu 3:

Ban kinh r: 15

3 duong tron da nhap:

Ban kinh r: 5

Ban kinh r: 8

Ban kinh r: 15

## Bài giải Thực hành Lập trình Hướng đối tượng – Buổi 2

Diện tích của 3 đường tròn:

Diện tích đường tròn 1: ...

Diện tích đường tròn 2: ...

Diện tích đường tròn 3: ...

Diện tích đường tròn lớn nhất: ...

### File Circle.cs:

```
using System;

namespace ThuchanhOOP
{
    class Circle
    {
        private int banKinh;

        public int BanKinh
        {
            get => banKinh;
            set
            {
                if (value > 0)
                    banKinh = value;
            }
        }

        //constructors
        public Circle()
        {
            BanKinh = 1;
        }
        public Circle(int bankinh)
        {
            BanKinh = bankinh;
        }

        //methods
        public void Input()
        {
            Console.Write("Nhập vào bán kính r: ");
            BanKinh = (int.Parse(Console.ReadLine()));
        }
        public void Output()
        {
            Console.WriteLine($"Bán kính r: {BanKinh}");
        }

        public double Dientich()
        {
            return Math.PI * BanKinh * BanKinh;
        }
        public double Chuvi()
        {
            return 2 * Math.PI * BanKinh;
        }
    }
}
```

### File Program.cs

```
using System;

namespace ThuchanhOOP
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.Write("Nhap so luong hình tròn: ");
            int n = int.Parse(Console.ReadLine());

            Circle[] circles = new Circle[n];

            double maxDT = 0;
            for (int i = 0; i < circles.Length; i++)
            {
                circles[i] = new Circle();
                Console.WriteLine($"Nhap thông tin hình tròn thu {i + 1}: ");
                circles[i].Input();
                if (maxDT < circles[i].Dientich())
                    maxDT = circles[i].Dientich();
            }

            Console.WriteLine();

            Console.WriteLine($"{circles.Length} hình tròn đã nhập:");
            for (int i = 0; i < circles.Length; i++)
            {
                circles[i].Output();
                Console.WriteLine($"Diện tích = {circles[i].Dientich()}");
            }
            // Hoặc
            //foreach (Circle dt in circles)
            //{
            //    dt.Output();
            //    Console.WriteLine($"Diện tích = {dt.Dientich()}");
            //}

            Console.WriteLine();

            Console.WriteLine($"Diện tích lớn nhất: {maxDT.ToString("0.00")}");

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

## Bài 6. Tạo lớp Color

Trên máy tính, màu (color) được biểu diễn bằng các giá trị *red*, *green*, *blue* (có giá trị từ 0 đến 255) và *alpha* (hay transparency, mức độ trong suốt của màu, cũng có giá trị từ 0 đến 100, với 100 là hoàn toàn trong suốt, 0 là đậm đặc). Hãy tạo lớp **Color**

- Thêm các **instance variables** (một cách gọi khác của **fields**)
- Thêm các **constructors**
  - **Constructor không tham số (default constructor)** để tạo đối tượng sao cho các fields: red, green, blue, có giá trị mặc định là 255 (màu trắng) và alpha = 0
  - **Constructor** có tham số để tạo đối tượng màu có 4 thông tin trên
  - **Constructor** có tham số là đối tượng color cho trước. Constructor này có nhiệm vụ tạo đối tượng màu và sao chép thông tin của đối tượng color trong tham số vào đối tượng mới tạo (constructor này còn được gọi là **Copy constructor**)

```
public Color() {}  
public Color(int red, int green, int blue, int alpha) {}  
public Color(Color color) {}
```

- Thêm các **properties** cho các fields với các ràng buộc của các field như đã mô tả
- Thêm phương thức tính giá trị mức xám (grayscale) của màu (grayscale bằng trung bình của red, green, blue)

### File Color.cs

```
using System;  
  
namespace ThuchanhOOP  
{  
    class Color  
    {  
        //Fields  
        private int red;  
        private int green;  
        private int blue;  
        private int alpha;  
  
        //Properties  
        public int Red  
        {  
            get => red;  
            set  
            {  
                if (value >= 0 && value <= 255)  
                    red = value;  
            }  
        }  
    }  
}
```

```
public int Green
{
    get => green;
    set
    {
        if (value >= 0 && value <= 255)
            green = value;
    }
}
public int Blue
{
    get => blue;
    set
    {
        if (value >= 0 && value <= 255)
            blue = value;
    }
}
public int Alpha
{
    get => alpha;
    set
    {
        if (value >= 0 && value <= 100)
            alpha = value;
    }
}

//Constructors
public Color()
{
    Red = 255;
    Green = 255;
    Blue = 255;
    Alpha = 0;
}
public Color(int r, int g, int b, int a)
{
    Red = r;
    Green = g;
    Blue = b;
    Alpha = a;
}
//Copy constructor
public Color(Color a)
{
    Red = a.Red;
    Green = a.Green;
    Blue = a.Blue;
    Alpha = a.Alpha;
}

//Methods
public int getGrayscale()
{
    return (Red + Green + Blue) / 3;
}
```

## Bài giải Thực hành Lập trình Hướng đối tượng – Buổi 2

```
public void Input()
{
    Console.Write("Gia tri Red: ");
    Red = int.Parse(Console.ReadLine());
    Console.Write("Gia tri Green: ");
    Green = int.Parse(Console.ReadLine());
    Console.Write("Gia tri Blue: ");
    Blue = int.Parse(Console.ReadLine());
    Console.Write("Gia tri Alpha: ");
    Alpha = int.Parse(Console.ReadLine());
}

public void Output()
{
    Console.WriteLine($"Red = {Red}");
    Console.WriteLine($"Green = {Green}");
    Console.WriteLine($"Blue = {Blue}");
    Console.WriteLine($"Alpha = {Alpha}");
}
}
```

### Bài 7 Xây dựng lớp **Ball**, sử dụng lớp **Color** làm field

Trái banh có các fields: kích thước (*size*), màu sắc (*color*) và số lần ném banh (*timeOfThrown*)

- Khai báo các **fields** cho **Ball** (field **color** nên dùng kiểu **Color** đã tạo trong bài trước)
- Thêm các **constructors**
  - **Constructor** không tham số để tạo đối tượng Ball có những thông tin mặc định: kích thước bằng 1, số lần ném banh là 0, màu là màu mặc định
  - **Constructor** có tham số để tạo đối tượng có 3 thông tin trên
- Thêm các **properties** cho các **fields** (nhớ ràng buộc dữ liệu hợp lý)
- Thêm phương thức **Pop** (Pop nghĩa là mô hình việc làm cho trái banh nổ): Banh nổ nghĩa là thay đổi size thành giá trị 0
- Thêm phương thức **Throw** (**Throw** nghĩa là mô hình việc ném trái banh): Ném banh nghĩa là cộng 1 vào số lần ném (nhưng chỉ cộng khi banh chưa bị nổ)
- Viết phương thức **Input()** để nhập thông tin trái banh
- Viết phương thức **Output()** xuất thông tin trái banh



Viết phương thức **Main()** thực hiện:

- Khai báo và tạo n trái banh có giá trị mặc định (số n được nhập từ bàn phím)
- Nhập lại thông tin trái banh thứ 0
- Xuất thông tin n trái banh ra màn hình (kiểm tra xem dữ liệu có như mong đợi không)
- Ném trái banh thứ 1, 2
- Cho nổ trái banh thứ 1 và ném trái banh thứ 1
- In số lần ném của các trái banh

### File Ball.cs

```
using System;

namespace ThuchanhOOP
{
    class Ball
    {
        //Fields
        private int size;
        private Color colorOfBall;
        private int timeOfThrown;

        internal Color ColorOfBall { get => colorOfBall; set => colorOfBall = value; }
        public int TimeOfThrown { get => timeOfThrown; }

        //Constructors
        public Ball()
        {
            size = 1;
            timeOfThrown = 0;
            ColorOfBall = new Color();
        }

        public Ball(int size, Color ballColor, int timeOfThrown) : this()
        {
            if (size > 0)
                this.size = size;
            ColorOfBall = ballColor;
            if (timeOfThrown >= 0)
                this.timeOfThrown = timeOfThrown;
        }

        //Methods
        public void Pop()
        {
            size = 0;
        }

        public void Throw()
        {
            if (size > 0)
                timeOfThrown += 1;
        }
    }
}
```

## Bài giải Thực hành Lập trình Hướng đối tượng – Buổi 2

```
public void Input()
{
    Console.Write("Nhập vào kích co qua banh: ");
    int tmp = int.Parse(Console.ReadLine());
    if (tmp > 0)
        size = tmp;
    Console.WriteLine("Mau cua qua banh: ");
    ColorOfBall.Input();
    Console.Write("Nhập vào so lan da nem: ");
    tmp = int.Parse(Console.ReadLine());
    if (tmp >= 0)
        timeOfThrown = tmp;
}

public void Output()
{
    Console.WriteLine($"Kich co: {size}");
    Console.WriteLine("Mau sac:");
    ColorOfBall.Output();
    Console.WriteLine($"So lan da nem: {TimeOfThrown}");
}
}
```

### File Program.cs

```
using System;

namespace ThuchanhOOP
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.Write("Nhập vào số lượng qua banh: ");
            int n = int.Parse(Console.ReadLine());

            Ball[] balls = new Ball[n];

            for (int i = 0; i < balls.Length; i++)
                balls[i] = new Ball();

            Console.WriteLine("Nhập thông tin qua banh thu 1:");
            balls[0].Input();

            Console.WriteLine();

            for (int i = 0; i < balls.Length; i++)
            {
                Console.WriteLine($"Thông tin qua banh thu {i + 1}:");
                balls[i].Output();
            }

            Console.WriteLine();

            Console.WriteLine("Nem qua banh thu 1!");
            balls[0].Throw();
            Console.WriteLine("Nem qua banh thu 2!");
            balls[1].Throw();

            Console.WriteLine("Lam no qua banh thu 1!");
            balls[0].Pop();
        }
    }
}
```

## Bài giải Thực hành Lập trình Hướng đối tượng – Buổi 2

---

```
        Console.WriteLine("Nem qua banh thu 1!");
        balls[0].Throw();

        Console.WriteLine();

        Console.WriteLine("So lan da nem cua cac qua banh:");
        for (int i = 0; i < n; i++)
            Console.WriteLine($"Qua banh thu {i + 1}: {balls[i].TimeOfThrown}
                                lan");

        Console.ReadKey();
    }
}
```