

## BÀI 5. KẾ THỪA

### Bài 1. Lớp cơ sở

Một người (**Person**) cần mô hình trên máy gồm có các thông tin: tên (**name**) và **email**

- Khai báo các **Fields**
- Cài đặt các **Properties**
- Viết **default constructor** để tạo đối tượng có giá trị mặc định
- Viết **parametered constructor** để tạo đối tượng có 2 thông tin trên

#### File Person.cs

```
namespace ThuchanhOOP
{
    class Person
    {
        private string name;
        private string email;

        public string Name { get => name; set => name = value; }
        public string EMail { get => email; set => email = value; }

        public Person()
        {
            Name = "";
            EMail = "";
        }
        public Person(string _name, string _email)
        {
            Name = _name;
            EMail = _email;
        }

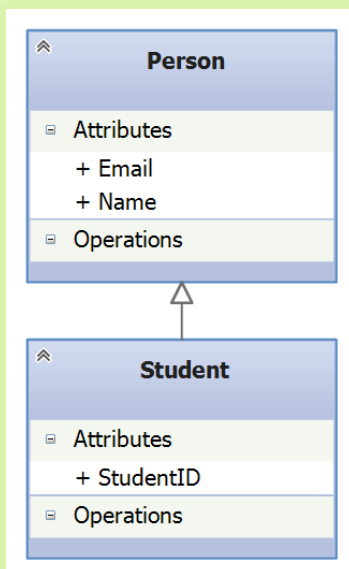
        public string Show()
        {
            string result = $"Ho ten: {Name} \nEmail: {EMail}";
            return result;
        }
    }
}
```

## Bài 2. Lớp dẫn xuất

Chúng ta cần quản lý thông tin của một sinh viên (**Student**) trên máy tính. **Sinh viên (Student) cũng là một người (Person)** thông thường, nghĩa là cũng có các thông tin như: tên (**name**), **email**. Tuy nhiên, **sinh viên còn có thêm thông tin khác** là mã sinh viên (**StudentID**)

Hãy xây dựng lớp **Student**

- Lớp **Student** **kế thừa** lớp **Person**
- Khai báo **Field**
- Cài đặt **Property**



- Viết **default constructor** để tạo đối tượng có giá trị mặc định
- Viết **parametered constructor** để tạo đối tượng có 3 thông tin trên

Viết phương thức **Main()** tạo 1 đối tượng sinh viên (dùng constructor để tạo) và xuất thông tin sinh viên lên màn hình.

### File Student.cs

```
namespace ThuchanhOOP
{
    class Student : Person
    {
        private string studentID;

        public string StudentID { get => studentID; set => studentID = value; }

        public Student()
        {
            Name = "";
            EMail = "";
            StudentID = "";
        }
        public Student(string _name, string _email, string _studentID)
        {
            Name = _name;
            EMail = _email;
            StudentID = _studentID;
        }

        public string Show()
        {
            string result = $"Ho ten: {Name}\nMail: {EMail}\nMSSV: {StudentID}";
            return result;
        }
    }
}
```

### File Program.cs

```
using System;
namespace ThuchanhOOP
{
    class Program
    {
        static void Main(string[] args)
        {
            Student stu = new Student("Nguyen Van Cuong", "cuong@gmail.com", "012312");
            Console.WriteLine(stu.Show());
            Console.ReadKey();
        }
    }
}
```

### Bài 3. Gọi constructor của lớp cơ sở

Trong Bài 2, chỉnh sửa lại **parameter constructor** của lớp **Student** sao cho constructor này có thể gọi constructor của lớp cơ sở (**Person**) để nhờ khởi tạo **name** và **email** khi tạo đối tượng sinh viên.

#### File Student.cs

```
namespace ThuchanhOOP
{
    class Student : Person
    {
        private string studentID;

        public string StudentID { get => studentID; set => studentID = value; }

        public Student() : base()
        {
            StudentID = "";
        }
        public Student(string _name, string _email, string _studentID) : base(_name,
                                                                                   _email)
        {
            StudentID = _studentID;
        }

        public string Show()
        {
            string result = base.Show() + $"{"\nMSSV: {StudentID}"}";
            return result;
        }
    }
}
```

### Bài 4. Xây dựng lớp Tam giác và lớp Chữ nhật

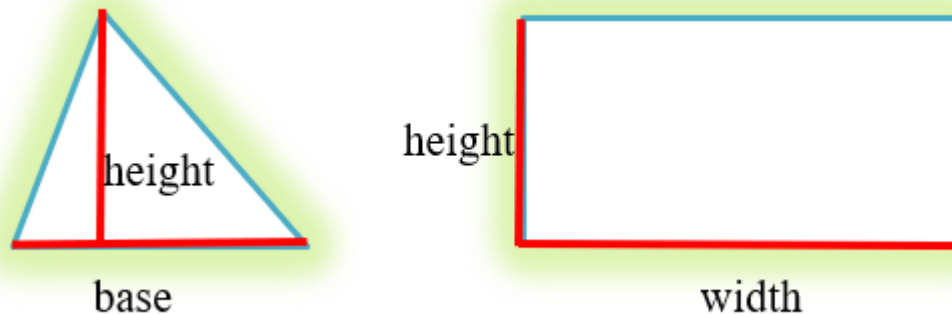
#### Lớp cơ sở: Lớp hình (Shape)

Xây dựng lớp hình (**Shape**) dùng để mô hình hóa một hình gồm có thông tin chiều cao (**height**) và chiều rộng (**width**) của hình.

- Khai báo các **Fields**
- Cài đặt **default constructor** để tạo đối tượng có giá trị mặc định
- Cài đặt **parametered constructors** để tạo đối tượng có 2 thông tin trên
- Cài đặt các **properties**

**Lớp dẫn xuất:** Xây dựng lớp tam giác (**Triangle**) và lớp hình chữ nhật (**Rectangle**)

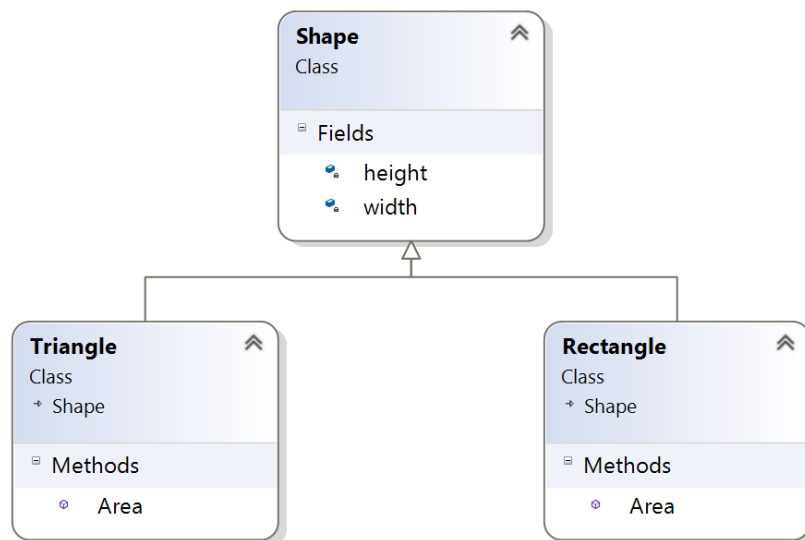
Hình Tam giác và hình chữ nhật cũng là một hình có các thông tin như sau:



Trong đó tham số base của tam giác có thể coi như là width

Xây dựng 2 lớp dẫn xuất là lớp hình tam giác (**Triangle**) và lớp hình chữ nhật (**Rectangle**) **thừa kế từ** lớp **Shape**.

Sơ đồ kế thừa:



- Cài đặt **default constructor**
- Cài đặt **parametered constructors**
- Cài đặt các **properties** (Chú ý: lớp **Triangle** tạo property có tên là **Base** thay vì **Width**)
- Cài đặt thêm phương thức **Area()** để tính diện tích các hình tương ứng

$$S_{tam\ gi\acute{a}c} = \frac{1}{2} \times height \times base$$

$$S_{ch\ddot{u} nh\hat{a}t} = height \times width$$

### File Shape.cs

```
namespace ThuchanhOOP
{
    class Shape
    {
        private int height;
        private int width;

        public int Height { get => height; set => height = value; }
        protected int Width { get => width; set => width = value; }

        public Shape()
        {
            Height = 1;
            Width = 1;
        }
        public Shape(int h, int w)
        {
            Height = h;
            Width = w;
        }
    }
}
```

### File Triangle.cs

```
namespace ThuchanhOOP
{
    class Triangle : Shape
    {
        public int Base { get => Width; set => Width = value; }

        public Triangle() : base()
        { }
        public Triangle(int h, int b) : base(h, b)
        { }

        public double Area()
        {
            return 0.5 * Height * Base;
        }

        public string Show()
        {
            return $"Chieu cao= {Height} \nDay= {Base} \nDien tich = {Area():0.000}";
        }
    }
}
```

### File Rectangle.cs

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

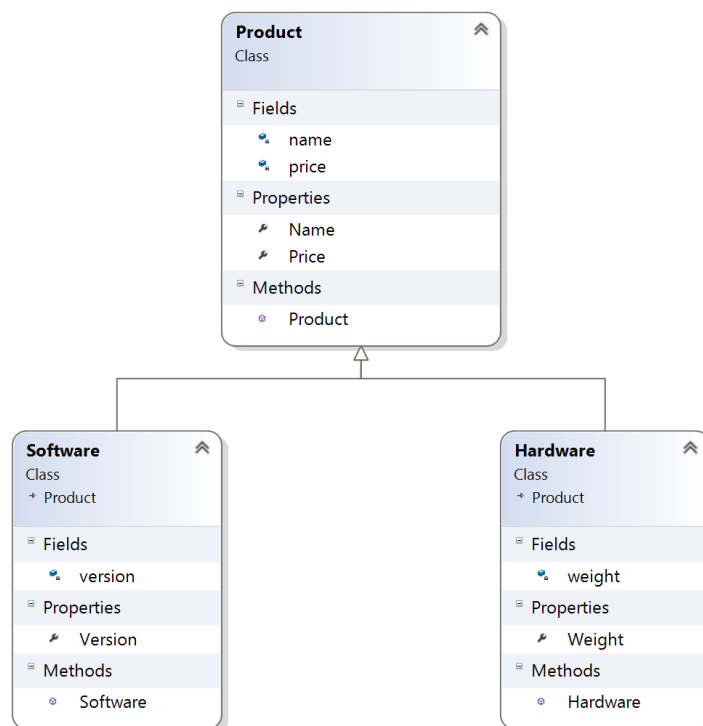
        public Rectangle()
        {
            Height = 1;
            Width = 1;
        }
        public Rectangle(int h, int w)
        {
            Height = h;
            Width = w;
        }

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

        public string Show()
        {
            return $"Chieu cao= {Height} \nChieu rong= {Width} \nDien tich = {Area():0.000}";
        }
    }
}
```

### Bài 5. Xây dựng lớp dựa trên sơ đồ

Cho sơ đồ lớp như sau



Hãy xây dựng các lớp thừa kế theo sơ đồ trên

- **Product** có constructor với 2 tham số
- **Software** có constructor với 3 tham số;
- **Hardware** có constructor với 3 tham số
- Tạo đối tượng kiểu **Software** và xuất thông tin đối tượng ra màn hình
- Tạo đối tượng kiểu **Hardware** và xuất thông tin đối tượng ra màn hình

### File Product.cs

```
namespace ThuchanhOOP
{
    class Product
    {
        private string name;
        private double price;

        public string Name { get => name; set => name = value; }
        public double Price { get => price; set => price = value; }

        public Product()
        {
            Name = "";
            Price = 0;
        }

        public Product(string _name, double _price)
        {
            Name = _name;
            Price = _price;
        }

        public string Show()
        {
            return $"Ten: {Name} \nGia: {Price}";
        }
    }
}
```

### File Software.cs

```
namespace ThuchanhOOP
{
    class Software : Product
    {
        private string version;
        public string Version { get => version; set => version = value; }
        public Software() : base()
        {
            Version = "";
        }
        public Software(string _name, double _price, string _version) : base(_name, _price)
        {
        }
    }
}
```



## Bài giải Thực hành Lập trình Hướng đối tượng – Buổi 5

```
        Version = _version;
    }

    public string Show()
    {
        return base.Show() + $"\\nPhien ban: {Version}";
    }
}
```

### File Hardware.cs

```
namespace ThuchanhOOP
{
    class Hardware : Product
    {
        private double weight;
        public double Weight { get => weight; set => weight = value; }

        public Hardware() : base()
        {
            Weight = 0;
        }
        public Hardware(string _name, double _price, double _weight) : base(_name,
                                                                                   _price)
        {
            Weight = _weight;
        }

        public string Show()
        {
            return base.Show() + $"\\nTrong luong: {Weight:0.00}";
        }
    }
}
```

### File Program.cs

```
using System;

namespace ThuchanhOOP
{
    class Program
    {
        static void Main(string[] args)
        {
            Software soft = new Software("Visual Studio 2021", 340, "15.1.2");
            Console.WriteLine(soft.Show());

            Console.WriteLine();

            Hardware hard = new Hardware("Thin IBM PC", 790, 3.7);
            Console.WriteLine(hard.Show());

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