

BÀI GIẢNG LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

CHƯƠNG 3

LỚP VÀ ĐỐI TƯỢNG

TRẦN THỊ THU THẢO

BỘ MÔN TIN HỌC QUẢN LÝ, KHOA THỐNG KÊ – TIN HỌC

TRƯỜNG ĐẠI HỌC KINH TẾ, ĐẠI HỌC ĐÀ NẴNG

THAOTRAN@DUE.EDU.VN

NỘI DUNG

1. Lớp

2. Đối tượng

3. Phương thức

4. Phương thức khởi tạo

5. Phương thức hủy

LỚP (CLASS)

□ KHÁI NIỆM

Một **lớp (class)** trong lập trình hướng đối tượng đại diện cho tập hợp các đối tượng có cùng các đặc điểm, hành vi, phương thức hoạt động.

Trong đó:

- Đặc điểm của đối tượng được thể hiện ra bên ngoài là các thuộc tính (property) của lớp;
- Các hành vi hay phương thức hoạt động của đối tượng gọi là phương thức của lớp;
- Các thuộc tính và phương thức được gọi chung là thành viên của lớp.

LỚP (CLASS)

❑ KHAI BÁO LỚP:

- Khai báo một lớp bắt đầu bằng từ khóa **class** theo sau là tên của lớp, phần thân của lớp được đánh dấu bằng cặp dấu mở đóng ngoặc nhọn **{}**.

- **Cú pháp**

```
class <tên lớp>
```

```
{
```

```
    <Phạm vi truy cập> <Các thành phần của lớp>;
```

```
}
```

- **<tên lớp>** là tên do người dùng đặt và tuân theo quy tắc đặt tên
- **<Phạm vi truy cập>** bao gồm các từ khoá như **public**, **protected**, **private**, **static**,...
- **<Các thành phần của lớp>** bao gồm các biến, phương thức của lớp.

LỚP (CLASS)

❑ KHAI BÁO LỚP

■ Phạm vi truy cập

Mức độ truy cập	Ý nghĩa
public	Không hạn chế. Thành phần mang thuộc tính này có thể được truy cập ở bất kỳ nơi nào
private	Chỉ có các phương thức của lớp A mới được phép truy cập đến thành phần được đánh dấu private trong các lớp A.
protected	Chỉ có các phương thức của lớp A hoặc của lớp dẫn xuất từ A mới được phép truy cập đến thành phần được đánh dấu protected trong lớp A.
internal	Các thành viên internal trong lớp A được truy xuất trong các phương thức của bất kỳ lớp trong khối kết hợp (assembly) của A
protected internal	Tương đương với protected or internal

ĐỐI TƯỢNG

❑ TẠO ĐỐI TƯỢNG

- **Sử dụng từ khóa “new”**

Là một đối tượng trong thế giới thực, có thuộc tính và hành vi. Trong C#, đối tượng là một thể hiện của lớp. Sử dụng từ khóa **new** để khởi tạo một đối tượng của lớp.

- **Cú pháp khai báo đối tượng và cấp phát vùng nhớ cho đối tượng:**

<TênLớp> <TênBiếnĐốiTượng>;

TênBiếnĐốiTượng = new TênLớp(DanhSáchĐốiSố);

hoặc

TênLớp TênBiếnĐốiTượng = new TênLớp(DanhSáchĐốiSố);

ĐỐI TƯỢNG

❑ TẠO ĐỐI TƯỢNG

- Ví dụ

```
HìnhChuNhat h;  
h = new HìnhChuNhat();
```

Hoặc

```
HìnhChuNhat h = new HìnhChuNhat();
```

PHƯƠNG THỨC

□ KHÁI NIỆM VÀ CÚ PHÁP

- Phương thức (hàm thành viên) là một nhóm lệnh của C# cùng nhau thực hiện một tác vụ
- **Cú pháp khai báo một phương thức**
<mức độ truy cập> <kiểu dữ liệu thuộc tính> Tên phương thức(<tham số>)

```
{  
    // thân phương thức  
}
```

Ví dụ:

```
public float Chuvi()  
{  
    return (Dai+Rong)*2;  
}
```


PHƯƠNG THỨC

□ KHÁI NIỆM VÀ CÚ PHÁP

- Cú pháp gọi một phương thức của đối tượng

<Tên_Đối_Tượng>.<Tên_PhươngThức>(<các tham số nếu có>);

Ví dụ:

```
h.Nhap();  
h.Xuat();  
h.Chuvi();
```

PHƯƠNG THỨC

Ví dụ 1: Viết chương trình tính chu vi hình chữ nhật

```
class HìnhChuNhat  
{
```

HìnhChuNhat là tên của lớp

```
    public float Dai;  
    public float Rong;  
    public void Nhap()  
{
```

Dai, Rong là 2 thuộc tính

```
        Console.WriteLine("Nhap chieu dai hình chu nhật :");  
        Dai = float.Parse(Console.ReadLine());  
        Console.WriteLine("Nhap chieu rong hình chu nhật :");  
        Rong = float.Parse(Console.ReadLine());  
    }
```

Nhap(), Xuat(), Chuvi()
là phương thức

```
    public void Xuat()  
{
```

Kiểu trả về là **void** nên thân phương thức **không chứa lệnh return giá trị**

```
        Console.WriteLine("Hình chu nhật: Dai = {0}, Rong = {1}",  
        Dai, Rong);  
    }
```

```
    public float Chuvi()  
{
```

Kiểu trả về là **float (int/ double/..)** nên thân phương thức **phải chứa lệnh return theo đúng kiểu đã quy định**

```
        return (Dai+Rong)*2;  
    }
```

PHƯƠNG THỨC

Ví dụ 1: Viết chương trình tính chu vi hình chữ nhật

```
class Program
{
    static void Main(string[] args)
    {
        HìnhChuNhat h = new HìnhChuNhat();
        h.Nhap();
        h.Xuat();
        Console.WriteLine("Chu vi HCN: {0}", h.Chuvi());
        Console.ReadLine();
    }
}
```

Tạo đối tượng

Gọi phương thức của đối tượng

TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE

```
PS E:\OOP> cd ViduClass
PS E:\OOP\ViduClass> dotnet run "e:\OOP\ViduClass\Program.cs"
Nhập chiều dài hình chữ nhật :10.2
Nhập chiều rộng hình chữ nhật :3.5
Hình chữ nhật: Dài = 10.2, Rộng = 3.5
Chu vi hình chữ nhật: 27.4
```

PHƯƠNG THỨC

□ KHÁI NIỆM

▪ Phương thức khởi tạo (Constructor)

Đây là một phương thức đặc biệt của lớp. Nó được thực thi khi khởi tạo đối tượng.

Đặc điểm của phương thức khởi tạo:

- Có tên trùng với tên lớp.
- Không có kiểu trả về.
- Được tự động gọi khi 1 đối tượng thuộc lớp được khởi tạo.
- Nếu như bạn không khai báo bất kỳ phương thức khởi tạo nào thì hệ thống sẽ tự tạo ra phương thức khởi tạo mặc định không đối số và không có nội dung gì.
- Có thể có nhiều constructor bên trong 1 lớp.

PHƯƠNG THỨC KHỞI TẠO

□ KHÁI NIỆM

▪ Phương thức khởi tạo (Constructor)

Có 2 loại phương thức khởi tạo:

• Phương thức khởi tạo **không đối số**:

- Là phương thức khởi tạo không có bất kỳ tham số truyền vào nào.
- Thường dùng để khởi tạo các giá trị mặc định cho các thuộc tính bên trong class khi khởi tạo đối tượng (giá trị mặc định này do người lập trình quyết định).

PHƯƠNG THỨC KHỞI TẠO

❑ CÚ PHÁP

▪ Phương thức khởi tạo (Constructor)

• Phương thức khởi tạo có đối số:

- Là phương thức khởi tạo có tham số truyền vào. Và khi khởi tạo đối tượng để phương thức này được gọi ta cần truyền đầy đủ các tham số.
- Thường dùng để khởi tạo các giá trị cho các thuộc tính bên trong class khi khởi tạo đối tượng (các giá trị này do người khởi tạo đối tượng truyền vào).
- Cú pháp:

TênBiếnĐốiTượng = *new* *TênLớp(DanhSáchĐốiSố);*

PHƯƠNG THỨC KHỞI TẠO

□ VÍ DỤ

- **Ví dụ 2:** Viết chương trình xây dựng một lớp *Time*, trong đó có một phương thức tạo lập nhận tham số có kiểu *DateTime* (kiểu xây dựng sẵn của trình biên dịch) làm tham số khởi tạo gán cho các thành phần dữ liệu của đối tượng thuộc lớp *Time*

PHƯƠNG THỨC KHỞI TẠO

❑ VÍ DỤ

▪ Ví dụ 2:

```
using System;
namespace C3Vd2
{
    public class Time
    {
        // private member variables
        int Year;
        int Month;
        int Date;
        int Hour;
        int Minute;
        int Second = 30;
        public void DisplayCurrentTime( )
        {
            Console.WriteLine("Current time is: {0}/{1}/{2} {3}:{4}:{5}",Date,
Month, Year, Hour, Minute, Second);
        }
    }
}
```


PHƯƠNG THỨC KHỞI TẠO

❑ VÍ DỤ

```
public Time(System.DateTime dt)
{
```

Phương thức khởi tạo có đối số (constructor)

```
    Console.WriteLine("Ham constructor tu dong duoc goi!");
    Year = dt.Year;
    Month = dt.Month;
    Date = dt.Day;
    Hour = dt.Hour;
    Minute = dt.Minute;
```

Khởi tạo các giá trị cho các thuộc tính bên trong class

```
    }
}
class DateTimeConstructorApp
{
```

```
    static void Main()
    {
```

```
        System.DateTime currentTime = System.DateTime.Now;
        Time t = new Time(currentTime);
        t.DisplayCurrentTime( );
        Console.ReadLine();
    }
```

Truyền giá trị vào main

```
    }
}
```

```
PS E:\OOP\C3Vd2> dotnet run
Ham constructor tu dong duoc goi!
Current time is: 22/5/2022 14:55:30
```

PHƯƠNG THỨC KHỞI TẠO

❑ VÍ DỤ

- **Ví dụ 3:** Viết chương trình mô tả số đo của học sinh.

```
using System;
namespace C3Vd3
{
    class Hocsinh
    {
        public double Weight;
        public double Height;
        public void Info()
        {
            Console.WriteLine(" Học sinh có cân nặng là: "
+ Height + " và chiều cao là: " + Weight);
        }
    }
}
```

PHƯƠNG THỨC KHỞI TẠO

□ VÍ DỤ

- **Ví dụ 3:** Viết chương trình mô tả số đo của học sinh.

```
public Hocsinh() //Constructor không có tham số
{
    Weight = 40;
    Height = 150;
}

public Hocsinh(int w, int h) //Constructor có tham số
{
    Weight = w;
    Height = h;
}
}
```

PHƯƠNG THỨC KHỞI TẠO

❑ VÍ DỤ

- **Ví dụ 3:** Viết chương trình mô tả số đo của học sinh.

```
class SodoHocsinh
{
    static void Main()
    {
        Hocsinh hs1 = new Hocsinh();
        hs1.Info();

        Hocsinh hs2 = new Hocsinh(48, 160);
        hs2.Info();
    }
}
```

```
PS E:\OOP\C3Vd3> dotnet run
```

```
Hoc sinh co can nang la: 150 va chieu cao la: 40
```

```
Hoc sinh co can nang la: 160 va chieu cao la: 48
```

PHƯƠNG THỨC HỦY

□ KHÁI NIỆM

▪ Phương thức hủy (destructor)

Đây là phương thức đặc biệt được gọi đến trước khi 1 đối tượng bị thu hồi.

Đặc điểm của phương thức hủy:

- Có tên trùng với tên lớp nhưng để phân biệt với constructor thì ta thêm dấu “~” vào trước tên lớp.
- Không có kiểu trả về.
- Được tự động gọi khi 1 đối tượng thuộc lớp kết thúc “vòng đời” của nó thông qua bộ thu dọn rác tự động GC (Garbage Collection).
- Nếu bạn không khai báo destructor thì C# sẽ tự động tạo ra 1 destructor mặc định và không có nội dung gì.
- Chỉ có 1 destructor duy nhất trong 1 lớp.


PHƯƠNG THỨC HỦY

❑ VÍ DỤ

▪ Phương thức hủy (destructor)

Ví dụ:

```
class Hocsinh
{
    public double Weight;
    public double Height;
    public void Info()
    public Hocsinh(int w, int h) //Constructor có tham số
    {
        Weight = w;
        Height = h;
    }
    ~Hocsinh()
    {
        Console.WriteLine("Doi tuong Hocsinh da bi huy.");
    }
}
```



BÀI TẬP KẾT THÚC CHƯƠNG 3

❑ **Bài 1:** Xây dựng lớp Sach gồm:

- Thuộc tính: Tensach, Tacgia, NamXB, Soluong
- Phương thức:
 - Hàm Khởi tạo, Hàm hủy
 - Hàm Nhập, Xuất

Xuất ra thông tin các loại sách đã nhập ra màn hình

❑ **Bài 2:** Xây dựng lớp Meo gồm:

- Thuộc tính: Ten, Giong, Gioitinh, Tuoi
- Phương thức:
 - Hàm Khởi tạo, Hàm hủy
 - Hàm Nhập, Xuất, Đếm

a) Xuất ra danh sách thông tin mèo đã nhập ra màn hình

b) Đếm số lượng mèo có trong danh sách

BÀI TẬP KẾT THÚC CHƯƠNG 3

❑ **Bài 3:** Xây dựng lớp **Organization** gồm:

- Thuộc tính: **Name, Country, City, Address, Telephone** là Tên Doanh nghiệp, Quốc gia, Thành phố, Địa chỉ, Số điện thoại
- Phương thức:
 - Hàm Khởi tạo, Hàm hủy
 - Hàm Nhập, Xuất – Để nhập và xuất thông tin

Nhập và xác nhận các thông tin đầu vào trong đó yêu cầu:

- Thông tin đầu vào các thuộc tính không được để trống
- Thông tin về Số điện thoại phải đảm bảo đủ 10 chữ số và không được để trống

Xuất ra màn hình thông tin của Doanh nghiệp nếu tất các thông tin nhập vào là hợp lệ.

Trường hợp thông tin không hợp lệ thì thông báo “Nhập thông tin lỗi” và Dừng chương trình.

BÀI TẬP KẾT THÚC CHƯƠNG 3

❑ **Bài 4:** Xây dựng lớp **Triangle** gồm:

- Thuộc tính: **a, b, c** (kiểu int) là 3 cạnh của một tam giác
- Phương thức:
 - Hàm Khởi tạo, Hàm hủy
 - Hàm Nhập, Xuất – Để nhập và xuất thông tin
 - Hàm Tính để tính chu vi và diện tích hình tam giác

Tạo một **Menu** để thực hiện lặp lại các chức năng cho đến khi thoát:

Bấm 1: Nhập các cạnh a,b,c của tam giác

Bấm 2: Tính chu vi và diện tích hình tam giác

Bấm 3: Xuất ra các giá trị a,b,c.

Trường hợp các cạnh tạo được một tam giác thì Xuất ra giá trị Chu vi và Diện tích của tam giác. Trường hợp không tạo được tam giác thì xuất ra màn hình “**Không hợp lệ**”

Bấm 0: Thoát khỏi chương trình