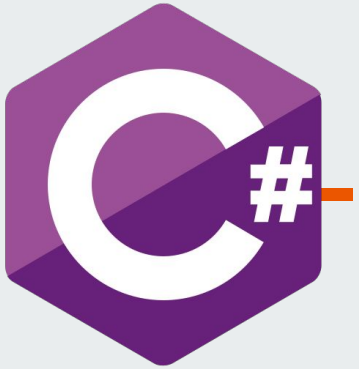

Bài 6

Phương thức khởi tạo và nạp chồng

Mục tiêu

- Phương thức khởi tạo
- Phương thức huỷ
- Nạp chồng phương thức
- Nạp chồng phương thức khởi tạo

Phương thức khởi tạo 1-3

- Phương thức khởi tạo (Constructor) là phương thức đặc biệt có cùng tên với tên lớp, nó có vai trò khởi tạo các thành viên dữ liệu của đối tượng khi nó được tạo.
- Constructor được gọi ngay sau khi khởi tạo đối tượng bằng lệnh new, và tương ứng với mỗi đối tượng nó chỉ được gọi một lần duy nhất.
- Mặc định trình biên dịch sẽ cung cấp một constructor không tham số. Tuy nhiên bạn cũng có thể viết các constructor tường minh cho việc khởi tạo đối tượng.

Phương thức khởi tạo 2-3

```
class Student
{
    //Khai trường
    private string id;
    private string name;
    private double mark;
    //phương thức khởi tạo không tham số
    public Student()
    {
        id = "";
        name = "";
        mark = 0.0;
    }
    //phương thức khởi tạo có tham số
    public Student(string id, string name, double mark)
    {
        this.id = id;
        this.name = name;
        this.mark = mark;
    }
}
```

Phương thức khởi tạo 3-3

- Sử dụng constructor

```
class Program
{
    static void Main(string[] args)
    {
        //tạo đối tượng sử dụng phương thức khởi tạo không tham số
        Student st = new Student();
        //tạo đối tượng sử dụng phương thức khởi tạo có tham số
        Student st1 = new Student("B001", "Le Thanh Tung", 18.0);
    }
}
```

Từ khóa “this”

- Từ khóa this được sử dụng để đại diện cho đối tượng hiện tại, nó dùng để truy cập vào các thành viên trùng tên trong cùng phạm vi (xem ví dụ trang trước)
- Từ khóa this không thể truy cập vào các biến tĩnh và phương thức tĩnh

Phương thức khởi tạo tĩnh

- Phương thức khởi tạo tĩnh được sử dụng để khởi tạo các biến tĩnh hoặc gọi các phương thức tĩnh. Trong một lớp chỉ có duy nhất một constructor tĩnh. Constructor tĩnh không có bất kỳ tham số nào, không có phạm vi truy cập, nó được gọi trực tiếp bởi CLR thay vì đối tượng, nó không thể truy cập tới các thành viên không tĩnh của lớp

```
class Table
{
    string name;
    static int leg;
    static Table()
    {
        leg = 4;
    }
}
```

Phương thức hủy 1-2

- Phương thức hủy(Destructor) là phương thức đặc biệt có cùng tên với tên lớp nhưng khi tạo phương thức các bạn thêm ký tự “~” vào trước tên phương thức. Phương thức destructor sẽ gọi tự động để giải phóng bộ nhớ khi đối tượng khi đối tượng đó không sử dụng nữa.
- Một số đặc điểm của destructor
  - Destructor không thể ghi đè hoặc kế thừa.
  - Destructor không gọi tường minh
  - Destructor không có phạm vi truy cập và không có tham số

Phương thức hủy 2-2

```
class Employee
{
    //khai báo trường private
    private string id;
    private string fullName;
    private int salary;
    //constructor có tham số
    public Employee(string id, string fullName, int salary)
    {
        this.id = id;
        this.fullName = fullName;
        this.salary = salary;
    }
    //phương thức hủy
    ~Employee()
    {
        //các lệnh dọn dẹp rác
        Console.WriteLine("Doi tuong bi huy");
    }
}
```

Nạp chồng phương thức 1-2

- Nạp chồng phương thức (Overloading method) là khả năng của một lớp cho phép định nghĩa nhiều phương thức cùng tên nhau nhưng phải thỏa mãn 1 trong điều kiện sau:
  1. Có tên giống nhau nhưng phải khác nhau về số lượng tham số.
  2. Có tên giống nhau, cùng số lượng tham số nhưng phải khác nhau về kiểu tham số.

Nạp chồng phương thức 2-2

```
class NumberUtility
{
    public static int Sum(int a)
    {
        int s = 0;
        for (int i = 1; i <= a; i++)
        {
            s += i;
        }
        return s;
    }
    public static int Sum(int a, int b)
    {
        int s = 0;
        for (int i = a; i <= b; i++)
        {
            s += i;
        }
        return s;
    }
    public int Add(int a, int b)
    {
        return a + b;
    }
    public double Add(double a, double b)
    {
        return a + b;
    }
}
```

```
class Program
{
    static void Main(string[] args)
    {
        NumberUtility number = new NumberUtility();
        number.Add(
    }
}
```

▲ 1 of 2 ▼ double NumberUtility.Add(**double** a, double b)

Nạp chồng constructor 1-2

- Việc khai báo nhiều constructor trong một lớp được gọi là nạp chồng constructor, việc này hoàn toàn giống với nạp chồng phương thức
- Nạp chồng constructor cho phép người lập trình có nhiều cách tạo ra đối tượng

Nạp chồng constructor 2-2

```
class Student
{
    //Khai trường
    private string id;
    private string name;
    private double mark;
    //phương thức khởi tạo không tham số
    public Student()
    {
        id = "";
        name = "";
        mark = 0.0;
    }
    //nạp chồng phương thức khởi tạo
    public Student(string id, string name, double mark)
    {
        this.id = id;
        this.name = name;
        this.mark = mark;
    }
    //nạp chồng phương thức khởi tạo
    public Student(string id, string name)
    {
        this.id = id;
        this.name = name;
    }
}
```

```
class Program
{
    static void Main(string[] args)
    {
        Student st=new Student()
    }
}
```

▲ 3 of 3 ▼ Student(**string id**, string name, double mark)

HỎI ĐÁP

TRẢI NGHIỆM THỰC HÀNH