Kĩ thuật lập trình với C#

Lập trình Ứng dụng quản lí 2

Nội dung chính

Truyền tham trị vs Truyền tham chiếu

Delegate & Event

Extension methods

Truyền tham chiếu & tham trị

Câu hỏi số 1 - Kết quả là gì?

```
static void Main(string[] args)
{
    string s = "A";
    AppendB(s);

    Console.WriteLine(s);
}

static void AppendB(string s)
{
    s += "B";
}
```

Biến s bên trong main và biến s cục bộ hàm AppendB là **không liên quan** gì với nhau

Câu hỏi số 2 - Kết quả là gì?

```
static void Main(string[] args)
{
    string s = "A";
    AppendB(s);

    Console.WriteLine(s);
}

static void AppendB(string s)
{
    s = "B";
}
```

Biến s bên trong main và biến s cục bộ hàm AppendB là **không liên quan** gì với nhau

Bày binh bố trận, cho trước lớp Cat

```
class Cat
{
    public string Name {get; set;}
    public float Weight { get; set; }

    public override string ToString()
    {
        return string.Format("Name={0}, Weight={1}", Name, Weight);
    }
}
```

Câu hỏi số 3 - Con mèo nặng bao nhiêu kí?

```
static void Main(string[] args)
{
    var cat = new Cat { Name = "Kitty", Weight = 3};
    TakeCare(cat);

    Console.WriteLine(cat.ToString());
}

static void TakeCare(Cat cat)
{
    cat.Weight += 3;
}
```

Biến cat bên trong hàm main và biến cat cục bộ hàm TakeCare không liên quan gì nhau

Truyền rõ ràng là **tham trị**.
Nhưng giá trị gì được truyền?

Tham chiếu - Giữ địa chỉ đến vùng nhớ

Cat kitty = new Cat() { Name = "Kitty", Weight = 3};

Stack	Address	Variable	Value	Size
	1128	kitty	4096	4B

Heap	Address	Member	Value	Size
	4096	Name	Kitty	8B
	4104	Weight	3	8B

Câu hỏi 4 - Vẫn là con mèo cũ không?

```
static void Main(string[] args)
{
    var cat = new Cat { Name = "Kitty", Weight = 3};
    TakeCare(cat);

    Console.WriteLine(cat.ToString());
}

static void TakeCare(Cat cat)
{
    cat = new Cat { Name = "Mimi", Weight = 5};
}
```

Câu hỏi 5 - Biến cục bộ cat có liên quan gì với hàm main?

```
static void Main(string[] args)
{
    var cat = new Cat { Name = "Kitty", Weight = 3};
    TakeCare(ref cat);

    Console.WriteLine(cat.ToString());
}

static void TakeCare(ref Cat cat)
{
    cat = new Cat { Name = "Mimi", Weight = 5};
}
```

Điểm quan trọng cần nhớ

Truyền tham trị vẫn có thể là địa chỉ vùng nhớ

Từ khóa ref tạo ra bí danh - alias

Delegate & Event

Delegate

- Kiểu của hàm
- Ví dụ:

```
delegate int Calculator(int a, int b);
```

Calculator là kiểu hàm nhận hai biến int, trả ra kết quả kiểu int

Hàm nào sau đây tương thích?

```
delegate int Calculator(int a, int b);
```

int Check(int x, int y);

int BeginTransaction(int a, int b, int c);

bool IsValid(string name, int year);

bool IsPrime(int a, int b);

Ví dụ sử dụng Delegate

```
delegate int Calculator(int a, int b);

static int Sum(int a, int b)
{
    return a + b;
}

static int Multiply(int a, int b)
{
    return a * b;
}

static void Main(string[] args)
{
    Calculator func = Sum;
    Console.WriteLine(func(5, 7));

    func = Multiply;
    Console.WriteLine(func(3, 4));
}
```

Hàm nặc danh & biểu thức lambda

```
Calculator f = (int a, int b) => {
    return a / b;
};
Console.Write(f(7, 2));
```

```
Calculator f = (a, b) => a - b;
Console.Write(f(7, 9));
```

Event

Gần giống mảng các con trỏ hàm

```
delegate void Calculator(int a, int b);
static event Calculator actions;

static void Sum(int a, int b) => Console.WriteLine(a + b);
static void Substract(int a, int b) => Console.WriteLine(a - b);

static void Main() {
    var actions = new List<Calculator>();
    actions.Add(Sum);
    actions.Add(Substract);
    actions.Add((a, b) => Console.WriteLine(a / b));
    actions.Add((a, b) => Console.WriteLine(a * b));

    foreach(var action in actions) {
        action(2, 3);
    }
```

```
delegate void Calculator(int a, int b);
static event Calculator actions;

static void Sum(int a, int b) => Console.WriteLine(a + b);
static void Substract(int a, int b) => Console.WriteLine(a - b);

static void Main() {
    actions += Sum;
    actions += Substract;
    actions += (a, b) => Console.WriteLine(a / b);
    actions += (a, b) => Console.WriteLine(a * b);

    actions(2, 3);
```

Điểm quan trọng cần nhớ

Delegate: Kiểu của hàm

Event: Mảng các hàm

Extension methods

Cách viết rút gọn, bổ sung hàm vào lớp có sẵn

```
public static class StringExtension
{
  public static int GetWordCount(this string s) =>
    s.Split().Length;
}
```

Bài luyện tập

Bổ sung hàm kiểm tra chuỗi có độ dài thỏa điều kiện lớn hơn một độ dài nhất định

```
bool HasMinLength(int number) // 6
```