PHÂN TÍCH THIẾT KẾ PHẦN MỀM
.NET 3.0+ Features

Trình bày: Ngô Ngọc Đăng Khoa

.NET 3.0+ Features

Automatic Properties

```
string _prop;
public string Prop
{
    get { return _prop; }
    set { _prop = value; }
}
```

```
public string Prop { get; set; }
```

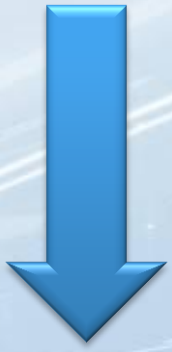
.NET 3.0+ Features

Object Initializer

```
class MyClass
{
    private string _prop1;
    public string Prop1
    {
        get { return _prop1; }
        set { _prop1 = value; }
    }

    string _prop2;
    public string Prop2
    {
        get { return _prop2; }
        set { _prop2 = value; }
    }
}
```

```
MyClass c = new MyClass();
c.Prop1 = "Prop1 value";
c.Prop2 = "null";
```



```
MyClass c = new MyClass { Prop1 = "Prop1 value", Prop2 = "null" };
```

.NET 3.0+ Features

Collection Initializers

```
List<string> list = new List<string>();
list.Add("test1");
list.Add("test2");
```

```
List<string> list = new List<string> { "test1", "test2" };
```

.NET 3.0+ Features

Dictionary Initializers

```
Dictionary<string, string> dict = new Dictionary<string, string>();
dict.Add("key1", "value1");
dict.Add("key2", "value2");
```

```
Dictionary<string, string> dict = new Dictionary<string, string> {
    {"key1", "value1"}, {"key2", "value2"}
};
```

.NET 3.0+ Features

Extension Methods

- Cho phép định nghĩa thêm hàm vào các class sẵn có.
- Cần reference tới **System.Core**

Extension Methods

- VD: mở rộng class **string**

```
static class StringExtensions
{
    public static string AppendMyNameToString(this string value)
    {
        return value + " KHOA";
    }
}
```

```
Console.WriteLine(
    "My name is: ".AppendMyNameToString()
);
```

.NET 3.0+ Features

Anonymous methods (.NET 2.0)

- Không làm rối class với các hàm chỉ dùng 1 lần
- Code được viết ngay chỗ mà nó được dùng
- Không cần đặt tên hàm
- Các class của .NET có hỗ trợ các phương thức dùng anonymous methods
- …

```
List<string> listOfstrings = new List<string>();
//....
listOfstrings.FindAll(
    delegate(string s)
    {
        return s.Contains("a");
    }
);
```

.NET 3.0+ Features

Lambda Expressions

- Phiên bản cải tiến của Anonymous Methods
- Syntax ngắn gọn
- Cấu trúc: ***argument-list => expression***

```
List<string> listOfstrings = new List<string>();
//....
listOfstrings.FindAll(s => s.Contains("a"));
```

```
//anonymous method
numbers.Sort(delegate(int x, int y){ return y-x; });

//lambda expression
numbers.Sort((x,y)=>  y-x);
```

.NET 3.0+ Features

❖Sử dụng Lambda Expression

```
List<int> numbers=GetNumbers();

//find the first number in the list that is below 10
int match=numbers.Find(n=> n<10);

//print all the numbers in the list to the console
numbers.ForEach(n=> Console.WriteLine(n));

//convert all the numbers in the list to floating-point values
List<float> floatNumbers=numbers.ConvertAll<float>(n=> (float)n);

//sort the numbers in reverse order
numbers.Sort((x, y) => y-x);

//filter out all odd numbers
numbers.RemoveAll(n=> n%2!=0);
```

.NET 3.0+ Features

Implicitly typed local variables

- var : cùng kiểu dữ liệu với hàm khởi tạo

```
var i = 3;            // i is implicitly of type int
var s = "sausage";    // s is implicitly of type string

// Therefore:

var rectMatrix = new int[,]     // rectMatrix is implicitly of type int[,]
{
  {0,1,2},
  {3,4,5},
  {6,7,8}
};

var jaggedMat = new int[][]     // jaggedMat is implicitly of type int[][]
{
  new int[] {0,1,2},
  new int[] {3,4,5},
  new int[] {6,7,8}
};
```

.NET 3.0+ Features

Anonymous Types

```
var dude = new { Name = "Bob", Age = 1 };
```

```
internal class AnonymousGeneratedTypeName
{
  private string name;  // actual field name is irrelevant
  private int    age;   // actual field name is irrelevant

  public string  Name get {return name;} {set {name = value;}}
  public int     Age  get {return age; } {set {age = value; }}
}
...

AnonymousGeneratedTypeName dude = new AnonymousGeneratedTypeName (  );
dude.Name = "Bob";
dude.Age = 1;
```

.NET 3.0+ Features

- **Các tính năng khác**
  - Partial Methods
  - Implicitly typed arrays
  - Expression trees
  - **LINQ**

Tài liệu tham khảo

- http://www.codeproject.com/KB/cs/csharp3.aspx
- Joseph Albahari & Ben Albahari, 2007, *C# 3.0 in a Nutshell, 3rd Edition*

Thank You !