TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

LẬP TRÌNH GIAO DIỆN

Nguyễn Thị Mai Trang

1

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Chương 3

Hướng đối tượng trong C#

2

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Mục tiêu

- Hiểu khái niệm và lợi ích của lập trình hướng đối tượng.
- Mô hình hóa được các đối tượng trong thế giới thực thành các lớp đối tượng trong C#.
- Xây dựng và sử dụng các đối tượng trong lập trình để giải quyết vấn đề.

3

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

NỘI DUNG

1. Giới thiệu về Lập trình hướng đối tượng (LTHĐT)
2. Lớp (Class)
3. Phương thức (Method)
4. Các phương thức nạp chồng
5. Phương thức khởi tạo
6. Thuộc tính (Property)
7. Tham chiếu this
8. Dữ liệu và phương thức tĩnh
9. Các cách truyền tham số
10. Thừa kế

4

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

3.1 Giới thiệu

- Mục tiêu của việc thiết kế một phần mềm:
  - Tính tái sử dụng (reusability): thiết kế các thành phần có thể được sử dụng trong nhiều phần mềm khác nhau
  - Tính mở rộng (extensibility): hỗ trợ các plug-ins.
  - Tính mềm dẻo (flexibility):
    - Có thể dễ dàng thay đổi khi thêm mới dữ liệu, chức năng.
    - Các thay đổi không làm ảnh hưởng nhiều đến toàn bộ hệ thống

5

5

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Giới thiệu (tt)

- Quá trình thiết kế phần mềm:
  - Quá trình thiết kế: chia phần mềm và thiết kế theo từng phần, từng component
  - Trừu tượng hóa: bỏ qua những chi tiết của component, quan tâm các thành phần ở mức trừu tượng.
  - Xác định các component: theo hướng top-down
  - Tích hợp: gắn kết các components nhỏ lại với nhau theo hướng bottom-up

6

6

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Giới thiệu (tt)

- Các cách tiếp cận trong thiết kế
  - Thiết kế theo hàm / thủ tục:
    - Tìm ra các hàm/thủ tục để hoàn tất các yêu cầu
    - Kết quả là hệ thống cấu trúc và mối quan hệ giữa các hàm/thủ tục
  - Thiết kế theo module:
    - Phân tích và tìm ra các module bao gồm thành phần dữ liệu và các hàm/thủ tục liên quan
    - Cách thực hiện dựa vào việc gom nhóm các thành phần tương tự nhau về ý nghĩa, phạm vi…

7

7

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Giới thiệu (tt)

- Các cách tiếp cận trong thiết kế (tt)
  - Thiết kế theo hướng đối tượng
    - Trừu tượng hóa dữ liệu và các hàm/thủ tục liên quan
    - Chia hệ thống ra thành các lớp/đối tượng
    - Mỗi lớp/đối tượng có các tính năng và hành động chuyên biệt
    - Các lớp có thể được sử dụng để tạo ra nhiều đối tượng cụ thể
    - Thiết kế các lớp đối tượng và chức năng để có thể trả lời các câu hỏi:
      - Chương trình liên quan tới những lớp đối tượng nào?
      - Mỗi đối tượng cần có những dữ liệu và thao tác nào?
      - Các đối tượng quan hệ với nhau như thế nào trong chương trình?

8

8

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Giới thiệu (tt)

- Lập trình hướng đối tượng:
  - Mô hình hóa đối tượng từ thế giới thực thành các đối tượng có thể lưu trữ và xử lý được trong chương trình.
  - Đối tượng trong thế giới thực: là các thực thể được quan sát trong quá trình thu thập thông tin
    - Các đặc điểm
    - Các hoạt động
  - →Trừu tượng hóa thành:
    - Các thuộc tính
    - Các hành động
  - →Phần mềm:
    - thuộc tính → dữ liệu (các trường - field)
    - hành động → hàm (phương thức).

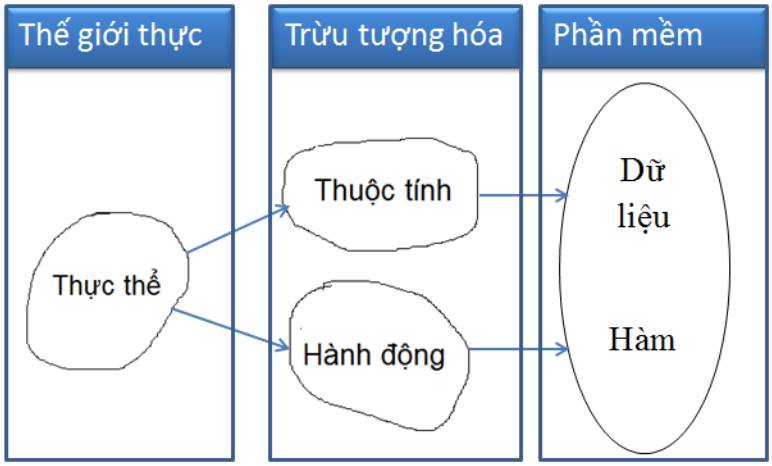
9

9

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Giới thiệu (tt)

- Trừu tượng hóa



10

10

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Giới thiệu (tt)

- Đối tượng:
  - Các thực thể trong hệ thống đều được xem là các đối tượng cụ thể.
  - Đối tượng là một thực thể hoạt động khi chương trình đang chạy, được xác định bằng ba yếu tố:
    - Định danh đối tượng: xác định, nhằm phân biệt các đối tượng với nhau.
    - Trạng thái của đối tượng: tổ hợp các giá trị của các thuộc tính.
    - Hoạt động của đối tượng: là các hành động mà đối tượng có khả năng thực hiện được

11

11

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Giới thiệu (tt)

- Lớp:
  - Khái niệm mang tính trừu tượng → biểu diễn một tập các đối tượng.
  - Lớp cũng có thuộc tính và phương thức.
    - Thuộc tính của lớp tương ứng với thuộc tính mô tả trạng thái của đối tượng, là thành phần dữ liệu mô tả đối tượng.
    - Phương thức của lớp tương ứng với các hành động của đối tượng, dùng để mô tả và thực hiện các hành vi của đối tượng
  - Trong ngôn ngữ lập trình:
    - Đối tượng là thể hiện của lớp, là biến kiểu lớp.

12

12

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Giới thiệu (tt)

- Giả sử, lớp Hinhtron, được định nghĩa để lưu trữ kiểu dữ liệu hình tròn
  - Phương thức Tinhdientich() để tính diện tích hình tròn.
  - Trong đoạn chương trình bên dưới, biến ht chính là đối tượng của lớp Hinhtron.
    - Hinhtron ht = new Hinhtron();
      double dientich = ht.Tinhdientich();

13

13

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

3.2 Lớp (class)

- Trong ngôn ngữ lập trình, lớp (class) là một kiểu cấu trúc mở rộng, được định nghĩa để tạo nên một kiểu dữ liệu mới.
- Một lớp có:
  - các thành phần dữ liệu
  - thuộc tính
  - phương thức
  - phương thức khởi tạo
  - phương thức tĩnh
  - ...

14

14

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

class (tt)

- Sự khác nhau giữa class và struct

Struct	Class
Kiểu giá trị	Kiểu tham chiếu
Không thể khai báo phương thức khởi tạo không tham số vì không thể ghi đè phương thức khởi tạo mặc định.	Có thể khai báo phương thức khởi tạo không tham số.
Không thể khởi tạo giá trị cho biến thành viên khi khai báo (ví dụ: int a;).	Có thể khởi tạo giá trị cho biến thành viên khi khai báo (ví dụ: int a = 10;).
Các biến phải được khởi tạo trong phương thức khởi tạo.	Các biến không bắt buộc phải được khởi tạo trong phương thức khởi tạo, vì chúng sẽ được khởi tạo mặc định.
Không thể kế thừa.	Có thể kế thừa.

15

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

class (tt)

- Khai báo lớp: sử dụng từ khoá class:

```
[MucTruyCap] class TenLop [:LopCoSo]
{
    - Khai báo các trường dữ liệu (các biến)
    - Định nghĩa các phương thức, thuộc tính
}
```

- Khai báo các thành phần của lớp:
  - Khai báo biến thành viên:
    - [MucTruyCap] TenBien [= GiaTri];
  - Khai báo phương thức:
    - [MucTruyCap] TenPhuongThuc ([DanhSachThamSo])
      {
        // mã lệnh
      }

16

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Class (tt)

Mức truy cập	Ý nghĩa
public	Có thể truy xuất từ mọi nơi
internal	Truy xuất trong trong phạm vi lớp, các lớp thừa kế và trong cùng một khối assembly (file .dll, .exe)
protected	Truy xuất trong phạm vi lớp và các lớp thừa kế
private	Chỉ được truy xuất trong nội bộ lớp
protected internal	Có thể truy xuất từ mọi nơi

17

17

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Class (tt)

- Cách sử dụng lớp:
  - Tạo đối tượng của lớp:
    - Tenlop TenBienDoituong;
      TenBienDoituong = new Tenlop ([DanhSachDoiSo]);
  - hoặc
    - Tenlop TenBienDoituong = new Tenlop ([DanhSachDoiSo]);
  - Sử dụng đối tượng đã tạo để truy xuất các thành phần của lớp:
    - TenBienDoituong.TenBien
      TenBienDoituong.TenPhuongthuc ([DanhSachDoiSo])

18

18

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Class (tt)

```
class Hinhtron
{
   private double bankinh;
   public Hinhtron (double ban_kinh)
   {
      bankinh = ban_kinh;
   }
   public double TinhDientich ()
   {
      return bankinh * bankinh * Math.PI;
   }
}
```

```
Hinhtron ht = new Hinhtron (5);
double dientich = ht.TinhDientich ();
```

19

19

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

3.3 Phương thức (Method)

- Dùng để mô tả và thực hiện các hành vi của đối tượng lớp.
- Mỗi phương thức là một hàm.
- Khai báo tương tự C++, thường đặt sau từ khóa public, private, protected,...
- Các phương thức đều phải được khai báo bên trong class.

20

20

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Phương thức (tt)

- Phương thức có trả về giá trị:
  - Trước tên phương thức phải chỉ ra kiểu dữ liệu
  - Cuối phương thức phải có lệnh return để trả về giá trị đúng với kiểu dữ liệu đã khai báo.

```
[MucTruyCap] kieu_dulieu TenPhuongthuc ([CacThamSo])
{
  //code định nghĩa bên trong phương thức
  //return value;
}
```

  - Ví dụ:

```
public int Tonghaiso (int a, int b)
{
  return (a + b);
}
```

21

21

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Phương thức (tt)

- Phương thức không có trả về giá trị:
  - Trước tên phương thức có từ khóa void
  - Cuối phương thức không cần phải có lệnh return (hoặc dùng return; để thoát khỏi phương thức)

```
[MucTruyCap] void TenPhuongthuc ([DanhSachThamSo])
{
  //code định nghĩa bên trong phương thức
}
```

  - Ví dụ:

```
public void InTonghaiso (int a, int b)
{
  Console.Write("Tong cua {0} va {1} la {2}", a, b, a+b);
}
```

22

22

3.4 Các phương thức nạp chồng

- Là các phương thức có cùng tên nhưng khác tham số.
- Khi chương trình thực thi, tùy thuộc vào lời gọi hàm và đối số truyền vào mà phương thức thích hợp nhất được gọi.

Các phương thức nạp chồng (tt)

- Ví dụ

```
class Hinhtron
{
   private double bankinh;
   public Hinhtron (double ban_kinh)
   {
      bankinh = ban_kinh;
   }
  public double TinhDientich ()
   {
      return bankinh * bankinh * Math.PI;
   }
   public double TinhDientich (double bk)
   {
      return bk * bk * Math.PI;
   }
}
```

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

3.5 Phương thức khởi tạo (Constructor)

- Là phương thức có tên trùng với tên của lớp
- Thường được khai báo sau từ khóa public.
- Không có giá trị trả về
- Được gọi một cách tự động khi một đối tượng của lớp được tạo ra.
- Được dùng để khởi tạo các giá trị ban đầu cho các thành phần dữ liệu của đối tượng thuộc lớp.

25

25

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Phương thức khởi tạo (tt)

- class không có phương thức khởi tạo:
  - Trình biên dịch sẽ tự động sử dụng phương thức khởi tạo mặc định.
  - Các thuộc tính không được khởi tạo trong phương thức khởi tạo sẽ được khởi tạo mặc định.
    - Kiểu số: 0
    - Kiểu luận lý: false
    - Kiểu đối tượng: null

26

26

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Phương thức khởi tạo (tt)

- Ví dụ:

```
class Hinhtron
{
   private double bankinh;
   public Hinhtron ()
   {
      bankinh = 1.0;
   }
   public double TinhDientich ()
   {
      return bankinh * bankinh * Math.PI;
   }
}
```

27

27

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Phương thức khởi tạo (tt)

- Nạp chồng phương thức khởi tạo: các phương thức khởi tạo khác tham số

```
class Hinhtron
{
   private double bankinh;
   public Hinhtron ()
   {
      bankinh = 1.0;
   }
   public Hinhtron (double ban_kinh)
   {
      bankinh = ban_kinh;
   }
   ...
}
```

28

28

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Phương thức khởi tạo (tt)

- Phương thức khởi tạo sao chép: khởi gán giá trị cho đối tượng mới bằng cách sao chép dữ liệu của đối tượng cùng kiểu đã tồn tại.

```
class Hinhtron
{
   private double bankinh;
   public Hinhtron ()
   {
      bankinh = 1.0;
   }
   public Hinhtron (Hinhtron ht)
   {
      bankinh = ht.bankinh;
   }
   ...
}
```

29

29

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

3.6 Properties

- Mục đích: tăng sức mạnh của tính đóng gói.
- Các biến dữ liệu của lớp thường được che dấu ở mức độ truy cập private.
- Khai báo properties → truy cập, thay đổi giá trị của các trường dữ liệu mà không cần truy cập trực tiếp vào các trường đó.
- Mỗi property có các mức độ truy cập:
  - get: chỉ cho phép đọc giá trị của trường dữ liệu.
  - set: cho phép gán giá trị cho trường dữ liệu.
  - get và set: cho phép đọc và ghi.

30

30

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Properties (tt)

- Property chỉ đọc: chỉ cho phép truy xuất dữ liệu ở mức độ chỉ đọc, sử dụng từ khóa get

```
class Hinhtron
{
   private double bankinh;
   public Hinhtron ()
   {
      bankinh = 1.0;
   }
   public double Bankinh
   {
      get {return bankinh;}
   }
   …
}
```

```
Hinhtron ht = new Hinhtron();
double bankinh = ht.Bankinh;
ht.Bankinh = 10;//lệnh này bị lỗi cú pháp
```

31

31

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Properties (tt)

- Property chỉ ghi: chỉ cho phép gán giá trị vào trường dữ liệu, sử dụng từ khóa set

```
class Hinhtron
{
   private double bankinh;
   public Hinhtron ()
   {
      bankinh = 1.0;
   }
   public double Bankinh
   {
      set {bankinh = value;}
   }
   …
}
```

```
Hinhtron ht = new Hinhtron();
ht.Bankinh = 10;
double bankinh = ht.Bankinh; //lệnh này bị lỗi cú pháp
```

32

32

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Properties (tt)

- Property đọc và ghi: cho phép truy xuất dữ liệu ở cả hai mức độ đọc và ghi, sử dụng từ khóa get, set

```
class Hinhtron
{
   private double bankinh;
   public Hinhtron ()
   {
      bankinh = 1.0;
   }
   public double Bankinh
   {
      get {return bankinh;}
      set {bankinh = value;}
   }
   …
}
```

```
Hinhtron ht = new Hinhtron();
ht.Bankinh = 10;
double bankinh = ht.Bankinh;
```

33

33

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Properties (tt)

- Lợi ích khi sử dụng Properties:
  - Tăng sức mạnh của tính đóng gói
  - Bảo vệ dữ liệu của đối tượng, không cho phép gán các giá trị bất hợp lệ.
  - Ví dụ, không cho phép gán giá trị âm cho bán kính hình tròn:

```
public double Bankinh
{
   get {return bankinh;}
   set
   {
      bankinh = value;
      if (bankinh < 0) bankinh = 0;
   }
}
```

34

34

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

3.7 Tham chiếu this

- Được dùng để tham chiếu đến thể hiện hiện hành của một đối tượng.
- Tham chiếu this thường được dùng để:
  - Tránh xung đột tên khi tham số của phương thức trùng tên với tên biến dữ liệu của đối tượng.
  - Dùng để truyền đối tượng hiện tại làm tham số cho một phương thức khác.
  - Dùng trong mục đích làm chỉ mục.

35

35

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Tham chiếu this (tt)

- Tránh xung đột tên (nên tránh sử dụng)

```
class Hinhtron
{
  private double bankinh;
  public Hinhtron (double bankinh)
  {
    this.bankinh = bankinh;
  }
}
```

36

36

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Tham chiếu this (tt)

- Làm tham số cho phương thức

```
public partial class Form1 : Form
{
   public Form1()
   {
      InitializeComponent();
   }
   public void ChangeBackground(Form1 f)
   {
      f.BackColor = Color.Blue;
   }
   private void btShow_Click(object sender, EventArgs e)
   {
      ChangeBackground(this);
   }
}
```

37

37

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Tham chiếu this (tt)

- Làm chỉ mục:

```
class Mang {
   int [ ] mang ;
   int sopt;//số phần tử
   public Mang(int n)   {
      sopt = n;
      mang = new int[sopt];
   }
   public int this [int index]   {
      get      {
         if (index >= 0 && index < sopt)  return mang[index];
         return 0;
      }
      set      {
         if (index >= 0 && index < sopt)   mang[index] = value;
      }
   }
}
```

```
//Sử dụng: tạo đối tượng Mang 10 phần tử
Mang arr = new Mang(10);
//gán giá trị cho các phần tử Mang
arr[0] = 10;
arr[1] = 5;
//truy xuất phần tử Mang
int x = arr[0];
```

38

38

3.8 Dữ liệu và phương thức tĩnh

- Là các biến, phương thức được khai báo sau từ khóa static
- Các đối tượng thuộc lớp đều có thể truy cập.
- Thường được dùng trong trường hợp cần lưu trữ giá trị các biến hoặc cung cấp các phương thức dùng chung cho các lớp khác nhau trong ứng dụng.
- Truy cập đến dữ liệu và phương thức tĩnh:
  - Tenlop.TenBien;
    Tenlop.TenPhuongthuc (...);
  - Ví dụ: Math.PI; String.Compare(str1,str2);

39

39

3.9 Các cách truyền tham số

- Truyền bằng tham trị (giá trị):
  - Tương tự C++.
  - Giá trị của đối số sẽ được sao chép vào tham số của phương thức được gọi.
- Truyền bằng tham chiếu (reference):
  - Truyền địa chỉ của biến vào làm đối số cho phương thức.
  - Có hai cách truyền:
    - truyền tham chiếu với từ khóa ref
    - truyền tham chiếu với từ khóa out.

40

40

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Các cách truyền tham số (tt)

- Từ khóa ref: sử dụng trong trường hợp biến truyền vào đã được khởi gán giá trị

```
static void Main(string[] args)
{
   int a = 5, b = 10;
   Hoanvi(ref a, ref b);
   //Kết quả: a = 10, b = 5
}
static void Hoanvi (ref int a, ref int b)
{
   int tmp = a;
   a = b;
   b = tmp;
}
```

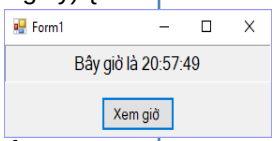
41

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Các cách truyền tham số (tt)

- Từ khóa out:
  - Biến (đối số) truyền vào chưa được gán giá trị trước
  - giá trị của đối số được gán trong phương thức

```
public void LayGioHienTai(out int gio, out int phut, out int giay) {
   gio = DateTime.Now.Hour;
   phut = DateTime.Now.Minute;
   giay = DateTime.Now.Second;
}
private void btHiengio_Click(object sender, EventArgs e) {
   int gio, phut, giay;
   LayGioHienTai (out gio, out phut, out giay);
   lbThoigian.Text =
      String.Format("Bây giờ là {0}:{1}:{2}", gio, phut, giay);
}
```



42

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

3.10 Kế thừa

- Lớp kế thừa (lớp dẫn xuất) là lớp được định nghĩa dựa trên lớp đã có sẵn (lớp cơ sở).
- Có thể bổ sung thêm các thuộc tính và các phương thức cho lớp dẫn xuất.
- Lớp dẫn xuất có hầu hết các thành phần giống như lớp cơ sở, ngoại trừ:
  - các thành phần private
  - phương thức khởi tạo
  - phương thức hủy
  - phương thức tĩnh.

43

43

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Kế thừa (tt)

- Kế thừa thường được dùng để:
  - Phản ánh mối quan hệ giữa các lớp
  - Phản ánh sự chia sẻ mã lệnh chương trình giữa các lớp, không phải viết lặp lại các mã lệnh
- Ví dụ về kế thừa:

Giống nhau		
Mô tả	*Có chân* *Có mắt*	
Hành động	*Biết ăn* *Biết ngủ*	
Khác nhau		
	Chó	***Chim***
Mô tả	*Có răng*	*Có mỏ*
Hành động	*Biết sủa* *Biết chạy*	*Biết hót* *Biết bay*

44

44

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Kế thừa (tt)

- Ví dụ về kế thừa:

Lớp Động vật	
Mô tả	***Hành động***
Có chân *Có mắt*	*Biết ăn* *Biết ngủ*

Lớp Chó	
Mô tả	***Hành động***
Có răng	*Biết sủa* *Biết chạy*

Lớp Chim	
Mô tả	***Hành động***
Có mỏ	*Biết hót* *Biết bay*

45

45

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Kế thừa (tt)

- Khai báo lớp dẫn xuất:
  - class Tenlopdanxuat : Tenlopcoso
    {
      // nội dung
    }
- Ví dụ:
  - class Diem

```
class Diem {
   protected int x, y;
   public int X   {
      get { return x; }
      set { x = value; }
   }
   public int Y   {
      get { return y; }
      set { y = value; }
   }
   public string InToado2D ()   {
      return String.Format("( {0} , {1} )", x, y);
   }
}
```

46

46

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Kế thừa (tt)

– class Diem3D kế thừa class Diem

```
class Diem3D : Diem {
   int z;
   public int Z   {
      get { return z; }
      set { z = value; }
   }
   public string InToado3D()   {
      return String.Format("( {0} , {1} , {2} )", x, y, z);
   }
}
```

– Sử dụng:

```
Diem p2 = new Diem();
p2.X = 100;
p2.Y = 50;
string strDiem = p2.Intoado2D ();
Diem3D p3 = new Diem3D() ;
p3.X = 50 ;
p3.Y = 100 ;
p3.Z = 80 ;
string strDiem3D = p3.InToado3D() ;
string strDiem_2 = p3.InToado2D() ;
```

47

47

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Kế thừa (tt)

- Gọi phương thức thiết lập của lớp cơ sở
  - Lớp cơ sở có phương thức khởi tạo mặc định (không có hoặc có phương thức khởi tạo không có tham số): phương thức khởi tạo của lớp dẫn xuất được định nghĩa một cách bình thường.
  - Lớp cơ sở có phương thức khởi tạo có tham số: định nghĩa phương thức khởi tạo cho lớp dẫn xuất theo cú pháp:

```
TenLopDanXuat (ThamSoLopDanXuat): base (ThamSoLopCoSo)
{
   // Khởi tạo cho các thành phần của lớp dẫn xuất
}
```

48

48