


B môn Công nghệ Phần mềm
Viện CNTT & TT
Trường Đại học Bách Khoa Hà Nội

LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG


Bài 05. K t t p và k th a



M c tiêu bài học

- Giải thích và khái niệm tái sử dụng mã nguồn
- Chỉ ra các bản chất, mô tả các khái niệm liên quan đến k t t p và k th a
- So sánh k t t p và k th a
- Biểu diễn các k t t p và k th a trên UML
- Giải thích nguyên lý k th a và th t kh i t o, h y b i t ng trong k th a
- Áp dụng các kỹ thuật, nguyên lý về k t t p và k th a trên ngôn ngữ lập trình Java


2



N i dung

1. Tái sử dụng mã nguồn
2. K t t p (Aggregation)
3. K th a (Inheritance)


3




N i dung

1. Tái sử dụng mã nguồn
2. K t t p (Aggregation)
3. K th a (Inheritance)


4



1. Tái sử dụng mã nguồn (Re-usability)

5



1. Tái sử dụng mã nguồn (2)

- Các cách sử dụng lập trình đã có:

6

u i m c a tái s d ng mã ngu n



7

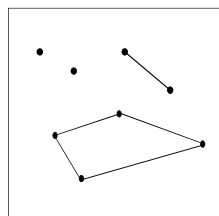
N i dung

1. Tái s d ng mã ngu n
2. K t t p (Aggregation)
3. K th a (Inheritance)

8

2. K t t p

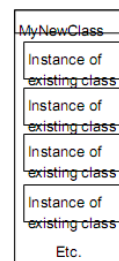
- Ví d :



9

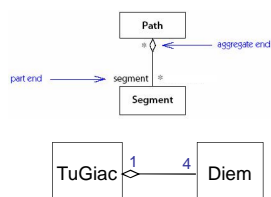
2.1. B n ch t c a k t t p

- L p toàn th ch a i t ng c a l p thành ph n



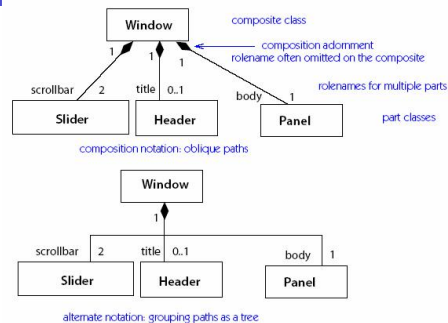
10

2.2. Bi u di n k t t p b ng UML



11

Ví d



12

2.3. Minh họa trên Java

```
class Diem {
    private int x, y;
    public Diem(){}
    public Diem(int x, int y) {
        this.x = x; this.y = y;
    }
    public void setX(int x){ this.x = x; }
    public int getX() { return x; }
    public void hienThiDiem(){
        System.out.print("(" + x + ", "
                          + y + ")");
    }
}
```

13

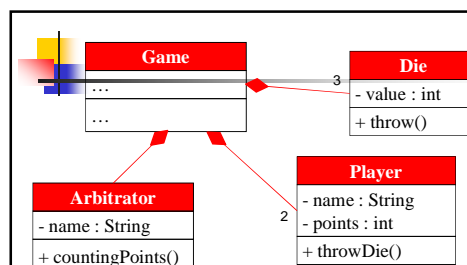
```
class TuGiac {
    private Diem d1, d2;
    private Diem d3, d4;
    public TuGiac(Diem p1, Diem p2,
                  Diem p3, Diem p4){
        d1 = p1; d2 = p2; d3 = p3; d4 = p4;
    }
    public TuGiac(){
        d1 = new Diem();      d2 = new Diem(0,1);
        d3 = new Diem (1,1); d4 = new Diem (1,0);
    }
    public void printTuGiac(){
        d1.printDiem(); d2.printDiem();
        d3.printDiem(); d4.printDiem();
        System.out.println();
    }
}
```

14

Ví dụ khác về K t t p

- Mô tả trò chơi ném 2 viên xúc xắc và 1 trình tài.

15



16

2.4. Thuật toán trong k t t p

17

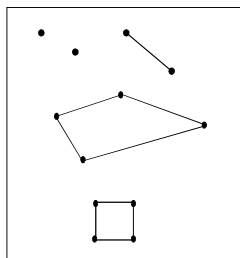
Nội dung

1. Tái sử dụng mã nguồn
2. K t t p (Aggregation)
3. Kế thừa (Inheritance)

18

3.1. T ng quan v k th a

- Ví d :



19

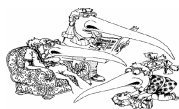
3.1.1. B n ch t k th a

- K th a (Inherit, Derive)

20

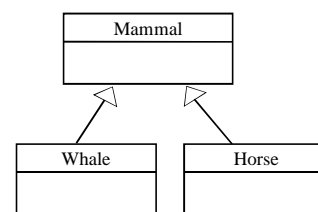
3.1.1. B n ch t k th a (2)

- L p con
 - Là m t lo i (is-a-kind-of) c a l p cha



21

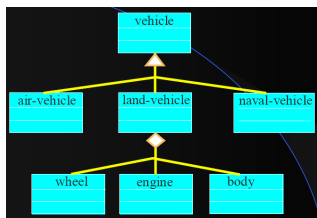
3.1.2. Bi u di n k th a trong UML



22

3.1.3. K t t p và k th a

- So sánh k t t p và k th a?
 - Gi ng nhau?
 - Khác nhau?



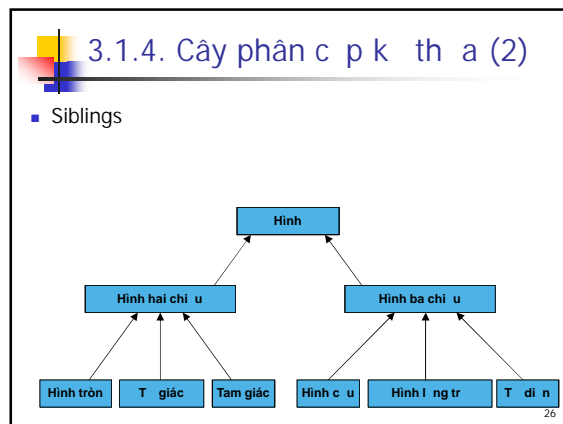
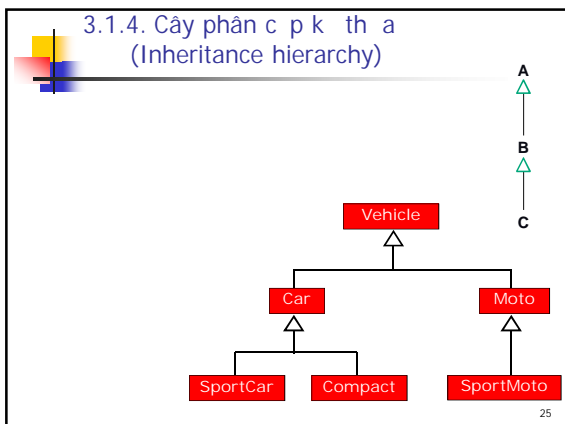
23

Phân bi t k th a và k t t p

K th a

K t t p

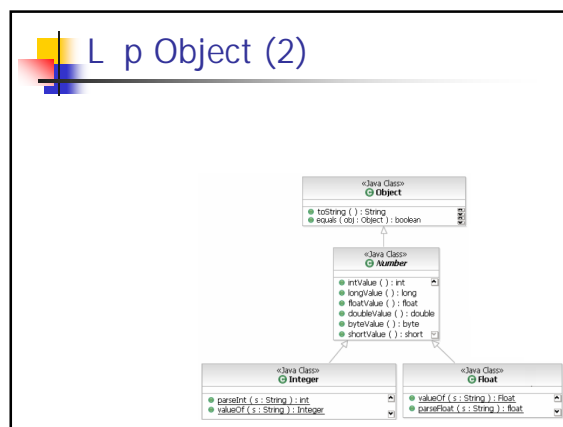
24



L p Object

- Trong gói java.lang

27



3.2. Nguyên lý kế thừa

	public	Không có	protected	private
Cùng l p				
L p con cùng gói				
L p con khác gói				
Khác gói, non-inher				

29

3.2. Nguyên lý kế thừa (2)

- Các tr ng h p không c phép kế thừa?

30

3.3. Cú pháp kế thừa trên Java

- Cú pháp kế thừa trên Java:

- Ví dụ:

```
class HìnhVuong extends TuGiac {
    ...
}
```

31

```
public class TuGiac {
    protected Diem d1, d2, d3, d4;
    public void setD1(Diem _d1) {d1=_d1;}
    public Diem getD1(){return d1;}
    public void printTuGiac(){...}
}

public class HìnhVuong extends TuGiac {
    public HìnhVuong(){
        d1 = new Diem(0,0); d2 = new Diem(0,1);
        d3 = new Diem(1,0); d4 = new Diem(1,1);
    }
}

public class Test{
    public static void main(String args[]){
        HìnhVuong hv = new HìnhVuong();
        hv.printTuGiac();
    }
}
```

32

```
public class TuGiac {
    protected Diem d1, d2, d3, d4;
    public void printTuGiac(){...}
    public TuGiac(){...}
    public TuGiac(Diem d1, Diem d2,
        Diem d3, Diem d4) { ...}
}

public class HìnhVuong extends TuGiac {
    public HìnhVuong(){ super(); }
    public HìnhVuong(Diem d1, Diem d2,
        Diem d3, Diem d4){
        super(d1, d2, d3, d4);
    }
}

public class Test{
    public static void main(String args[]){
        HìnhVuong hv = new HìnhVuong();
        hv.printTuGiac();
    }
}
```

33

Ví dụ 2

```
class Person {
    private String name;
    private Date birthday;
    public String getName() {return name;}
    ...
}

class Employee extends Person {
    private double salary;
    public boolean setSalary(double sal){
        salary = sal;
        return true;
    }
    public String getDetail(){
        String s = name+", "+birthday+", "+salary;
    }
}
```

34

Ví dụ 2

```
class Person {
    protected String name;
    protected Date birthday;
    public String getName() {return name;}
    ...
}

class Employee extends Person {
    private double salary;
    public boolean setSalary(double sal){
        salary = sal;
        return true;
    }
    public String getDetail(){
        String s = name+", "+birthday+", "+salary;
    }
}
```

35

Ví dụ 2 (tiếp)

```
public class Test {
    public static void main(String args[]){
        Employee e = new Employee();
        e.setName("John");
        e.setSalary(3.0);
    }
}
```

36

Ví dụ 3 – Cùng gói

```
public class Person {
    Date birthday;
    String name;
    ...
}
public class Employee extends Person {
    ...
    public String getDetail() {
        String s;
        String s = name + "," + birthday;
        s += ", " + salary;
        return s;
    }
}
```

37

Ví dụ 3 – Khác gói

```
package abc;
public class Person {
    protected Date birthday;
    protected String name;
    ...
}

import abc.Person;
public class Employee extends Person {
    ...
    public String getDetail() {
        String s;
        s = name + "," + birthday + "," + salary;
        return s;
    }
}
```

38

3.4. Khái niệm và phân loại

- Khái niệm:
- Phân loại:

39

3.4.1. Tạo gọi constructor của lớp cha

```
public class TuGiac {
    protected Diem d1, d2;
    protected Diem d3, d4;
    public TuGiac(){
        System.out.println
            ("Lớp cha TuGiac()");
    }
    //...
}

public class Test {
    public static void
        main(String arg[])
    {
        HinhVuong hv =
            new HinhVuong();
    }
}

public class HinhVuong
    extends TuGiac {
    public HinhVuong(){
        //Tu dong gọi TuGiac()
        System.out.println
            ("Lớp con HinhVuong()");
    }
}
```

40

3.4.2. Gọi trực tiếp constructor của lớp cha

- Bất luận lớp nào không có phương thức khởi tạo mặc định

41

Ví dụ

```
public class Test {
    public static
        void main(String
            arg[])
    {
        HinhVuong hv =
            new
                HinhVuong();
    }
}

public class TuGiac {
    protected Diem d1, d2;
    protected Diem d3, d4;
    public TuGiac(Diem d1,
        Diem d2, Diem d3, Diem d4){
        System.out.println("Lớp cha
            TuGiac(d1, d2, d3, d4)");
        this.d1 = d1; this.d2 = d2;
        this.d3 = d3; this.d4 = d4;
    }
}

public class HinhVuong extends TuGiac {
    public HinhVuong(){
        System.out.println
            ("Lớp con HinhVuong()");
    }
}
```

42