


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

LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG


Bài 03. Đóng gói và xây dựng lớp, tổng hợp và sử dụng lớp



Mục tiêu bài học

- Nêu được bản chất, vai trò của trừu tượng hóa
- Giải thích về đóng gói và che giấu thông tin
- Xây dựng lớp
 - Nhận dạng lớp, thể hiện
 - Tạo các phương thức, các thuộc tính
- Tổng hợp và sử dụng lớp
 - Phương thức khởi tạo
 - Khai báo và khởi tạo lớp
 - Sử dụng lớp


2



Nội dung

1. Trừu tượng hóa dữ liệu
2. Đóng gói và xây dựng lớp
3. Tổng hợp và sử dụng lớp


3



Nội dung


1. Trừu tượng hóa dữ liệu
2. Đóng gói và xây dựng lớp
3. Tổng hợp và sử dụng lớp

4



1.1. Trừu tượng hóa

- 2 loại trừu tượng hóa




1.1. Trừu tượng hóa (2)

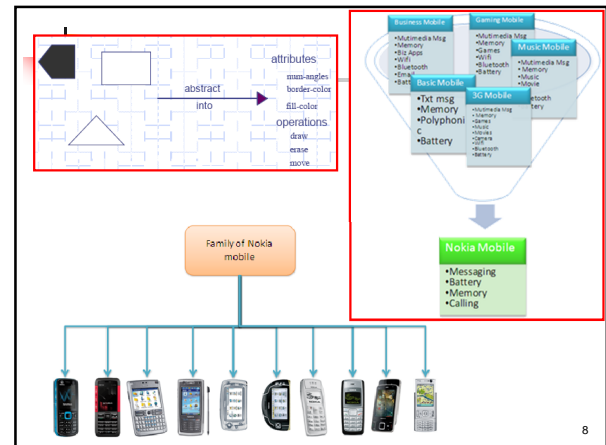
- Trừu tượng hóa khi cần
- Trừu tượng hóa dữ liệu

1.2. Tr u t ng hóa đ li u trong OOP

- it ng trong th c t ph c t p



7




1.2. Tr u t ng hóa đ li u (3)

- Any model that includes the most important, essential, or distinguishing aspects of something while suppressing or ignoring less important, immaterial, or diversionary details. The result of removing distinctions so as to emphasize commonalities (*Dictionary of Object Technology*, Firesmith, Eykholt, 1995).

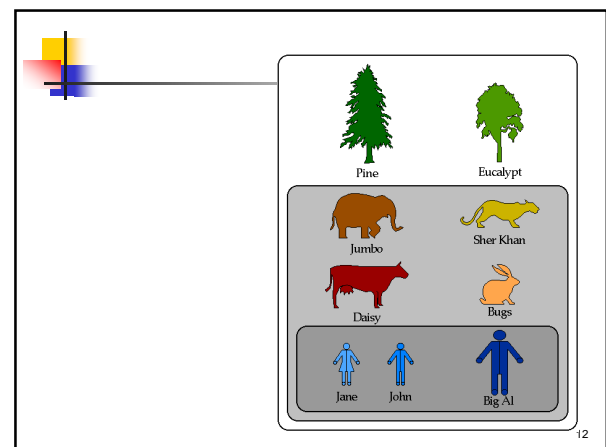
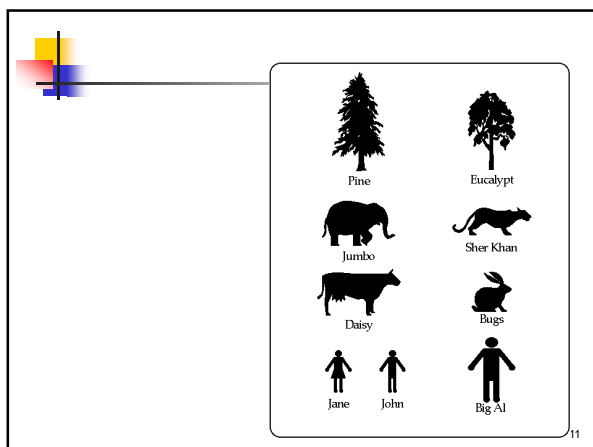
9

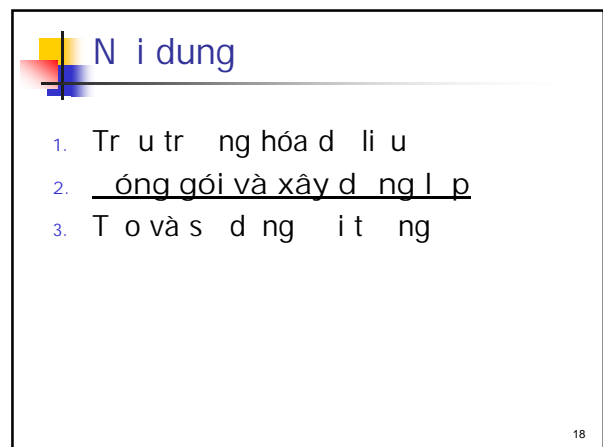
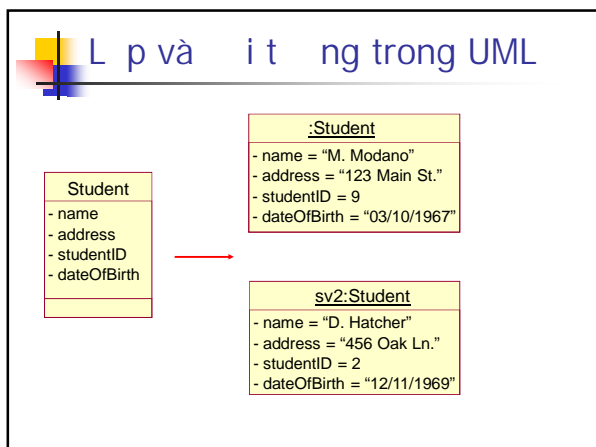
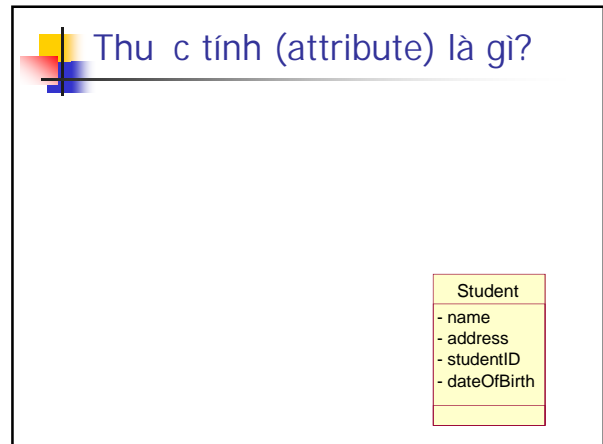
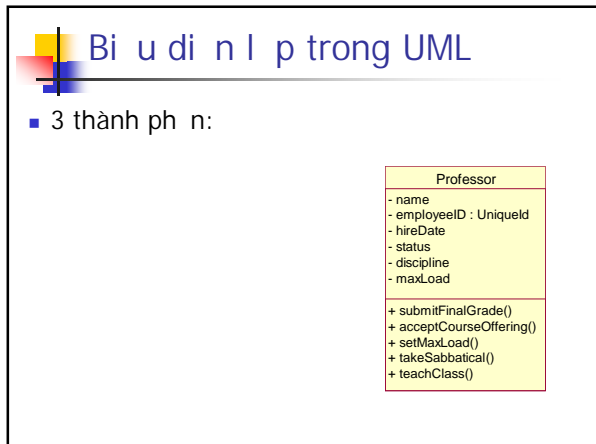
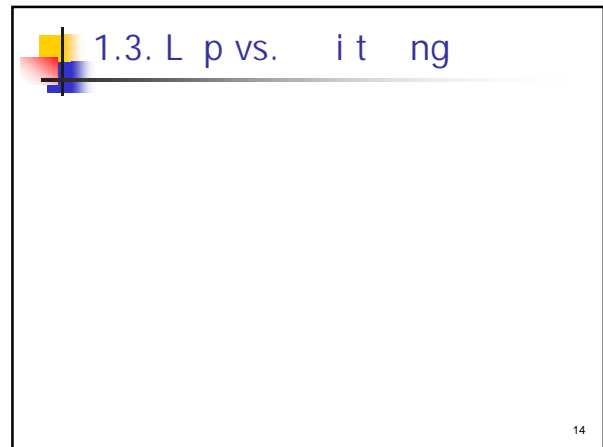
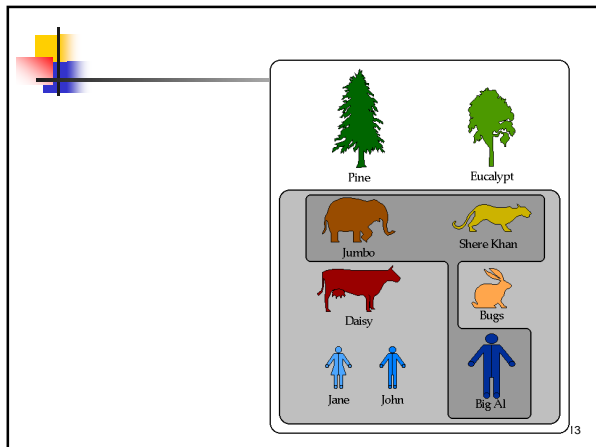
1.2. Tr u t ng hóa đ li u (2)

- T p h p các th h n c a các th c th thành các nhóm có chung các thu c tính



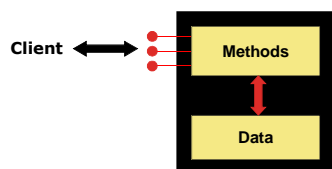
10





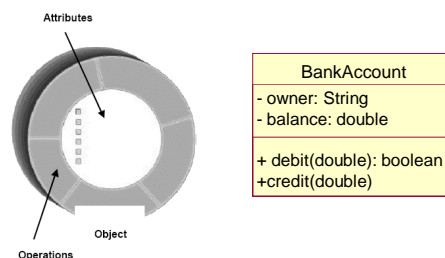
2.1. Đóng gói (Encapsulation)

- Một đối tượng có hai khung nhìn:



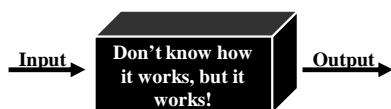
19

2.1. Đóng gói (2)



20

2.1. Đóng gói (3)



21

2.2. Xây dựng lớp

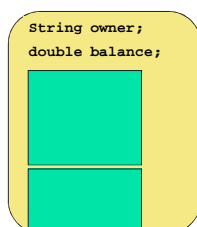
- Thông tin cần thiết để xây dựng lớp

BankAccount
- owner: String
- balance: double
+ debit(double): boolean
+ credit(double)

22

2.2. Xây dựng lớp (2)

- Lớp đóng gói các thành viên (member)



23

2.2. Xây dựng lớp (3)

- Các lớp con nhóm lại thành package

24

2.2.1. Khai báo l p

- Cú pháp khai báo:

```
package tenpackage;
chi_dinh_truy_cap class TenLop {
    // Than lop
}
```

25

Ví d - Khai báo l p

```
package oop.k52.cnpm;

public class Student {
    ...
}
```

26

2.2.2. Khai báo thành viên c a l p

	public	Không có	private
Cùng l p			
Cùng gói			
Khác gói			

27

Ví d : private

```
class Student{
    private String name;
    public String getName() {
        return this.name;
    }
    public void setName(String name)
    {
        this.name = name;
    }
}
```

28

Ví d : private (2)

```
class Student{
    private String name;
    public String getName() {
        return this.name;
    }
    public void setName(String name)
    {
        this.name = name;
    }
}
class Manager{
    private Student[] students;
    public initianize()
    {
        students[0] = new Students();
        //student.name = "Hung"; error
        student.setName("Hung");
    }
}
```

29

a. Thu c tính

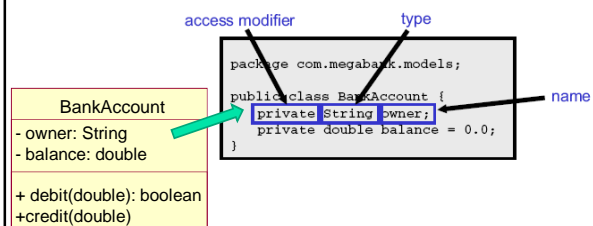
- Các thu c tính ph i c khai báo bên trong l p



30

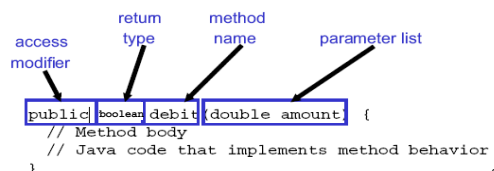
a. Thu c tính (2)

- Thu c tính có th c kh i t o khi khai báo



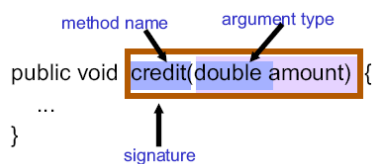
31

b. Ph ng th c



32

* Ch ký ph ng th c (signature)



33

* Ki u d li u tr v

- L nh return

34

Ví d

```

public Boolean checkOdd(int i)
{
    if (i % 2 == 0)
        return true;
    else
        return false;
}

public Boolean checkOdd(int i)
{
    return true;
    return false;
}

```

35

c. Thành viên h ng

- Ví d :


```

final double PI = 3.141592653589793;
public final int VAL_THREE = 39;
private final int[] A = { 1, 2, 3, 4, 5, 6 };

```

```

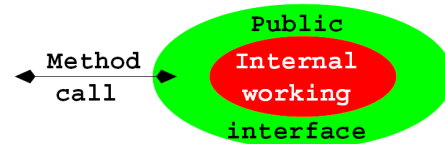
package com.megabank.models;
public class BankAccount {
    private String owner;
    private double balance;

    public boolean debit(double amount){
        if (amount > balance)
            return false;
        else {
            balance -= amount; return true;
        }
    }
    public void credit(double amount){
        balance += amount;
    }
}

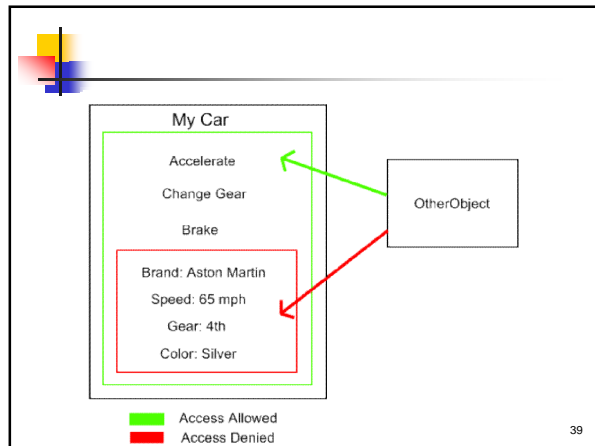
```

37

2.3. Che gi u d li u (Data hiding)

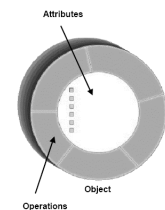


38



39

C ch che gi u d li u



```

BankAccount
- owner: String
- balance: double
+ debit(double): boolean
+ credit(double)

```

40

C ch che gi u d li u (2)

- Accessor (getter):
- Mutator (setter):

```

package com.megabank.models;
public class BankAccount {
    private String owner;
    private double balance = 0.0;
}

```

```

public String getOwner() {
    return owner;
}

```

41

```

public class Time {
    private int hour;
    private int minute;
    private int second;

    public Time () {
        setTime(0, 0, 0);
    }

    public void setHour (int h) { hour = ( ( h >= 0 && h < 24 ) ? h : 0 ); }
    public void setMinute (int m) { minute = ( ( m >= 0 && m < 60 ) ? m : 0 ); }
    public void setSecond (int s) { second = ( ( s >= 0 && s < 60 ) ? s : 0 ); }

    public void setTime (int h, int m, int s) {
        setHour(h);
        setMinute(m);
        setSecond(s);
    }

    public int getHour () { return hour; }
    public int getMinute () { return minute; }
    public int getSecond () { return second; }
}

```

restricted access: *private* members are *not* externally accessible; but we need to know and modify their values

set methods: *public* methods that allow clients to *modify* *private* data; also known as *mutators*

get methods: *public* methods that allow clients to *read* *private* data; also known as *accessors*

Nội dung

1. Tru trình hóa dữ liệu
2. Đóng gói và xây dựng lớp
3. Tổ và sử dụng dữ liệu

43

3.1. Khái niệm dữ liệu



44

Khái niệm và hệ thống dữ liệu

- Tổ
- Hệ

45

3.2. Phân tích khái niệm

- Mục đích chính?



46

3.2. Phân tích khái niệm (2)

- Ví dụ :

```
public BankAccount(String o, double b){
    owner = o;
    balance = b;
}
```

47

3.2. Phân tích khái niệm (3)

- Các chức năng truy cập có thể dùng?
- Các xem xét là thành viên của lớp?

48

3.2. Ph ng th c kh i t o (4)

- Ph ng kh i t o m c nh (default constructor)

49

3.3. Khai báo và kh i t o i t ng

- i t ng c t o ra, th h i n h o a (instantiate) t m t m u chung (l p).

50

3.3. Khai báo và kh i t o i t ng (2)

- i t ng c n c kh i t o t r c khi s d ng

51

3.3. Khai báo và kh i t o i t ng (3)

- Ví d :

```
BankAccount account = new BankAccount();
```

52

3.3. Khai báo và kh i t o i t ng (4)

```
public BankAccount(String name) {
    setOwner(name);
}
```

Constructor
definition

Constructor use

```
BankAccount account = new BankAccount("Joe Smith");
```

53

3.3. Khai báo và kh i t o i t ng (5)

- Ví d :

```
Employee emp1 = new Employee(123456);
Employee emp2;
emp2 = emp1;
Department dept[] = new Department[100];
Test[] t = {new Test(1), new Test(2)};
```

54

Ví dụ 1

```
class BankAccount{
    private String owner;
    private double balance;
}
public class Test{
    public static void main(String args[]){
        BankAccount acc1 = new BankAccount();
    }
}
```

55

Ví dụ 2

```
public class BankAccount{
    private String owner;
    private double balance;
    public BankAccount(){
        owner = "noname";
    }
}
public class Test{
    public static void main(String args[]){
        BankAccount acc1 = new BankAccount();
    }
}
```

56

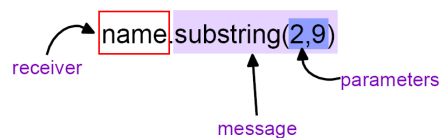
Ví dụ 3

```
public class BankAccount {
    private String owner;
    private double balance;
    public BankAccount(String name){
        setOwner(name);
    }
    public void setOwner(String o){
        owner = o;
    }
}
public class Test{
    public static void main(String args[]){
        BankAccount account1 = new BankAccount();
        BankAccount account2 = new BankAccount("Hoang");
    }
}
```

57

3.4. Sử dụng tính

- tính cung cấp các hoạt động phức tạp hơn các kiểu dữ liệu nguyên thủy



58

3.4. Sử dụng tính (2)

- Toán tử "."

```
BankAccount account = new BankAccount();
account.setOwner("Smith");
account.credit(1000.0);
System.out.println(account.getBalance());
...
```

BankAccount method

```
public void credit(double amount) {
    setBalance(getBalance() + amount);
}
```

59

```
public class BankAccount{
    private String owner;
    private double balance;
    public BankAccount(String name) { setOwner(name); }
    public void setOwner(String o){ owner = o; }
    public String getOwner(){ return owner; }
}
public class Test{
    public static void main(String args[]){
        BankAccount acc1 = new BankAccount("");
        BankAccount acc2 = new BankAccount("Hong");
        acc1.setOwner("Hoa");
        System.out.println(acc1.getOwner()
            + " " + acc2.getOwner());
    }
}
```

60



T tham chi u – this

- Cho phép truy cập vào biến instance.

61

```
public class BankAccount{
    private String owner;
    private double balance;
    public BankAccount() { }
    public void setOwner(String owner){
        this.owner = owner;
    }
    public String getOwner(){ return owner; }
}

public class Test{
    public static void main(String args[]){
        BankAccount acc1 = new BankAccount();
        BankAccount acc2 = new BankAccount();
        acc1.setOwner("Hoa");
        acc2.setOwner("Hong");
        System.out.println(acc1.getOwner() + " " +
                           acc2.getOwner());
    }
}
```

62