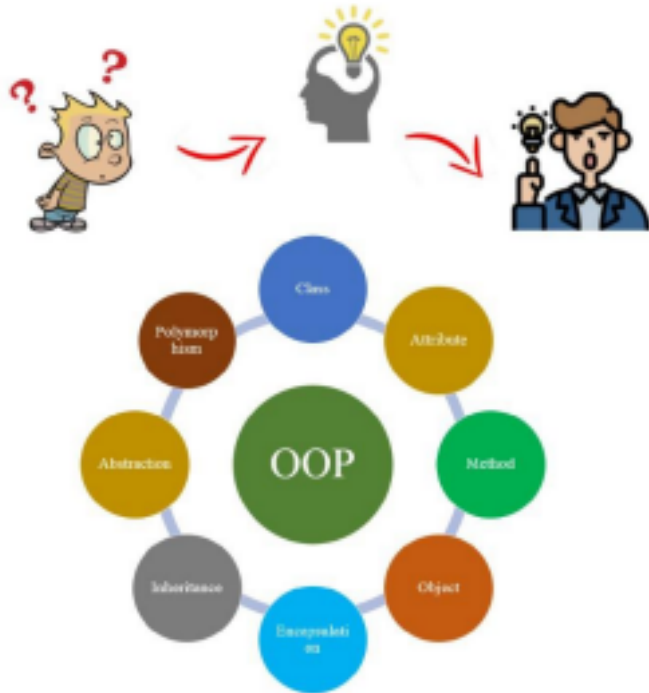


LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG (OOP_Object-Oriented Programming)



Mục Tiêu Bài Học

- ❖ Thế nào là lập trình hướng đối tượng
- ❖ Tìm hiểu về trừu tượng dữ liệu
- ❖ Định nghĩa lớp và đối tượng
- ❖ Constructor và Destructor
- ❖ Tìm hiểu về tính lưu trữ, bao bọc dữ liệu, tính kế thừa và đa hình

❖ Các ưu điểm của phương pháp lập trình hướng đối tượng

NTTU-2022

Lập Trình Hướng Đối Tượng

- Lập trình hướng đối tượng (**OOP-Object-Oriented Programming**) là một phương pháp hay mô hình giúp tăng năng suất, đơn giản hóa việc bảo trì, dễ dàng mở rộng trong thiết kế phần mềm bởi việc cung cấp một vài khái niệm như:
 - Đối tượng (Object)
 - Lớp (Class)
 - Kế thừa (Inheritance)
 - Đa hình (Polymorphism)

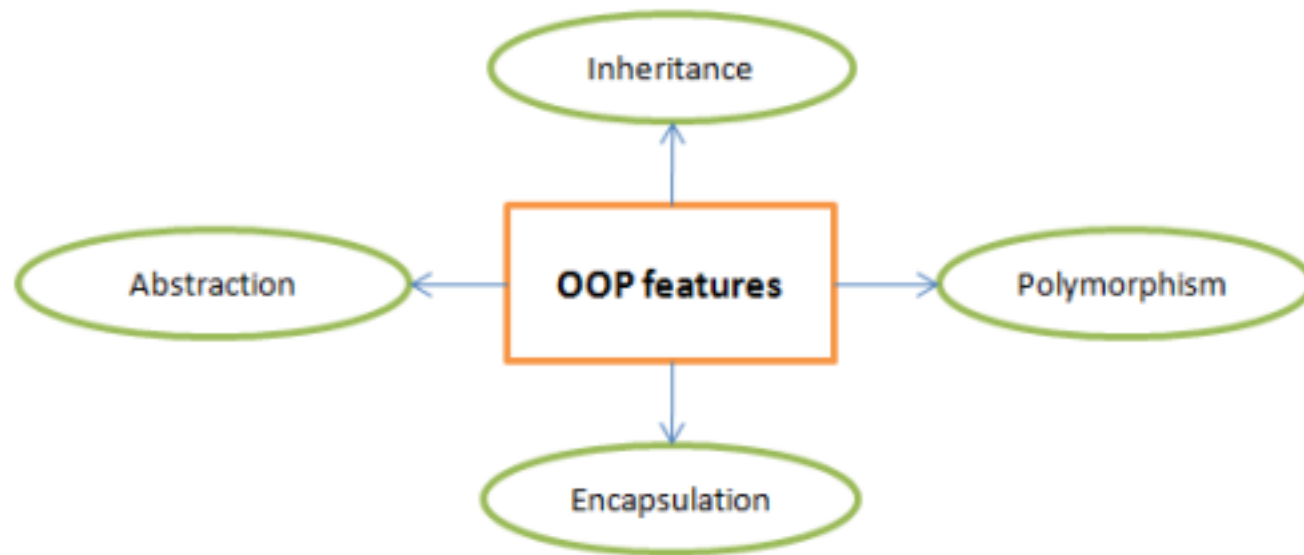
➤ Trừu tượng (Abstraction)

➤ Đóng gói (Encapsulation)

NTTU-2022

Lập trình hướng đối tượng có 4 tính

chất:



NTTU-2022

Trừu Tượng Dữ Liệu

- Là tiến trình xác định và tập hợp các **tính chất và các hành động của một thực thể** có liên quan đến ứng dụng.
- Lợi ích:
 - ✓ Tập trung vào vấn đề
 - ✓ Xác định những tính chất và hành động thiết yếu
 - ✓ Loại trừ những chi tiết không cần thiết

NTTU-2022

Trừu Tượng Dữ Liệu

Các tính chất của một đối tượng Người

Tên
Địa chỉ
Tuổi
Chiều cao

Màu tóc

Các tính chất của một đối tượng Khách hàng

Tên
Địa chỉ

NTTU-2022

Trình Tượng Dữ Liệu (tiếp theo)

Các thuộc tính Các hành động Tên của khách

hàng Nhập tên của khách hàng Địa chỉ của khách hàng Nhập

địa chỉ của khách hàng Đồi xe hơi đã mua Nhập đồi xe hơi
mua được Người bán xe hơi Nhập tên người bán xe hơi Lập
hoá đơn

NTTU-2022

Lớp (Class)

- Lớp là một nhóm các đối tượng có chung những tính chất và hành động

Lớp Khách hàng

Tên khách hàng

Địa chỉ khách hàng

Đồi xe hơi đã mua

Tên người bán xe hơi

Nhập tên

Nhập địa chỉ

Nhập đời của xe hơi mua được

Nhập tên của người bán xe hơi

Lập hóa đơn

NTTU-2022

Lớp

- Một lớp là một nhóm đối tượng có các thuộc tính chung. Nó là một mẫu hoặc thiết kế từ đó các đối tượng được tạo ra.

- **Một lớp trong java có thể chứa:**

✓ Thành viên dữ liệu

- ✓ Constructor
- ✓ Phương thức
- ✓ Khởi lệnh
- ✓ Lớp và Interface

NTTU-2022

Đối Tượng (Object)

- Đối tượng là một thể hiện của lớp
 - Toàn

- Anh
- Tuấn

NTTU-2022

Đối Tượng (tiếp theo)

- Thuộc tính
 - ✓ Tính chất mô tả một đối tượng
- Hành động

✓ Dịch vụ mà đối tượng có thể đáp ứng

- Phương thức

✓ Đặc tả cách đáp ứng bằng hành động khi được yêu

cầu • Thông điệp

✓ Yêu cầu một hành động

- Biến cố

✓ Sự kích thích từ đối tượng này gửi sang đối tượng khác

NTTU-2022

Đối tượng (tt)

- Một đối tượng có ba đặc điểm:

✓ Trạng thái: Đại diện cho dữ liệu (giá trị) của một đối tượng.

- ✓ **Hành vi:** Đại diện cho hành vi (chức năng) của một đối tượng như gửi tiền, rút tiền, ...
- ✓ **Danh tính:** Danh tính của một đối tượng thường được cài đặt thông qua một ID duy nhất. ID này được ẩn đối với user bên ngoài. Tuy nhiên nó được sử dụng trong nội bộ máy ảo JVM để định danh từng đối tượng.

Ví dụ: Bút chì là một đối tượng. Tên của nó là A, màu trắng, ... được gọi là trạng thái. Nó được sử dụng để viết, viết được gọi là hành vi.

12

NTTU-2022

Ví dụ

- Trong ví dụ này, chúng tôi đã tạo ra một lớp Student có hai thành viên dữ liệu **id** và **name**. Chúng ta tạo ra các đối tượng của lớp Student bởi từ khóa `new` và in giá trị của các đối tượng.

```
public class Student {  
    int id; // thành viên dữ liệu  
    String name; // thành viên dữ liệu  
  
    public static void main(String args[]) {  
        Student student1 = new Student(); // tạo một đối tượng student1  
        System.out.println(student1.id);  
        System.out.println(student1.name);  
    }  
} Kết quả: 0 null
```

Lớp Và Đối Tượng

❖ Lớp là một thực thể, còn đối tượng là một thực thể thực tế.

- ❖ Lớp là một mô hình ý niệm định rõ các tính chất và các hành động được quy định bởi một đối tượng, còn đối tượng là một mô hình thực sự.
- ❖ Lớp là khuôn mẫu từ đó đối tượng được tạo ra.
- ❖ Tất cả các đối tượng trong cùng một lớp có các tính chất và các hành động như nhau.

NTTU-2022

Có những cách nào để tạo đối tượng trong java?

- ❖ Sử dụng từ khóa new
- ❖ Sử dụng phương thức newInstance()
- ❖ Sử dụng phương thức clone()
- ❖ Sử dụng phương thức factory,...

Đối tượng Anonymous trong java

- Anonymous nghĩa là vô danh. Một đối tượng không có tham chiếu gọi là đối tượng Anonymous.
- Nếu bạn sử dụng đối tượng 1 lần duy nhất, thì lựa chọn tạo đối tượng Anonymous là tốt nhất trong

trường hợp này

```
public class Calculation {
```

```
void fact(int n) {
```

```
int giaithua = 1;
```

```
for (int i = 1; i <= n; i++) {
```

```
    giaithua = giaithua * i;
```

```
}
```

```
System.out.println("Giải thừa của " + n + " là: " + giaithua);
```

```
}
```

```
public static void main(String args[]) {
```

```
// gọi phương thức của đối tượng anonymous new Calculation().fact(5);
```

```
}
```

```
}
```

Kết quả:

Giai thừa của 5 là: 120_{16}

Sự khác nhau giữa lớp và đối tượng trong java

No	Đối tượng	Lớp
1.	Đối tượng là thể hiện của 1 lớp.	Lớp là một khuôn mẫu hay thiết kế để tạo ra các đối tượng.
2.	Đối tượng là 1 thực thể trong thế giới thực như Bút chì, Xe đạp, ...	Lớp là một nhóm các đối tượng tương tự nhau.
3.	Đối tượng là 1 thực thể vật lý	Lớp là 1 thực thể logic
4.	Đối tượng được tạo ra chủ yếu từ từ khóa new. Ví dụ: Student s1=new Student();	Lớp được khai báo bằng việc sử dụng từ khóa class. Ví dụ: class Student{}
5.	Đối tượng có thể được tạo nhiều lần.	Lớp được khai báo 1 lần duy nhất.
6.	Đối tượng được cấp bộ nhớ khi nó được tạo ra.	Lớp không được cấp bộ nhớ khi nó được tạo ra.
7.	Có rất nhiều cách để tạo ra đối tượng trong java như từ khóa new, phương thức newInstance(), phương thức clone(), phương thức factory và deserialization.	Chỉ có một cách để định nghĩa lớp trong java sử dụng từ khoá class.

17

NTTU-2022

Package trong java

- Package được sử dụng để phân loại lớp và interface giúp dễ dàng bảo trì.
- Package cung cấp bảo vệ truy cập
- Package khắc phục được việc đặt trùng tên

Sử dụng packagename.classname

```
1 package pack;
2 public class A {
3     public void msg() {
4         System.out.println("Hello");
5     }
6 }
```

```
1 package mypack;
2 import pack.A;
3
4 class B {
5     public static void main(String args[]) {
6         A obj = new A();
7         obj.msg();
8     }
9 }
```

Kết quả:

```
Hello
```

Constructor

- Tiến trình tạo ra một đối tượng được gọi là

Constructor • Một Constructor:

- ✓ Cấp phát vùng nhớ
- ✓ Khởi gán những thuộc tính (nếu có)
- ✓ Cho phép truy cập những thuộc tính và phương thức

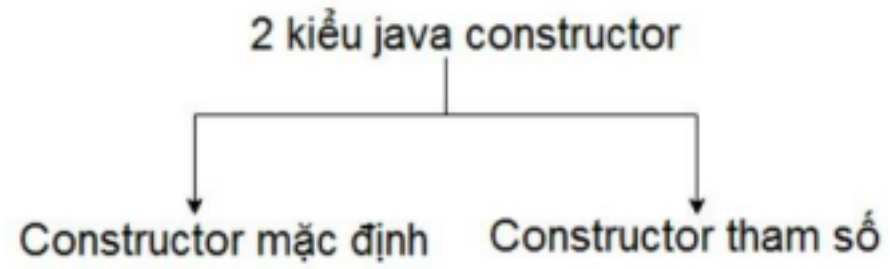
Constructor

- Các quy tắc tạo constructor trong java. Có 2 quy tắc cơ bản cho việc tạo constructor:
 - Tên constructor phải giống tên lớp chứa nó.
 - Constructor không có kiểu trả về tường minh.

NTTU-2022

Các kiểu của java constructor

- Có 2 kiểu của constructor
 - ✓ Constructor mặc định (không có tham số truyền vào)
 - ✓ Constructor tham số



Constructor mặc định trong java

Cú pháp của constructor mặc định:

```
1 | <class_name>() {  
2 |     // code  
3 | }
```

Ví dụ về constructor mặc định:

```
1 | class Bike1 {  
2 |     Bike1() {  
3 |         System.out.println("Bike is created");  
4 |     }  
5 |     public static void main(String args[]) {  
6 |         Bike1 b=new Bike1();  
7 |     }  
8 | }
```

Output:

```
Bike is created
```

Sự khác nhau giữa constructor và phương thức trong java

Constructor	Phương thức
Constructor được sử dụng để khởi tạo trạng thái của một đối tượng.	Phương thức được sử dụng để thể hiện hành động của một đối tượng.
Constructor không có kiểu trả về.	Phương thức có kiểu trả về.
Constructor được gọi ngầm.	Phương thức được gọi tường minh.
Trình biên dịch Java tạo ra constructor mặc định nếu bạn không có constructor nào.	Phương thức không được tạo ra bởi trình biên dịch Java.
Tên của constructor phải giống tên lớp.	Tên phương thức có thể giống hoặc khác tên lớp.

Destructor

- Hàm huỷ sẽ tự động được gọi trước khi giải phóng một đối tượng để giải phóng vùng nhớ trước khi đối tượng được huỷ bỏ. Hàm huỷ có các đặc điểm sau:
 - – Mỗi lớp chỉ có một hàm huỷ.
 - – Hàm huỷ không có kiểu, không có giá trị trả về và không có đối số.
 - – Tên hàm huỷ cùng tên với tên lớp và có một dấu ngã (~) ngay trước tên

Destructor

- Tiến trình hủy một đối tượng gọi là

Destructor • Một Destructor:

- ✓ Giải phóng bộ nhớ
- ✓ Cấm truy cập thuộc tính và phương thức

VD

```
1  #include <iostream>
2  using namespace std;
3
4  class Circle{
5  private:
6      float r;
7  public:
8      //Hàm khởi tạo có tham số
9      Circle(float r){
10         this->r = r;
11     }
12     void setRadius(float r){
13         this->r = r;
14     }
15     float getRadius(){
16         return r;
17     }
18     float calculateArea(){
19         return 3.14 * r * r;
20     }
21     ~Circle(){
22         cout<<"Radius of circle has destroyed."<<endl;
23     }
24 };
25
26 void main()
27 {
28     Circle c1(3.5);
29     cout<<"Area of Circle = "<<c1.calculateArea();
30 }
```

Area of Circle = 3.14

NTTU-2022

Từ khóa **this** trong java

- ❖ Từ khóa this trong java là một biến tham chiếu được sử dụng để tham chiếu tới đối tượng của lớp hiện tại.
- ❖ Trong java, Từ khóa this có 6 cách sử dụng như sau:

- Từ khóa `this` có thể được dùng để tham chiếu tới biến instance của lớp hiện tại.
- `this()` có thể được dùng để gọi Constructor của lớp hiện tại.
- Từ khóa `this` có thể được dùng để gọi phương thức của lớp hiện tại.
- Từ khóa `this` có thể được truyền như một tham số trong phương thức.
- Từ khóa `this` có thể được truyền như một tham số trong phương Constructor.
- Từ khóa `this` có thể được dùng để trả về instance của lớp hiện tại.

```
1 public class Student11 {
2     int id;
3     String name;
4
5     Student11(int id, String name) {
6         this.id = id;
7         this.name = name;
8     }
9
10    void display() {
11        System.out.println(id + " " + name);
12    }
13
14    public static void main(String args[]) {
15        Student11 s1 = new Student11(111, "Viet");
16        Student11 s2 = new Student11(222, "Nam");
17        s1.display();
18        s2.display();
19    }
20 }
```

Kết quả:

```
111 Viet
222 Nam
```

29

NTTU-2022

VD

Nếu biến cục bộ và biến toàn cục có tên khác nhau thì không cần sử dụng từ khóa this.

```
1 public class Student12 {  
2     int id;  
3     String name;  
4  
5     Student12(int i, String n) {  
6         id = i;  
7         name = n;  
8     }  
9  
10    void display() {  
11        System.out.println(id + " " + name);  
12    }  
13  
14    public static void main(String args[]) {  
15        Student12 e1 = new Student12(111, "Viet");  
16        Student12 e2 = new Student12(222, "Nam");  
17        e1.display();  
18        e2.display();  
19    }  
20 }
```

Kết quả:

```
111 Viet  
222 Name
```

Sử dụng this() gọi Constructor của lớp hiện tại.

- Phương thức this() có thể được sử dụng để gọi Constructor của lớp hiện tại. Cách sử dụng này sẽ hữu dụng hơn nếu bạn có nhiều Constructor trong một lớp và bạn muốn sử dụng lại Constructor.

```
1 public class Student13 {
2     int id;
3     String name;
4
5     Student13() {
6         System.out.println("call Constructor mặc định");
7     }
8
9     Student13(int id, String name) {
10        this(); // nó được sử dụng để gọi Constructor của lớp hiện tại
11        this.id = id;
12        this.name = name;
13    }
14
15    void display() {
16        System.out.println(id + " " + name);
17    }
18
19    public static void main(String args[]) {
20        Student13 e1 = new Student13(111, "Việt");
21        Student13 e2 = new Student13(222, "Nam");
22        e1.display();
23        e2.display();
24    }
25 }
```

Kết quả:

```
call Constructor mặc định
call Constructor mặc định
111 Việt
222 Nam
```

Tính Lưu Trữ

- Tính lưu trữ là khả năng của đối tượng có thể lưu lại dữ liệu của nó sau khi đã bị hủy

Tính Bao Bọc Dữ Liệu

- Tiến trình **che dấu những chi tiết hiện thực** một đối tượng được gọi là tính bao bọc
- Ưu điểm:
 - ✓ Tất cả những thuộc tính và phương thức cần thiết đều được tạo
 - ✓ Một lớp có thể có nhiều tính chất và phương thức nhưng chỉ một số trong đó được hiển thị cho người dùng

Tính Kế Thừa

LỚP SINH VIÊN LỚP NHÂN VIÊN LỚP KHÁCH HÀNG

Tên
Địa chỉ
Lương
Chức vụ
Nhập tên
Nhập địa chỉ
Nhập lương

Tên Tên

Địa chỉ Địa chỉ

Điểm môn 1 Sản phẩm mua được

Điểm môn 2 Nhập tên

Nhập tên Nhập địa chỉ Nhập địa chỉ Nhập mã sản phẩm Nhập điểm Lập

hóa đơn Tính tổng số điểm Tính lương

NTTU-2022

Tính Kế Thừa (tiếp theo)

Lớp Người

Tên

Địa chỉ

Nhập tên

Nhập địa chỉ

Tính Kế Thừa (tiếp
theo)

NTTU-2022

Địa chỉ

Nhập tên

Nhập địa chỉ

+

LỚP NGƯỜI

Tên

Thêm các thuộc tính và hành động Lập hóa đơn

= **Lớp Khách Hàng**

cần thiết vào lớp khách hàng

Nhập mã sản phẩm đã mua

Tính Kế Thừa (tiếp theo)

- Tính Thừa kế
 - Là cơ chế cho phép một lớp chia sẻ những thuộc tính và những hành động đã định nghĩa trong một hoặc nhiều lớp khác
- Lớp con
 - Là lớp thừa kế từ lớp khác
- Lớp cha

- Là lớp từ đó một lớp khác thừa kế các ứng xử của nó
- Đa thừa kế
 - Khi một lớp con thừa kế từ hai hoặc nhiều lớp

NTTU-2022

Cú pháp của kế thừa trong java

- Sử dụng từ khóa *extends* để kế thừa.

```
class Subclass-name extends Superclass-name {  
    //methods and fields  
}
```

```
1  class Employee {  
2      float salary = 1000;  
3  }  
4  
5  class Programmer extends Employee {  
6      int bonus = 150;  
7  }  
8  
9  public class InheritanceSample1 {  
10     public static void main(String args[]) {  
11         Programmer p = new Programmer();  
12         System.out.println("Programmer salary is: " + p.salary);  
13         System.out.println("Bonus of Programmer is: " + p.bonus);  
14     }  
15 }
```

Kết quả:

```
Programmer salary is: 1000.0  
Bonus of Programmer is: 150
```

NTTU-2022

38

Các kiểu kế thừa trong java



VD đơn kế thừa

NTTU-2022

File:
TestInheritance1.java



Ví dụ về kế thừa nhiều cấp

- File: TestInheritance2.java



Ví dụ về kế thừa thứ bậc

- File: TestInheritance3.java



NTTU-2022

Câu hỏi: Tại sao đa kế thừa không được support trong java?

- Để giảm thiểu sự phức tạp và đơn giản hóa ngôn ngữ, đa kế thừa không được support trong java.
- Hãy suy xét kịch bản sau: Có 3 lớp A, B, C. Trong đó lớp C kế thừa từ các lớp A và B. Nếu các lớp A và B có phương thức giống nhau và bạn gọi nó từ đối tượng của lớp con, như vậy khó có thể xác định được việc gọi phương thức của lớp A hay B.
- Vì vậy lỗi khi biên dịch sẽ tốt hơn lỗi khi runtime, java sẽ print ra lỗi "compile time error" nếu bạn cố tình kế thừa 2 class.

VD



Tính Đa Hình

- Tính đa hình là thuộc tính cho phép một hành động ứng xử khác nhau trên các lớp khác nhau

Hành động
Display

Text Line Picture

How are you?

Đa hình (Polymorphism)

- ❖ Đa hình trong java (Polymorphism) là một khái niệm mà chúng ta có thể thực hiện một hành động bằng nhiều cách khác nhau. Polymorphism được cấu tạo từ 2 từ Hy Lạp: poly và morphs. Trong đó "poly" có nghĩa là nhiều và "morphs" có nghĩa là hình thể. Vậy polymorphism có nghĩa là nhiều hình thể.
- ❖ Có hai kiểu của đa hình trong java, đó là đa hình lúc biên dịch (compile) và đa hình lúc thực thi (runtime). Chúng ta có thể thực hiện đa hình trong java bằng cách nạp chồng phương thức và ghi đè phương thức.
- ❖ Nếu bạn nạp chồng phương thức static trong java, đó là một ví dụ về đa hình lúc biên dịch. Trong bài này, chúng ta tập trung vào đa hình lúc runtime trong java.

Đa hình lúc runtime trong java

Đa hình lúc runtime là quá trình gọi phương thức đã được ghi đè trong thời gian thực thi chương trình. Trong quá trình này, một phương thức được ghi đè được gọi thông qua biến tham chiếu của một lớp cha.

Trước khi tìm hiểu về đa hình tại runtime, chúng ta cùng tìm hiểu về

Upcasting. **Upcasting là gì?**

Khi biến tham chiếu của lớp cha tham chiếu tới đối tượng của lớp con, thì đó là Upcasting. Ví

dụ:



NTTU-2022

47

Ví dụ về đa hình lúc runtime trong java

- ❖ Chúng ta tạo hai lớp Bike và Splendar. Lớp Splendar kế thừa lớp Bike và ghi đè phương thức run() của nó. Chúng ta gọi phương thức run bởi biến tham chiếu của lớp cha. Khi nó tham chiếu tới đối tượng của lớp con và phương thức lớp con ghi đè phương thức của lớp cha, phương thức lớp con được gọi lúc runtime.
- ❖ Khi việc gọi phương thức được quyết định bởi JVM chứ không phải Compiler, vì thế đó là đa hình lúc

runtime.



NTTU-2022

48

Ví dụ 2: Bank

Giả sử Bank là một lớp cung cấp chức năng xem thông tin tỷ lệ lãi suất. Nhưng mỗi ngân hàng có một lãi xuất khác nhau, ví dụ các ngân hàng SBI, ICICI và AXIS có tỷ lệ lãi suất lần lượt là 8%, 7% và 9%.



NTTU-2022

49

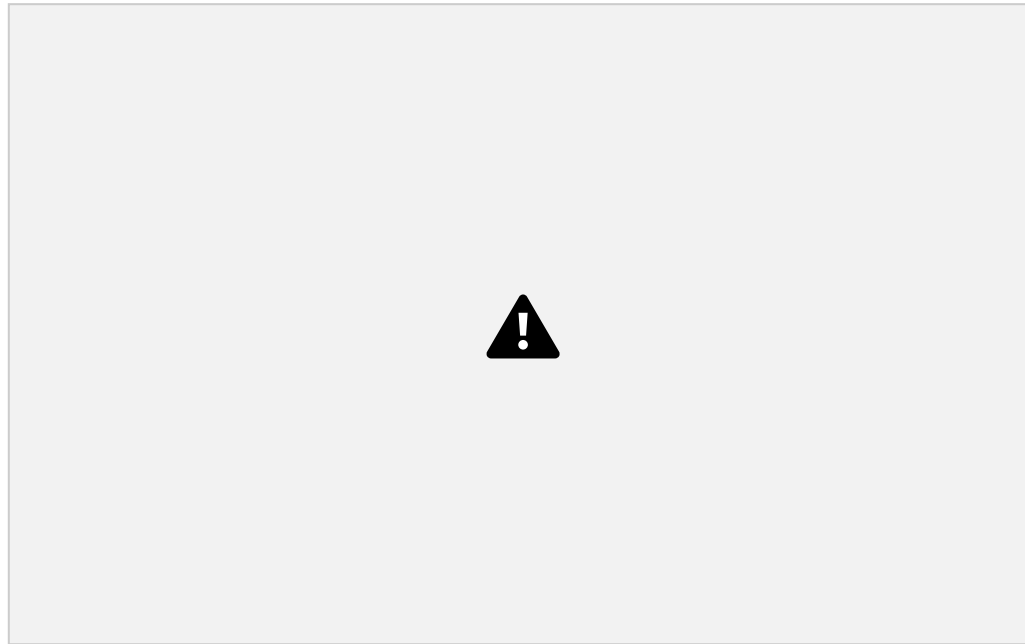
VD





Tính đóng gói trong java

Tính đóng gói trong java là kỹ thuật ẩn giấu thông tin không liên quan và hiện thị ra thông liên quan. Mục đích chính của đóng gói trong java là giảm thiểu mức độ phức tạp phát triển phần mềm.



Tính đóng gói trong java

Đóng gói cũng được sử dụng để bảo vệ trạng thái bên trong của một đối tượng. Bởi việc ẩn giấu các biến biểu diễn trạng thái của đối tượng. Việc chỉnh sửa đối tượng được thực hiện, xác nhận thông qua các phương thức. Hơn nữa, việc ẩn giấu các biến thì các lớp sẽ không chia sẻ thông tin với nhau được. Điều này làm giảm số lượng khớp nối có thể có trong một ứng dụng.

- ❖ Bạn có thể tạo lớp **read-only** hoặc **write-only** bằng việc cài đặt phương thức **setter** hoặc **getter**.
- ❖ Bạn có thể kiểm soát đối với dữ liệu. Giả sử bạn muốn đặt giá trị của id chỉ lớn hơn 100 bạn có thể viết logic bên trong lớp setter.

NTTU-2022

VD đóng gói

Hãy xem ví dụ sau về đóng gói trong java với một lớp chỉ có một trường và các phương thức setter và getter của nó.

File: Student.java



NTTU-2022

Các ưu điểm của OPP

- ✓ Chia sẻ trong phạm vi một ứng dụng

- ✓ Đẩy mạnh sự dùng lại của các đối tượng khi hiện thực những ứng dụng mới
- ✓ Về lâu dài, chi phí giảm đáng kể
- ✓ Giảm lỗi và rắc rối trong bảo trì
- ✓ Điều chỉnh nhanh hơn

NTTU-2022

TỔNG KẾT

- Tiếp cận hướng đối tượng đưa ra một giải pháp toàn diện cho một bài toán

cụ thể

- Trừu tượng dữ liệu là một tiến trình xác định và tập hợp các tính chất và các hành động có quan hệ với một thực thể cụ thể
- Lớp mô tả một thực thể, còn đối tượng là một thực thể thực tế
- Constructor và Destructor
- Tính lưu trữ, bao bọc dữ liệu, tính kế thừa và đa hình

NTTU-2022

THANK YOU VERY

MUCH



Q/A: khoihue@gmail.com