


TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI – BÀI GIẢNG ĐIỆN TỬ




LẬP TRÌNH JAVA


Bài 3:
LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn> © 2021 Hanoi University of Industry All rights reserved 1



LẬP TRÌNH JAVA




ÔN TẬP


- **Abstract Data Type**: Khái quát hóa một tập đối tượng thành một kiểu dữ liệu.
- **Object** = các thuộc tính mô tả (thành phần dữ liệu)+ các hành vi (thành phần hàm)
- **class**: ADT cho một tập các đối tượng tương tự nhau (cùng data, cùng functions).
- **Instance**: Một mô tả của đối tượng trong bộ nhớ của chương trình.

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn> © 2021 Hanoi University of Industry All rights reserved 2



LẬP TRÌNH JAVA



NỘI DUNG


1. Xây dựng lớp (giáo trình trang 85 – 100)
2. Giới thiệu một số lớp cơ bản trong java (163-178)
3. Kết tập, kế thừa và đa hình (giáo trình trang 102 – 124)
4. Lớp trừu tượng (giáo trình trang 124 – 131)
5. Interface-Giao diện (giáo trình trang 131 – 135)
6. Gói và quản lý gói (giáo trình trang 101)

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG


Website: <https://hau.edu.vn>

© 2021 Hanoi University of Industry All rights reserved

3



LẬP TRÌNH JAVA



1. XÂY DỰNG LỚP


- Mô hình hóa hướng đối tượng với ngôn ngữ UML
- Lớp và các đặc điểm

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG


Website: <https://hau.edu.vn>

© 2021 Hanoi University of Industry All rights reserved

4



LẬP TRÌNH JAVA




1. XÂY DỰNG LỚP

- Mô hình hóa hướng đối tượng với UML
 - Lớp: Hiện thực chung cho một tập các đối tượng có cùng đặc điểm và hành vi.
 - Lớp được ký hiệu là hình chữ nhật 3 ngăn:
 - Ngăn tên: danh từ/cụm danh từ. Tên đặt theo cú pháp chữ đầu mỗi từ viết hoa
 - Ngăn thuộc tính: danh từ/cụm danh từ. Quy cách đặt tên phương thức chữ cái mỗi từ viết hoa, trừ từ đầu tiên
 - Ngăn phương thức: động từ + cụm danh từ. Quy cách đặt tên giống thuộc tính.
 - Mỗi ngôn ngữ OOP quy định cú pháp riêng cho việc hiện thực lớp.


SinhVien
+maSV +tenSV +diaChi
+nhapThongTin() +xuatThongTin() +tinhDiemTB()

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn> © 2021 Hanoi University of Industry All rights reserved



LẬP TRÌNH JAVA



1. XÂY DỰNG LỚP

- Hiện thực lớp
 - Lớp như khuôn mẫu tạo ra đối tượng
 - Thuộc tính: thông tin mô tả đối tượng
 - Hành vi của lớp: các hành động mà đối tượng thực hiện nhiệm vụ của mình
 - Các bước hiện thực lớp
 - Khai báo mức cho phép sử dụng.
 - Khai báo lớp/lớp cha
 - Khai báo các thành phần thuộc tính.
 - Hiện thực thành mã nguồn các hành vi

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn> © 2021 Hanoi University of Industry All rights reserved



1. XÂY DỰNG LỚP

• Cú pháp khai báo lớp

```
[modifier] class class_name {
    [modifier] data_type field_name [=Value];
    [modifier] data_type field_name [=Value];
    [modifier] data_type method_name(parameters) {
        <code>
    }
}
```

• Modifier (phạm vi):

- public; protected; private.
- Không khai báo - mặc định

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn>

© 2021 Hanoi University of Industry All rights reserved

7



1. XÂY DỰNG LỚP



Phạm vi	private	Mặc định	protected	public
Cùng class	YES	YES	YES	YES
Cùng gói, khác class	NO	YES	YES	YES
Lớp con trong cùng gói với lớp cha	NO	YES	YES	YES
Khác gói, khác lớp	NO	NO	NO	YES
Lớp con khác gói với lớp cha	NO	NO	YES	YES

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn>

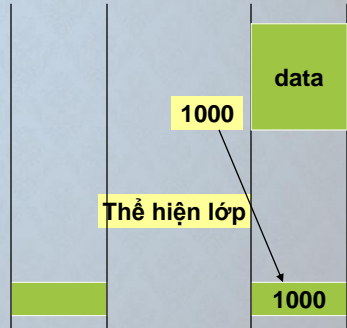
© 2021 Hanoi University of Industry All rights reserved

8


LẬP TRÌNH JAVA




1. XÂY DỰNG LỚP

- Sử dụng lớp:
 - Lớp được sử dụng thông qua đối tượng.
 - `Class_name object;`
 - `object= new class_name();`



`Class_name object= new class_name();`

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG
© 2021 Hanoi University of Industry All rights reserved
9


LẬP TRÌNH JAVA


1. XÂY DỰNG LỚP

- Bài toán minh họa sử dụng lớp
 - Đề bài: Hình tròn mô tả thông qua bán kính. Minh họa bài toán in thông hình tròn (bán kính, chu vi).
- Yêu cầu sinh viên:
 - Thực hiện cài đặt mỗi lớp trong 1 file.java riêng rẽ.
 - Thử nghiệm với các phạm vi khác nhau của thuộc tính và phương thức

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG
© 2021 Hanoi University of Industry All rights reserved
10



1. XÂY DỰNG LỚP- GỢI Ý THỰC HIỆN

```
public class VongTron {
    private int r;
    public void setR(int r) {
        this.r = r;
    }
    protected double tinhCV( ) {
        return Math.Pi*r*r;
    }
    void inHinhTron() {
        System.out.print("bán kính
        r="+ r+" chu vi="+tinhCV());
    }
}
```

```
class VongTronDemo {
    public static void
    main(String args[]){
        VongTron v=new
        VongTron ();
        v.setR(5);//r =5.
        v.inHinhTron();
    }
}
```

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn>

© 2021 Hanoi University of Industry All rights reserved

11



1. XÂY DỰNG LỚP

• Hàm tạo

- Hành vi được thực thi ngay lúc khởi tạo đối tượng, không có kiểu trả về, tên trùng tên lớp, thực hiện khởi gán các thuộc tính của đối tượng về 0/null
- Một lớp có thể có 0/1/n constructor.
- Mỗi ngôn ngữ hỗ trợ đều tự động sinh hành vi khởi tạo nếu người dùng không định nghĩa lại.
- Nếu đã có hàm tạo được định nghĩa trong lớp thì hàm mặc định không còn được mặc định gọi tới nữa.

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn>

© 2021 Hanoi University of Industry All rights reserved

12



1. XÂY DỰNG LỚP

• Cú pháp hàm tạo

```
[modifier] class_name ([parameters]){
    <code>
}
```

- Một lớp có thể có 1 hàm tạo hoặc nhiều hơn 1 hàm tạo khác nhau về tham số truyền vào.
- Nếu đã có hàm tạo được định nghĩa trong lớp thì hàm mặc định không còn được mặc định gọi tới nữa.

Vậy khi nào cần định nghĩa lại hành vi khởi tạo?

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn>

© 2021 Hanoi University of Industry All rights reserved

13



1. XÂY DỰNG LỚP - MINH HỌA HÀM KHỞI TẠO.

```

VongTron.java
1 package NhaMinh;
2 public class VongTron {
3     private double r;
4     public void setR(double r) {
5         this.r = r;
6     }
7     double tinhCV() {
8         return Math.PI*2*r;
9     }
10    protected void inHinhTron() {
11        System.out.println(
12            "ban kinh r=" + r +
13            " chu vi="+tinhCV());
14    }
15 }
16

Demo.java
1 package NhaMinh;
2 public class Demo {
3     public static void main(String[] args) {
4         VongTron v=new VongTron();
5         //
6         v.setR(7);
7         v.inHinhTron();
8     }
9 }
10

Output - Chuong2\THĐTJava-AccessModifier (run)
Run:
ban kinh r=0.0 chu vi=0.0
BUILD SUCCESSFUL (total time: 0 seconds)

```


Nếu là constructor tự tạo thì việc khởi tạo trị ban đầu được ẩn định bởi mã lệnh trong hàm tạo.

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG


Website: <https://hau.edu.vn>

© 2021 Hanoi University of Industry All rights reserved

14



LẬP TRÌNH JAVA



1. XÂY DỰNG LỚP - MINH HỌA HÀM KHỞI TẠO (TT)

```

VongTron.java
3
4 private double r;
5
6 public VongTron(int r) {
7     this.r=r;
8 }
9
10 public void setR(double r) {
11     this.r = r;
12 }
13
14 double tinhCV()
15 {
16     return Math.PI*2*r;
17 }
18
19 protected void inHinhTron(){
20     System.out.println(
21         "ban kinh r=" + r +
22         " chu vi="+tinhCV());
23 }
24

```

```


Demo.java
1
2 package NhaMinh;
3 public class Demo {
4     public static void main(String[] args)
5     {
6         VongTron v=new VongTron();
7         //
8         v.setR(7);
9         v.inHinhTron();
10    }

```


Nếu một lớp có constructor thì constructor mặc định của ngôn ngữ không được sử dụng nữa.

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn> © 2021 Hanoi University of Industry All rights reserved 15



LẬP TRÌNH JAVA



1. XÂY DỰNG LỚP - MINH HỌA HÀM KHỞI TẠO (TT)

```

VongTron.java
1 package NhaMinh;
2 public class VongTron {
3     private double r;
4
5     public VongTron() {
6         this.r=0;
7     }
8
9     public VongTron(int r) {
10        this.r=r;
11    }
12
13    public VongTron(VongTron v) {
14        this.r=v.r;
15    }
16

```

```

Demo.java
1
2 package NhaMinh;
3 public class Demo {
4     public static void main(String[] args)
5     {
6         VongTron v=new VongTron();
7         v.inHinhTron();
8         VongTron v1=new VongTron(5);
9         v1.inHinhTron();
10        VongTron v2=new VongTron(v1);
11        v2.inHinhTron();
12        v.inHinhTron();
13    }

```


```

run:
ban kinh r=0.0 chu vi=0.0
ban kinh r=5.0 chu vi=31.41592653589793
ban kinh r=5.0 chu vi=31.41592653589793
ban kinh r=0.0 chu vi=0.0
BUILD SUCCESSFUL (total time: 0 seconds)


```

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn> © 2021 Hanoi University of Industry All rights reserved 16



LẬP TRÌNH JAVA



1. XÂY DỰNG LỚP

- Hành vi hủy (destructor)
 - Đối tượng tồn tại từ khi được khởi tạo cho đến hết khối chứa nó.
 - Khi ra khỏi khối chứa *đối tượng đã không còn giá trị sử dụng*.

```
{ .....
.....
ClassName obj = new ClassName();
.....
.....
.....
}
```

Góc nhìn thời gian:
Thời gian sống của đối tượng


Góc nhìn nội dung văn bản chương trình:
phạm vi (scope) của đối tượng

Java chỉ dùng cách cấp phát đối tượng động, có trình gom rác riêng và không hỗ trợ hành vi hủy.


Website: <https://hau.edu.vn>

© 2021 Hanoi University of Industry All rights reserved

17



LẬP TRÌNH JAVA



1. XÂY DỰNG LỚP – NẠP CHỒNG PHƯƠNG THỨC

- Kỹ thuật nạp chồng (Overloading):
 - Kỹ thuật cho phép xây dựng các hành vi trùng tên nhưng khác chữ ký (signature) trong cùng một lớp.
 - Ý nghĩa nạp chồng là để đa dạng hóa các xử lý của đối tượng.
 - Chữ ký bao gồm:
 - Số lượng tham số.
 - Thứ tự các kiểu của các tham số
 - Giữ nguyên kiểu kết quả trả về

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

© 2021 Hanoi University of Industry All rights reserved

18



1. XÂY DỰNG LỚP - SETTER/GETTER

- Các phương thức giao tiếp với các lớp khác

- Đặt giá trị cho thuộc tính – getter

```
public void setXYZ( datatype xyz){
    this.xyz=xyz;
}
```

- Lấy giá trị các thuộc tính

```
public data_type getXYZ(){
    return xyz;
}
```

Trong đó xyz là tên thuộc tính.





1. XÂY DỰNG LỚP – TOSTRING()

- Phương thức hiển thị trạng thái đối tượng kế thừa từ lớp cha Object nên được viết lại.

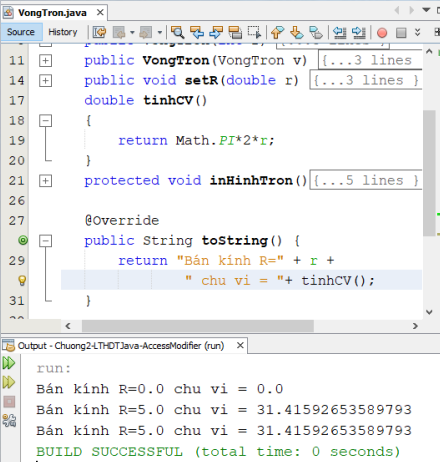
```
public String toString(){
    return chuỗi cần hiển thị;
}
```

- Ví dụ minh họa:

```
public String toString() {
    return "Bán kính R=" + r + " chu vi = "+ tinhCV();
}
```


LẬP TRÌNH JAVA


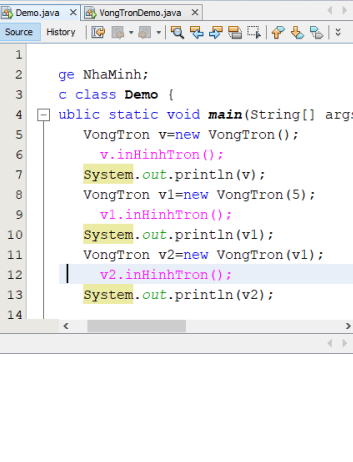
1. XÂY DỰNG LỚP – TOSTRING() - TT



```

11 public VongTron(VongTron v) {...3 lines}
14 public void setR(double r) {...3 lines}
17 double tinhCV()
18 {
19     return Math.PI*2*r;
20 }
21 protected void inHinhTron() {...5 lines}
26
27 @Override
28 public String toString() {
29     return "Bán kính R=" + r +
30         " chu vi = " + tinhCV();
31 }

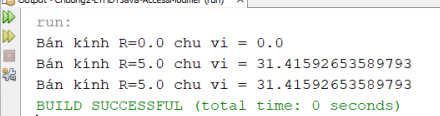
```



```

1 ge NhaMinh;
2
3 c class Demo {
4     public static void main(String[] args)
5     {
6         VongTron v=new VongTron();
7         v.inHinhTron();
8         System.out.println(v);
9         VongTron v1=new VongTron(5);
10        v1.inHinhTron();
11        System.out.println(v1);
12        VongTron v2=new VongTron(v1);
13        v2.inHinhTron();
14        System.out.println(v2);
15    }
16 }

```





```

run:
Bán kính R=0.0 chu vi = 0.0
Bán kính R=5.0 chu vi = 31.41592653589793
Bán kính R=5.0 chu vi = 31.41592653589793
BUILD SUCCESSFUL (total time: 0 seconds)

```

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Webiste: <https://haui.edu.vn> © 2021 Hanoi University of Industry All rights reserved 21


LẬP TRÌNH JAVA


1. XÂY DỰNG LỚP – THÀNH PHẦN STATIC

- Thành phần static.
 - Thành phần đại diện chung cho tất cả các đối tượng.
 - Từ khóa biểu diễn **static**.
 - Dữ liệu static là **DỮ LIỆU TOÀN CỤC**.
 - Hành vi static là **HÀNH VI TOÀN CỤC**
 - Đoạn mã static là **đoạn mã TOÀN CỤC**. Các lệnh trong đoạn mã chỉ thao tác lên dữ liệu toàn cục (static.)
- Truy xuất thuộc tính và phương thức qua **tên lớp** hoặc **đối tượng** thuộc lớp.
 - `ClassName.StaticMember`
 - `ClassName obj= new ClassName();`
`obj. StaticMember`

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Webiste: <https://haui.edu.vn> © 2021 Hanoi University of Industry All rights reserved 22



1. XÂY DỰNG LỚP – THÀNH PHẦN STATIC (TT)

- Thuộc tính static
 - Bài toán minh họa sách giáo trình mục 3.6.1.
 - Thuộc tính nextID dùng để tạo mã nhân viên tăng tự động
 - Thuộc tính tĩnh không bị hàm tạo khởi tạo lại giá trị ban đầu
 - Thuộc tính tĩnh là chung cho tất cả các đối tượng.

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn>

© 2021 Hanoi University of Industry All rights reserved

23



1. XÂY DỰNG LỚP – ĐOẠN MÃ STATIC

- Đoạn mã static

Cú pháp đưa một đoạn code static:

```
static
{ <code>
}
```

CodeStaticDemo.java

```
1 // static keyword demo
2 class StaticCode
3 { static int m;      // so luong nhan vien
4   static long n;
5   static char c;
6   static
7   { m=5;
8     n=100000;
9     c='A' ;
10  }
11 public static void main (String [] args)
12 { System.out.println(" " + StaticCode.m + " , " + StaticCode.n + " , " + StaticCode.c);
13 }
14 }
```



Code static được thực thi ngay khi lớp có mã nguồn được tham khảo tới.

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn>

© 2021 Hanoi University of Industry All rights reserved

24


LẬP TRÌNH JAVA


1. XÂY DỰNG LỚP - ĐOẠN MÃ STATIC (TT)

```

1 // static keyword demo
2 class StaticCode2
3 {
4     int m;
5     static long n;
6     char c;
7     static
8     {
9         m=5;
10        n=1000000;
11        c='A';
12    }
13    void M()
14    { System.out.println("Hello"); }
15    public static void main (String [] args)
16    { System.out.println(StaticCode2.n); }
17 }



```

Code static chỉ tham
Khảo dữ liệu static

Task List

	description
✖	non-static variable m cannot be referenced from a static context
✖	non-static variable c cannot be referenced from a static context

Website: <https://hau.edu.vn> © 2021 Hanoi University of Industry All rights reserved 25


LẬP TRÌNH JAVA


1. XÂY DỰNG LỚP

- Nếu có nhiều đoạn lệnh static thì thứ tự thực hiện tuần tự

```

1 // static keyword demo
2 class StaticCode
3 {
4     static int m; // so luong nhan vien
5     static long n;
6     static char c;
7     static
8     {
9         m=5;
10        n=1000000;
11        c='A';
12    }
13    void M()
14    { System.out.println("Hello"); }
15    static
16    {
17        m=15;
18        n=11000000;
19        c='B';
20    }
21    public static void main (String [] args)
22    { System.out.println(" " + StaticCode.m + ", " + StaticCode.n + ", " + StaticCode.c); }
23 }

```

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG Website: <https://hau.edu.vn> © 2021 Hanoi University of Industry All rights reserved 26



1. XÂY DỰNG LỚP – THÀNH PHẦN FINAL

- Thành phần final
 - Đó là chỉ thị được thiết lập **không** cho phép sửa dữ liệu của thuộc tính.
 - Hoặc **không** cho phép ghi đè/nạp chồng phương thức
 - Hoặc không cho phép lớp khác kế thừa.
- Từ khóa **final**
 - **Final class**: lớp không cho kế thừa
 - **Final thuộc tính**: thuộc tính hằng
 - **Final phương thức**: phương thức không được ghi đè/nạp chồng

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn>

© 2021 Hanoi University of Industry All rights reserved

27



1. XÂY DỰNG LỚP – THÀNH PHẦN FINAL (TT)

FinalDemo.java

```

1 // Final keyword Demo
2 class A
3 {
4     final int field1=2;
5     int field2;
6     A( int f1, int f2)
7     {
8         field1= f1;
9         field2= f2;
10    }
11    void outData()
12    { System.out.println(" " + field1 + " , " + field2); }
13    public static void main(String args[])
14    {
15        final int MaxN = 100;
16        A obj= new A(5,10);
17        MaxN=500;
18        obj.outData();
19        System.out.println(MaxN);
20    }
21 }
  
```


Task List		
	description	resource
	cannot assign a value to final variable field1	FinalDemo.java
	cannot assign a value to final variable MaxN	FinalDemo.java

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG


Website: <https://hau.edu.vn>

© 2021 Hanoi University of Industry All rights reserved

28



LẬP TRÌNH JAVA



1. XÂY DỰNG LỚP – THÀNH PHẦN FINAL (TT)

FinalDemo2.java

```


1 // Final keyword Demo
2 final class A
3 {
4     int field1=2;
5     int field2;
6     A( int f1, int f2)
7     {
8         field1= f1;
9         field2= f2;
10    }
11    void outData()
12    {
13        System.out.println(" " + field1 + ", " + field2);
14    }
15 }
16 class B extends A
17 {
18     int fb=5;
19     B( int f1, int f2, int f3)
20     {
21         super (f1,f2);
22         fb=f3;
23     }
24     public static void main(String args[])
25     { }
26 }

```


Task List		
	description	resource
❌	cannot inherit from final A	FinalDemo2.java

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn> © 2021 Hanoi University of Industry All rights reserved 29



LẬP TRÌNH JAVA



1. XÂY DỰNG LỚP – THÀNH PHẦN FINAL (TT)

FinalDemo3.java

```


1 // Final keyword Demo
2 class A
3 {
4     int field1=2;
5     int field2;
6     A( int f1, int f2)
7     {
8         field1= f1;
9         field2= f2;
10    }
11    final void outData()
12    {
13        System.out.println(" " + field1 + ", " + field2);
14    }
15 }
16 class B extends A
17 {
18     int fb=5;
19     B( int f1, int f2, int f3)
20     {
21         super (f1,f2);
22         fb=f3;
23     }
24     void outData()
25     {
26    }
27     public static void main(String args[])
28     { }
29 }

```


Task List		
	description	resource
❌	outData() in B cannot override outData() in A; overridden method is final	FinalDemo3.java

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn> © 2021 Hanoi University of Industry All rights reserved 30



LẬP TRÌNH JAVA




2. GIỚI THIỆU MỘT SỐ LỚP CƠ BẢN TRONG JAVA


- Một số lớp cơ bản được xây dựng sẵn trong Java.
- Chi tiết các lớp tham khảo giáo trình tại tiểu mục 5.1 (giáo trình trang 163 đến 178)
 - Lớp Object
 - Lớp String
 - Lớp Math
 - ...

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn> © 2021 Hanoi University of Industry All rights reserved 31



LẬP TRÌNH JAVA



2. GIỚI THIỆU MỘT SỐ LỚP CƠ BẢN TRONG JAVA

- Lớp Object
 - Là lớp cha của mọi lớp trong java (trực tiếp/gián tiếp).
 - Lớp được đặt trong gói *java.lang* (*java.lang.Object*)
 - Lớp đã định nghĩa các trạng thái cơ bản và các phương thức cơ bản của mọi lớp phải có như:
 - So sánh nó với 1 đối tượng khác- `equals()`,
 - Biểu diễn thông tin đối tượng – `toString()`,
 - Đợi điều kiện – `wait()`,
 - Nhận biết các đối tượng khác khi biến điều kiện có thay đổi- `notify()`,
 - Lấy Class – `getClass()`,
 - Hành vi `hashCode()` cho biết mã `hashCode` của *đối tượng*.

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn> © 2021 Hanoi University of Industry All rights reserved 32



2. GIỚI THIỆU MỘT SỐ LỚP CƠ BẢN TRONG JAVA (TT)

• Lớp String

- Là lớp dùng để thao tác với dữ liệu kiểu chuỗi
- Định nghĩa 1 chuỗi theo một trong 2 cách
 - String variable_name=[value];
 - Hoặc String variable_name= new String(value);
- Nối chuỗi
 - String Str3= Str1 + Str2; // Str3="HelloHi"
 - String Str4 = Str3 + 1; // Str4= "HelloHi1"

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn>

© 2021 Hanoi University of Industry All rights reserved

33



2. GIỚI THIỆU MỘT SỐ LỚP CƠ BẢN TRONG JAVA (TT)

• Lớp String

- charAt(int): char: trả lại ký tự tại vị trí được chỉ ra
- compareTo(String):int: so sánh 2 chuỗi
- compareTo(Object):int: so sánh hai chuỗi theo đối tượng
- compareToIgnoreCase(String): int: so sánh không phân biệt chữ hoa
- concat(String): String: nối chuỗi
- equal(Object): Boolean: so sánh hai chuỗi bằng nhau

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn>

© 2021 Hanoi University of Industry All rights reserved

34



LẬP TRÌNH JAVA



2. GIỚI THIỆU MỘT SỐ LỚP CƠ BẢN TRONG JAVA (TT)

- Wrapper
 - Là các lớp được đặt trong gói java.lang.
 - Các lớp này bao lấy các kiểu dữ liệu cơ bản nhằm tạo ra tính OOP cho các kiểu cơ bản.
 - Các hành vi của lớp Wrapper đa số là các hành vi static hiện thực các thao tác lên dữ liệu thuộc kiểu tương ứng.
 - Biến thuộc lớp Wrapper chứa địa chỉ của dữ liệu.

1100	32
1000	15
Integer obj2	1100
Integer obj1	1000
int x	5

```
int x=5;
Integer obj1= new Integer(15);
Integer obj2= new Integer("32");
```

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn> © 2021 Hanoi University of Industry All rights reserved 35



LẬP TRÌNH JAVA



2. GIỚI THIỆU MỘT SỐ LỚP CƠ BẢN TRONG JAVA (TT)

- Wrapper

Data type	Wrapper class
boolean	Boolean
byte	Byte
char	Character
double	Double
float	Float
int	Integer
long	Long
short	Short

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn> © 2021 Hanoi University of Industry All rights reserved 36



2. GIỚI THIỆU MỘT SỐ LỚP CƠ BẢN TRONG JAVA (TT)

- Lớp Math
 - Trong gói java.lang. Lớp này chứa các hàm xử lý toán học.
 - Là lớp hằng(final) – lớp này không có phép lớp khác kế thừa.
 - Một số phương thức phổ biến trong lớp:
 - PI()
 - Sqrt(double): double
 - pow(double): double
 - Sin(double):double
 - vvvv

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn>

© 2021 Hanoi University of Industry All rights reserved

37



TỔNG KẾT XÂY DỰNG LỚP


1. Khi thiết kế 1 lớp: Cần xem xét để chọn phạm vi truy cập phù hợp.
2. Sử dụng các chỉ thị static, final cho các thành phần đặc biệt.
3. Lớp được sử dụng thông qua đối tượng
4. Một số kỹ thuật xây dựng lớp: ghi đè, đặt – lấy giá trị thuộc tính, kết xuất đối tượng, vv

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG


Website: <https://hau.edu.vn>

© 2021 Hanoi University of Industry All rights reserved

38



LẬP TRÌNH JAVA




NỘI DUNG


1. Xây dựng lớp (giáo trình trang 85 – 100)
2. Giới thiệu một số lớp cơ bản trong java (
3. **Kết tập, kế thừa và đa hình (giáo trình trang 102 – 124)**
4. Lớp trừu tượng (giáo trình trang 124 – 131)
5. Interface-Giao diện (giáo trình trang 131 – 135)
6. Gói và quản lý gói (giáo trình trang 101)

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn> © 2021 Hanoi University of Industry All rights reserved 39

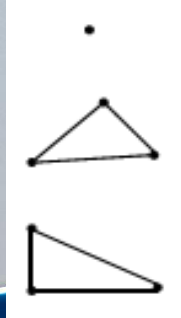


LẬP TRÌNH JAVA



3. KẾT TẬP, KẾ THỪA VÀ ĐA HÌNH

- Giới thiệu kế thừa và đa hình
 - Khả năng một lớp thừa hưởng dữ liệu và mã lệnh từ một hay nhiều lớp khác.
 - Lớp mới kế thừa những thành viên đã có trong lớp cũ.
 - Lớp cũ: lớp cha hay lớp cơ sở
 - Lớp mới: lớp con hay lớp dẫn xuất
 - Ví dụ
 - Lớp cũ: lớp điểm
 - Kết tập: Tam giác có 3 điểm
 - Kế thừa: Tam giác vuông



BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

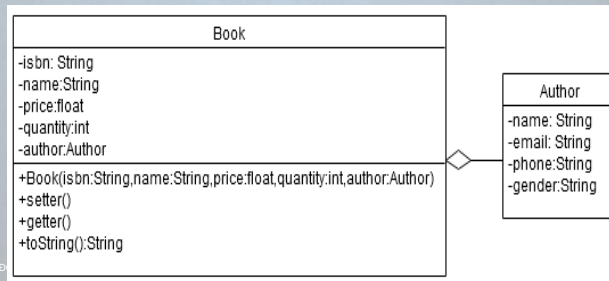
Website: <https://hau.edu.vn> © 2021 Hanoi University of Industry All rights reserved 40



3. KẾT TẬP, KẾ THỪA VÀ ĐA HÌNH

• Kết tập

- Là một trong hai cách sử dụng lại mã
- Kết tập cho phép khai báo thuộc tính của lớp là kiểu đối tượng hay kiểu tham chiếu



BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn>

© 2021 Hanoi University of Industry All rights reserved

41



3. KẾT TẬP, KẾ THỪA VÀ ĐA HÌNH

• Kế thừa

- Là kỹ thuật sử dụng các lớp chuyên biệt hóa hoặc tổng quát hóa
- Chuyên biệt hóa: cho phép xem xét các tính chất của một số các đối tượng giống nhau và xác định một vài tính chất chỉ có trên một số ít các đối tượng.
- Tổng quát hóa: xem xét tập các đối tượng và tìm ra đặc điểm chung giữa chúng.
- Kết quả của 2 quá trình này là các thuộc tính và phương thức của lớp không bị lặp đi lặp lại mà chia sẻ giữa các lớp liên quan với nhau.

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn>

© 2021 Hanoi University of Industry All rights reserved

42



3. KẾT TẬP, KẾ THỪA VÀ ĐA HÌNH

- Kế thừa có 2 loại: kế thừa đơn và kế thừa bội
 - Khai báo cú pháp xây dựng lớp
 - Khởi tạo trong kế thừa
 - Các chỉ thị truy cập
 - Đa hình trong kế thừa
 - Che dấu thuộc tính
 - Ghi đè phương thức
 - Java chỉ hỗ trợ đơn kế thừa

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn>

© 2021 Hanoi University of Industry All rights reserved

43



3. KẾT TẬP, KẾ THỪA VÀ ĐA HÌNH

- Khai báo kế thừa đơn


```
class Base {
    //Định nghĩa các thuộc tính
    //Định nghĩa các phương thức
}
```
- class Driver extends Base {


```
//Các thuộc tính bổ sung cho lớp con
//Các phương thức bổ sung cho lớp con hoặc viết lại của lớp cha
}
```
- Java chỉ hỗ trợ đơn kế thừa đầy đủ theo cú pháp trên. Chi tiết sách giáo trình tại tiểu mục 3.8.2

Lớp con là mở rộng của lớp cha

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn>

© 2021 Hanoi University of Industry All rights reserved

44



3. KẾT TẬP, KẾ THỪA VÀ ĐA HÌNH (TT)

- Truy xuất vào các thành phần của lớp cha
 - Từ khóa `super` được sử dụng để phân biệt các tình huống sau:
 - Phân biệt thành phần của lớp cha và lớp con nếu chúng trùng tên.
 - Gọi phương thức tạo của lớp cha trong phương thức tạo của lớp con.
 - Cú pháp lệnh
 - `super.field_name;`
 - `super.method_name`
 - `super();` // gọi phương thức khởi tạo không đối của lớp cha
 - `super(value);` // gọi phương thức khởi tạo có đối của lớp cha

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn>

© 2021 Hanoi University of Industry All rights reserved

45



3. KẾT TẬP, KẾ THỪA VÀ ĐA HÌNH (TT)


- Truy xuất vào các thành phần của lớp cha
 - Lớp con không kế thừa phương thức khởi tạo của lớp cha; lớp cha luôn được **khởi tạo trước lớp con**.
 - Các phương thức khởi tạo của lớp con luôn gọi phương thức khởi tạo của lớp cha theo cơ chế:
 - Tự động gọi: nếu lớp cha có phương thức khởi tạo mặc định.
 - Gọi trực tiếp (tường minh): nếu lớp cha có phương thức khởi tạo tự định nghĩa theo cú pháp:
 - Cú pháp
 - `super(values);`
 - Hoặc `super();`

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG


Website: <https://hau.edu.vn>

© 2021 Hanoi University of Industry All rights reserved

46



LẬP TRÌNH JAVA



VÍ DỤ MINH HỌA

```

class Person{
    public String ma,ten;
    public int tuoi;
    Person (String ma,String name, int tuoi)
    {this.ma=ma; ten = name; this.tuoi=tuoi;}
    public String toString() { return "ma="+ma+"\tten="+ten + "\t tuoi=" + tuoi;}
}

class Employee extends Person
{
    public float luong;
    public Employee(String ma,String ten, int tuoi, float luong)
    {super(ma,ten, tuoi);    this.luong=luong; }
    public String toString() {return super.toString() + "\tluong="+luong;}
}

class Manager extends Employee
{
    public float phuCap;
    public Manager(String ma,String ten, int tuoi, float luong,float phuCap)
    { super(ma,ten, tuoi,luong);    this.phuCap=phuCap; }
    public String toString(){return super.toString() + "\tphu cap="+phuCap;}
}

class EmployeeDemo2 {
    public static void main(String args[])
    {
        Person p=new Person("P01", "Hoa",21);
        Employee e=new Employee("P02", "Luân",21,300f);
        Manager m=new Manager("P03","Quang", 25, 150f,200f );
        System.out.println(p);    System.out.println(e);
        System.out.println(m);
    }
}

```


200
150
25
Quang
P03

150
25
Luan
P02


21
Hoa
P01

mng	300
emp	200
p	100

Website: <https://hau.edu.vn> © 2021 Hanoi University of Industry All rights reserved 47



LẬP TRÌNH JAVA



3. KẾT TẬP, KẾ THỪA VÀ ĐA HÌNH (TT)

- Phạm vi truy cập trong kế thừa:
 - Phạm vi của lớp:
 - Là khả năng đặt phạm vi nhìn thấy của đối tượng thuộc lớp từ bên ngoài.
 - **public**: cho phép lớp con kế thừa nằm ở bất kỳ đâu
 - **Không chỉ định**: chỉ cho phép lớp con kế thừa trong cùng gói
 - Phạm vi của thành phần trong lớp:
 - **public, protected**: cho phép lớp con ở bất kỳ đâu được kế thừa thuộc tính/phương thức.
 - **Không chỉ định**: Chỉ cho phép lớp con cùng gói kế thừa và sử dụng
 - **Private**: chỉ cho phép được sử dụng trong phạm vi lớp hiện tại

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG Website: <https://hau.edu.vn> © 2021 Hanoi University of Industry All rights reserved 48



3. KẾT TẬP, KẾ THỪA VÀ ĐA HÌNH (TT)

- Đa hình trong kế thừa - Che dấu thuộc tính
 - Thuộc tính trong lớp con có tên giống lớp cha thì trong nội dung của lớp con thuộc tính của lớp cha bị che đi.
 - Để truy cập tới thuộc tính thuộc lớp cha dùng từ khóa super
 - Để phân biệt ở lớp con dùng từ khóa this.

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn>

© 2021 Hanoi University of Industry All rights reserved

49



3. KẾT TẬP, KẾ THỪA VÀ ĐA HÌNH (TT)

- Minh họa che dấu biến

```
public class Parents{
    int intData;
    float floatData;
}
public class Children extends Parent {
    int intData;// che thuộc tính intData ở lớp cha
    public void overrideTest(){
        intData=1; //thuộc tính lớp con
        super.intData=-1; //truy cập đến thuộc tính cha
    }
}
```

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn>

© 2021 Hanoi University of Industry All rights reserved

50



3. KẾT TẬP, KẾ THỪA VÀ ĐA HÌNH (TT)

- Đa hình trong kế thừa - Ghi đè (overriding).
 - Đây là kỹ thuật cho phép thay thế mã lệnh trong hành vi thừa kế từ lớp cha để lớp con phản ứng khác với yêu cầu của bài toán khi cùng được yêu cầu thực thi hành vi này.
 - Phân biệt ghi đè với nạp chồng. Nạp chồng là kỹ thuật cho phép nhiều hành vi trùng tên nhưng khác chữ ký trong **cùng một lớp**.

Minh họa slide tiếp sau

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn>

© 2021 Hanoi University of Industry All rights reserved

51



THÍ DỤ VỀ OVERLOAD VÀ OVERRIDE

- Phương thức Add(..) được nạp chồng trong lớp MyArray1
- Phương thức Add(...) được ghi đè trong lớp MyArray2

OverLoad_Override.java



```
1 // OverLoad_Override.java
2 class MyArray1 // calculating area
3 {
4     int MaxN, a[], n;
5     MyArray1 ( int [] ar)
6     {
7         n=ar.length;
8         MaxN= 2* n; // bo nho co the chua gap 2 lan mang ban dau
9         a= new int [MaxN];
10        for (int i=0; i<n; i++) a[i]=ar[i];
11    }
12    void Output()
13    {
14        for (int i=0; i<n; i++)
15            System.out.print(i<n-1? a[i] + " , " : a[i] + "\n");
16    }
17    void Add (int x) // add to the last of array
18    {
19        if (n==MaxN) System.out.println("Array is full!");
20        else
21        {
22            a[n]=x; // add to the last
23            n++;
24        }
25    }
26    void Add (int x, int i)
27    {
28        if (n==MaxN) System.out.println("Array is full!");
29        else
30        {
31            for (int j=n; j>i; j--) a[j]=a[j-1];
32            a[i]=x;
33            n++;
34        }
35    }
36 }
37 class MyArray2 extends MyArray1
38 {
39     super (ar);
40 }
41 void Add(int x) // add to the first of array
42 {
43     if (n==MaxN) System.out.println("Array is full!");
44     else
45     {
46         for (int i=n; i>0; i--) a[i]=a[i-1];
47         a[0]=x;
48         n++;
49     }
50 }
```

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn>

© 2021 Hanoi University of Industry All rights reserved

52


LẬP TRÌNH JAVA


3. KẾT TẬP, KẾ THỪA VÀ ĐA HÌNH (TT)

- Minh họa xử lý trong main khi gọi các phương thức trùng tên

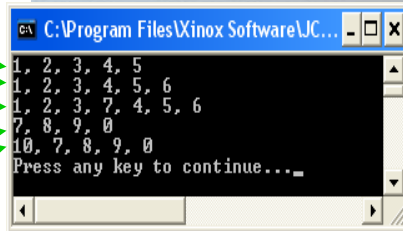
```

43 class Problem
44 { public static void main(String args[])
45 { int a1[] = { 1,2,3,4,5 };
46   MyArray1 obj1= new MyArray1(a1);
47   obj1.Output();
48   obj1.Add(6);
49   obj1.Output();
50   obj1.Add(7,3);
51   obj1.Output();
52   int a2[] = { 7, 8, 9, 0 };
53   MyArray2 obj2 = new MyArray2(a2);
54   obj2.Output();
55   obj2.Add(10);
56   obj2.Output();
57 }
58 }
59 }

```



thêm đuôi →

thêm đầu →



BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Webiste: <https://hau.edu.vn> © 2021 Hanoi University of Industry All rights reserved 53


LẬP TRÌNH JAVA


3. KẾT TẬP, KẾ THỪA VÀ ĐA HÌNH (TT)

```

1 // LopCon_1.java
2 // Bien doi tuong la lop cha nhung cu the lai la lop con
3 class Father
4 { void Out()
5 { System.out.println("I am a father!");
6 }
7 }
8 class Son extends Father
9 { void Out()
10 { System.out.println("I am a son!");
11 }
12 }
13 class Demo
14 { public static void main (String args[])
15 { Son aSon = new Son();
16   aSon.Out();
17   Father aFather = new Father();
18   aFather.Out();
19   Father obj= new Son();
20   obj.Out();
21 }
22 }

```

Tính đa hình

Biến đối tượng khi khai báo có thể khai báo biến là lớp cha nhưng khởi tạo biến là đối tượng thuộc lớp con.

Webiste: <https://hau.edu.vn> © 2021 Hanoi University of Industry All rights reserved 54



3. KẾT TẬP, KẾ THỪA VÀ ĐA HÌNH (TT)

- Thành phần final
 - Phương thức
 - không bị ghi đè khi kế thừa
 - mục đích để đảm bảo tính đúng đắn: việc ghi đè phương thức có thể gây sai lệch về ý nghĩa
 - Thuộc tính
 - Không cho che dấu và thay đổi giá trị

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn>

© 2021 Hanoi University of Industry All rights reserved

55



3. KẾT TẬP, KẾ THỪA VÀ ĐA HÌNH (TT)

```
public class Ngươi {
    protected String ten, hocVi;
    protected int tuoi;
    public final void hienThi() {
        System.out.println("ten: " + ten + " tuoi: "
            + tuoi + "hoc vi: " + hocVi);
    }
}
```

Nếu lớp con có phương thức hiển thị thông tin thì không được phép sửa đổi phương thức hiển thị lớp cha

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn>

© 2021 Hanoi University of Industry All rights reserved

56



3. KẾT TẬP, KẾ THỪA VÀ ĐA HÌNH (TT)

- Dữ liệu chung không thay đổi- static final

```
public class DuLieuChung {
    public static final double Pi=3.14;
    public static double soThuc(String st){
        double a;
        a=Double.valueOf(st).doubleValue();
        return a;
    }
}
```

```
public class HinhTronDemo {
    public static void main(String[] args) {
        double dt,r;
        Scanner s=new Scanner(System.in);
        System.out.print("nhap r=");
        r=DuLieuChung.soThuc(s.nextLine());
        DuLieuChung.Pi=0; // báo lỗi ở đây
        dt=DuLieuChung.Pi*r*r;
        System.out.println("dien tích hình tron =" +dt);
    }
}
```

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn>

TỔNG KẾT KẾ THỪA- ĐA HÌNH- KẾT TẬP


1. Java cho phép định nghĩa lớp kế thừa thông qua từ khóa extends .
2. Ghi đè là kỹ thuật cho phép viết lại code của hành vi mà lớp con thừa kế từ lớp cha để tạo ra sự khác biệt.
3. Đa hình trong OOP có được là nhờ kỹ thuật Overriding.

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG


Website: <https://hau.edu.vn>

© 2021 Hanoi University of Industry All rights reserved

58



LẬP TRÌNH JAVA




NỘI DUNG


1. Xây dựng lớp (giáo trình trang 85 – 100)
2. Giới thiệu một số lớp cơ bản trong java (
3. Kết tập, kế thừa và đa hình (giáo trình trang 102 – 124)
4. Lớp trừu tượng (giáo trình trang 124 – 131)
5. Interface-Giao diện (giáo trình trang 131 – 135)
6. Gói và quản lý gói (giáo trình trang 101)

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn> © 2021 Hanoi University of Industry All rights reserved 59



LẬP TRÌNH JAVA



4. LỚP TRỪU TƯỢNG

- Khái niệm lớp trừu tượng
- Đặc điểm lớp trừu tượng

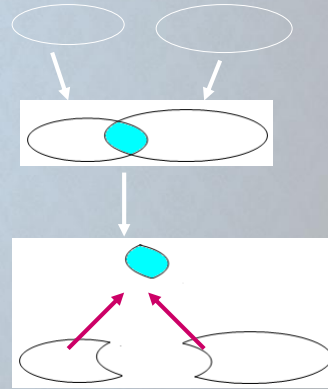
BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn> © 2021 Hanoi University of Industry All rights reserved 60



4. LỚP TRỪU TƯỢNG

- Kỹ thuật phân cấp thừa kế
 - Liệt kê đặc điểm của các đối tượng cần quan tâm.
 - Tìm tập giao của các tính chất giữa các lớp, tách tập giao này để xây dựng lớp cha.
 - Đặt 1 tên gọi có ý nghĩa cho lớp cha.
 - Phần còn lại sau khi tách tập giao là các lớp con.



BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn>

© 2021 Hanoi University of Industry All rights reserved

61



4. LỚP TRỪU TƯỢNG

- Khái niệm lớp trừu tượng
 - Là kết quả của việc khái quát hóa *cao “đến nỗi chưa biết viết code thế nào” mà chỉ biết tên việc cần thực hiện..*
 - Lớp trừu tượng là lớp chưa đầy đủ thường được dùng làm lớp cha, lớp con kế thừa sẽ hoàn thiện nốt phần còn lại.
 - Lớp trừu tượng không tạo được đối tượng, thường dùng để định nghĩa các “khái niệm” đóng vai trò lớp cơ sở cho các lớp “cụ thể” khác.

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn>

© 2021 Hanoi University of Industry All rights reserved

62



4. LỚP TRỪU TƯỢNG

- Vai trò của lớp trừu tượng
 - Lớp trừu tượng vẫn có các phương thức thông thường còn có các khai báo phương thức.
 - Các phương thức chỉ khai báo gọi là phương thức trừu tượng.
 - Lớp dẫn xuất có trách nhiệm định nghĩa lại các phương thức trừu tượng của lớp cha
 - Sử dụng các lớp trừu tượng đóng vai trò quan trọng trong thiết kế phần mềm.
 - Cho phép định nghĩa phần tử dùng chung trong cây kế thừa nhưng quá khái quát để tạo ra các thể hiện.

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn>

© 2021 Hanoi University of Industry All rights reserved

63



4. LỚP TRỪU TƯỢNG

- Hiện thực lớp trừu tượng
 - Khai báo lớp với từ khóa **abstract**
 - Lớp chứa ít nhất một phương thức không có mã nguồn-
phương thức trừu tượng.
 - Lớp con khi kế thừa phải **cài đặt cụ thể** phương thức trừu tượng của lớp cha.

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn>

© 2021 Hanoi University of Industry All rights reserved

64



4. LỚP TRỪU TƯỢNG

- Cú pháp khai báo lớp trừu tượng

```
abstract class class_name {
    //Khai báo các thành phần thuộc tính
    [modifier] data_type field_name [=Value];
    [modifier] data_type field_name [=Value];
    // Xây dựng các phương thức bình thường
    [modifier] data_type method_name ( Parameters ) { <code> }
    // Khai báo các phương thức trừu tượng nếu có
    abstract [modifier] data_type method_name (Parameters);
}
```

Lưu ý phạm vi: public; protected

Không cho phép: private; static; final

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn>

© 2021 Hanoi University of Industry All rights reserved

65



4. LỚP TRỪU TƯỢNG – MINH HỌA

```
public abstract class Hình {
    private String tenHình;
    public abstract double
    tinhChuVi();
    public abstract double
    tinhDienTich();
}

public class HìnhChuNhat extends
Hình {
    private double chieuDai;
    private double chieuRong;
```

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

```
@Override
public double tinhChuVi() {
    double cv;
    cv = (chieuDai+chieuRong)*2;
    return cv;
}

@Override
public double tinhDienTich() {
    double dt;
    dt = chieuDai*chieuRong;
    return dt;
}
```

Website: <https://hau.edu.vn>

© 2021 Hanoi University of Industry All rights reserved

66



4. LỚP TRỪU TƯỢNG

- Đặc điểm lớp trừu tượng
 - Không thể khởi tạo một đối tượng thuộc lớp trừu tượng (abstract)
 - Chỉ khởi tạo **đối tượng thuộc lớp cụ thể** (concrete).
 - Khi lớp con chưa cụ hóa hết hành vi trừu tượng của lớp cha thì lớp con vẫn là lớp trừu tượng

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn>

© 2021 Hanoi University of Industry All rights reserved

67



TỔNG KẾT LỚP TRỪU TƯỢNG.

1. Theo phương pháp phân cấp thừa kế, lớp cha có những hành vi chưa biết viết mã lệnh thế nào thì đó là hành vi đó là trừu tượng
2. Hành vi/phương thức trừu tượng chỉ thuộc lớp trừu tượng.
3. Nếu muốn một lớp được thừa kế từ nhiều lớp trừu tượng mà không bị khổng chế bởi tính đơn thừa kế.

- Giải pháp: Interface

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn>

© 2021 Hanoi University of Industry All rights reserved

68



NỘI DUNG

1. Xây dựng lớp (giáo trình trang 85 – 100)
2. Giới thiệu một số lớp cơ bản trong java (
3. Kết tập, kế thừa và đa hình (giáo trình trang 102 – 124)
4. Lớp trừu tượng (giáo trình trang 124 – 131)
5. Interface - Giao diện (giáo trình trang 131 – 135)
6. Gói và quản lý gói (giáo trình trang 101)

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn>

© 2021 Hanoi University of Industry All rights reserved

69



5. INTERFACE.

- Khái niệm
 - Interface mang ý nghĩa **“như là một lớp hoàn toàn trừu tượng”** khai báo trước một **nhóm các xử lý cần có**.
 - Interface giải quyết vấn đề không muốn hạn chế bởi tính đơn kế thừa thông qua khai báo lớp
 - **Các khai báo trong interface: các hằng, các hành vi.**
 - Interface là một đặc điểm của các ngôn ngữ OOP mới như Java, C#. Interface là một công cụ để hiện thực dạng đa thừa kế trong Java, C#.

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn>

© 2021 Hanoi University of Industry All rights reserved

70



5. INTERFACE.

- Cú pháp

```
[modifier] interface Interface_name{
    [ modifier] <final const_names>;
    [modifier] data_type method_name (parameters);
}
```

- Chọn phạm vi là public, mặc định không khai báo
- Nên khai báo 1 Interface riêng trong 1 file.java để trình biên dịch thành file.class từ đó hệ thống dễ quản lý sơ đồ các lớp và file.

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn>

© 2021 Hanoi University of Industry All rights reserved

71



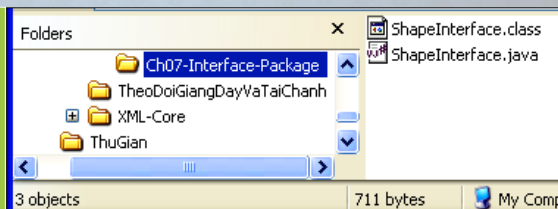
5. INTERFACE.

ShapeInterface.java

```
1 // ShapeInterface.java
2 public interface ShapeInterface
3 { final public double Pi=3.141592;
4   final public double pi=3.141592;
5   public double calcArea(); // tính diện tích
6   public double calcPerimeter(); // tính chu vi
7 }
```



Nhận xét:

Các hình vẽ đều cần
-tác vụ tính diện tích,
-hằng PI
-tính chu vi
-mà ta không muốn tạo
lớp trừu tượng
→ Tạo 1 interface.

Website: <https://hau.edu.vn>

© 2021 Hanoi University of Industry All rights reserved

72

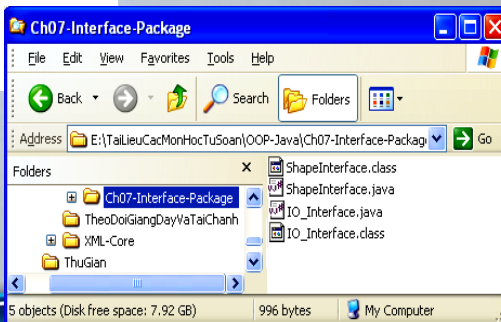

LẬP TRÌNH JAVA


5. INTERFACE.



- Tạo interface nhập/xuất dữ liệu

```

1 // IO_Interface.java
2 interface IO_Interface
3 { void input();
4   void output();
5 }
        
```




BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG
Website: <https://hau.edu.vn>


LẬP TRÌNH JAVA



5. INTERFACE.

- Hiện thực lớp triển khai interface
 - Một lớp có mã nguồn cụ thể hóa các hành vi của một interface thì gọi là lớp triển khai (**implementation**)
 - Một lớp có thể hiện thực nhiều interface (có dạng đa thừa kế).
 - Cú pháp xây dựng một lớp có hiện thực interface như sau:

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG
Website: <https://hau.edu.vn>



LẬP TRÌNH JAVA



5. INTERFACE.

- Cú pháp lớp hiện thực Interface

[modifier] class class_name extends super_class implements Interface1, Interface2,...{ //body }

<Data>

<method>

methods of interface1

methods of interface2


methods of interfaceX

}


Thứ tự
không
quan
trọng

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn> © 2021 Hanoi University of Industry All rights reserved 75



LẬP TRÌNH JAVA



5. INTERFACE.

```

public class HìnhTron implements
    ShapeInterface , IO_Interface{

    int r;

    public void input(){
        System.out.print("nhap r=");
        r=s.nextInt();    }

    public void output(){
        System.out.println("hinh tron r="+r
        +" chu vi =" +calcArea()+
        " dien tich = " + calcPerimeter());
    } }
        
```

```

public class HìnhTronDemo {

    public double calcArea(){
        return 2*pi*r;  }

    public double calcPerimeter(){
        return pi*r*r;  }

    public static void main(...) {
        HìnhTron h = new
        HìnhTron();
        h.input();
        h.output();
    } }
        
```

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn> © 2021 Hanoi University of Industry All rights reserved 76



5. INTERFACE.

- Kế thừa trong interface

- Interface có thể thừa kế interface cha
- Interface con có hội các khai báo hành vi.
- Cú pháp:

```
interface InterfaceSon extends InterfaceFather
{ < các khai báo thêm> }
```

- Nếu một lớp implements interface con thì phải cụ thể hóa cả các phương thức có trong interface cha.

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn>

© 2021 Hanoi University of Industry All rights reserved

77



NỘI DUNG



1. Xây dựng lớp (giáo trình trang 85 – 100)
2. Giới thiệu một số lớp cơ bản trong java (
3. Kết tập, kế thừa và đa hình (giáo trình trang 102 – 124)
4. Lớp trừu tượng (giáo trình trang 124 – 131)
5. Interface-Giao diện (giáo trình trang 131 – 135)
6. Gói và quản lý gói (giáo trình trang 101)

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn>

© 2021 Hanoi University of Industry All rights reserved

78


LẬP TRÌNH JAVA


6. GÓI (PACKAGE)

• Giới thiệu

Bài toán

class C1
class C2
class C3
.....
class Cn

Quản lý thế nào?
Phân nhóm
theo chức năng

Nhóm 1
class C1
class C2
class C3

Nhóm 2
class C1
class C2
class C3
Nhóm 2_1
class C4
class C5
Nhóm 2_2
class C6
class C7



Nhóm ...
class C8
....
class Cn

Mỗi nhóm để trong
một thư mục
→ Package

Phân tích, quản lý
project

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn> © 2021 Hanoi University of Industry All rights reserved 79


LẬP TRÌNH JAVA



6. GÓI (PACKAGE)

• Package: gói phần mềm.


- Là một nhóm các class, interface, các gói khác đã được biên dịch thành Java bytecode.
- Gói là công cụ tạo khả năng tái sử dụng mã (reusable code).
- Tổ chức của 1 package là 1 thư mục có tên của package
- Sub-package là 1 gói con (thư mục con) của 1 package mức cao hơn (giống cấu trúc thư mục).

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn> © 2021 Hanoi University of Industry All rights reserved 80



LẬP TRÌNH JAVA



6. GÓI (PACKAGE)

- Khai báo package trong mã nguồn của lớp (class).
 - Dùng từ khóa **package**
 - Cú pháp: **package PackageCha[.PackageCon];**
 - Khai báo package là **đòng đầu tiên** của code java.
- Ví dụ:


```
package PackageCha[.PackageCon];

[import OtherPackageName ;]


class ClassName
{ .....
}
```

Khi biên dịch thành công:


- Nếu chưa có thư mục PackageName thì thư mục này sẽ được tự động tạo ra
- file.class và mã nguồn sẽ được đặt vào thư mục này.

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn> © 2021 Hanoi University of Industry All rights reserved 81



LẬP TRÌNH JAVA



6. GÓI (PACKAGE)

- Sử dụng package
 - Dùng chỉ thị **import** với một trong các cú pháp sau:
- Ví dụ:


```
import PackageName.*;
import PackageCha.PackageCon.*;
import PackageName.ClassName;
import PackagCha.PackageCon.ClassName;
```

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn> © 2021 Hanoi University of Industry All rights reserved 82



TỔNG KẾT BÀI 3

1. Xây dựng lớp:
 1. phạm vi truy cập: public, private, protected
 2. super: Từ khóa truy cập lớp cha
 3. Một số từ khóa chỉ định: static; final
2. Kế thừa trong Java: đơn kế thừa
3. Lớp trừu tượng: Khái quát cao kế thừa
4. Interface bao gồm khai báo các hằng, các hành vi
5. Gói – quản lý các lớp, interface và gói khác

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn>

© 2021 Hanoi University of Industry All rights reserved

83



MỞ RỘNG - KHUYẾN KHÍCH TÌM HIỂU

- Lớp nội
- Vấn đề ép kiểu
- Toán tử phân biệt đối tượng thuộc lớp
- Tham số không xác định trong java
- Tham số của phương thức trong java

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn>

© 2021 Hanoi University of Industry All rights reserved

84



CHUẨN BỊ BUỔI HỌC TRÊN LỚP

- Các nhóm đọc slide, tài liệu tham khảo.
- Xem các video hướng dẫn bài 3
- Làm việc nhóm thực hiện phiếu bài tập 1,2,3.
- Nộp bài tập nhóm theo địa chỉ lớp trực tuyến theo quy định.

BÀI 3: LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Website: <https://hau.edu.vn>

© 2021 Hanoi University of Industry All rights reserved

85