

**I. MỤC TIÊU:**

- Tìm hiểu các khái niệm trong lập trình hướng đối tượng:
  - Thuộc tính: nội tại, class, tham chiếu
  - Phương thức: Hàm dựng, Get/Set, hành vi
  - Phương thức: toString(), hashCode() & equals()
- Cấu trúc tập: ArrayList, HashSet, HashMap
- Mô hình quan hệ

**II. TÓM TẮT LÝ THUYẾT:****A. Các khái niệm trong lập trình hướng đối tượng****1. Lớp và đối tượng:***a. Khai báo lớp:*

```
class <TenLop> {  
    Thuộc tính 1  
    Thuộc tính 2  
    ...  
    Hàm dựng 1  
    Hàm dựng 2  
    ...  
    Hành động 1  
    Hành động 2  
    ...  
}
```

*b. Tạo đối tượng:*

```
<TenLop> a = new TenLop();
```

**2. Thuộc tính:***a. Nội tại:*

- Khai báo:

```
class <TenLop> {  
  
    //Khai báo thuộc tính đối tượng  
  
    <kieu_du_lieu> ten_bien1;  
    <kieu_du_lieu> ten_bien2;  
    ...  
}
```

- Truy cập:

```
<TenLop> a = new <TenLop>();  
  
a.ten_bien1;  
  
a.ten_bien2;
```

*b. Lớp:*

- Khai báo:

```
class <TenLop> {  
    ...  
    static <kieu_du_lieu> ten_bien_lop;  
}
```

- Truy cập:

```
<TenLop>.ten_bien_lop;
```

*c. Tham chiếu:*

- Khai báo:

```
class <TenLop> {  
    //Khai báo thuộc tính đối tượng  
    <TenLopThamChieu> doi_tuong_1;  
    <TenLopThamChieu> doi_tuong_2;  
    ...  
}
```

- Truy cập:

```
<TenLop> b = new <TenLop>();  
b.doi_tuong_1.phuongthuc();  
b.doi_tuong_2.phuongthuc();
```

### 3. Phương thức:

#### a. Phương thức của đối tượng:

- Khai báo:

```
class <TenLop> {  
    Thuộc tính 1  
    Thuộc tính 2  
    ...  
    Hàm dựng 1  
    Hàm dựng 2  
    ...  
    //Khai báo các phương thức của đối tượng  
    <kieu_du_lieu> TenPhuongThuc1(tham số) {  
        //Định nghĩa  
    }  
    <kieu_du_lieu> TenPhuongThuc1(tham số) {  
        //Định nghĩa  
    }  
    ...  
}
```

- Truy cập:

```
<TenLop> a = new <TenLop>();  
a.TenPhuongThuc1(tham số);  
a.TenPhuongThuc2(tham số);
```

b. Phương thức lớp:

- Khai báo:

```
class <TenLop> {  
    Thuộc tính 1  
    Thuộc tính 2  
    ...  
    Hàm dựng 1  
    Hàm dựng 2  
    ...  
    //Khai báo các phương thức của lớp  
    static <kieu_du_lieu> TenPhuongThuc1(tham số) {  
        //Định nghĩa  
    }  
    static <kieu_du_lieu> TenPhuongThuc2(tham số) {  
        //Định nghĩa  
    }  
    ...  
}
```

- Truy cập:

```
<TenLop>.TenPhuongThuc1(tham số);  
<TenLop>.TenPhuongThuc2(tham số);
```

#### 4. Hàm dựng:

- Dùng để khởi gán giá trị ban đầu cho đối tượng, cùng tên với tên lớp, không có kiểu trả về (kể cả kiểu void)

- Khai báo:

```
class <TenLop> {  
    Thuộc tính 1  
    Thuộc tính 2  
    ...  
    <TenLop>() {  
        //Định nghĩa  
    }  
    <TenLop>(tham số) {  
        //Định nghĩa  
    }  
    ...  
}
```

- Gọi hàm dựng:

```
<TenLop> a = new <TenLop>();  
<TenLop> b = new <TenLop>(tham số);
```

**5. Phương thức Get/Set:**

- **Bao đóng** : là khả năng che dấu thông tin của đối tượng.
- Sử dụng các quyền truy cập trong java để che dấu thông tin:

Modifier	Class	Package	Subclass	World
public	✓	✓	✓	✓
protected	✓	✓	✓	✗
no modifier*	✓	✓	✗	✗
private	✓	✗	✗	✗

- **private**: dùng để che dấu
- **public**: để bên ngoài sử dụng
- Cung cấp phương thức **Get**(để đọc) / **Set**(để ghi) cho các thuộc tính private.

**6. Phương thức toString(), hashCode() & equals():**

- Được thừa kế từ lớp Object.
- *Phương thức toString()*: chuyển đối tượng thành đối tượng String, thường được dùng để mô tả đối tượng
- *Phương thức hashCode() & equals()*: dùng để định danh đối tượng, mặc định java dùng địa chỉ bộ nhớ của đối tượng để định danh, trên thực tế ta dùng thuộc tính khóa để định danh đối tượng

**B. Cấu trúc tập:**

1. **ArrayList:** lưu danh sách các phần tử có thứ tự, được phép trùng nhau.

- *Khai báo:*

`List<Kiểu dữ liệu> ds = new ArrayList<>();`

- *Các phương thức thường dùng:*

`add()`: thêm phần tử

`get()`: lấy phần tử

`remove()`: xóa phần tử

`size()`: trả về số phần tử trong danh sách

`indexOf()`: trả về vị trí đầu tiên của phần tử

`lastIndexOf()`: trả về vị trí cuối cùng của phần tử

2. **HashSet:** lưu danh sách các phần tử không có thứ tự, không được phép trùng nhau.

- *Khai báo:*

`Set<Kiểu dữ liệu> ds = new HashSet<>();`

- *Các phương thức thường dùng:*

`add()`: thêm phần tử

`remove()`: xóa phần tử

`size()`: trả về số phần tử trong danh sách

`toArray()`: trả về mảng các phần tử trong danh sách

3. **HashMap:** lưu danh sách các phần tử dưới dạng bảng băm (key → value)

- *Khai báo:*

`Map<K, V> ds = new HashMap<>();`

K: kiểu dữ liệu dùng làm khóa

V: kiểu dữ liệu giá trị cần lưu trữ

- *Các phương thức thường dùng:*

`put()`: thêm phần tử

`get()`: lấy phần tử

`remove`: xóa phần tử

`keySet()`: trả về tập khóa của bảng băm

`size()`: trả về số phần tử

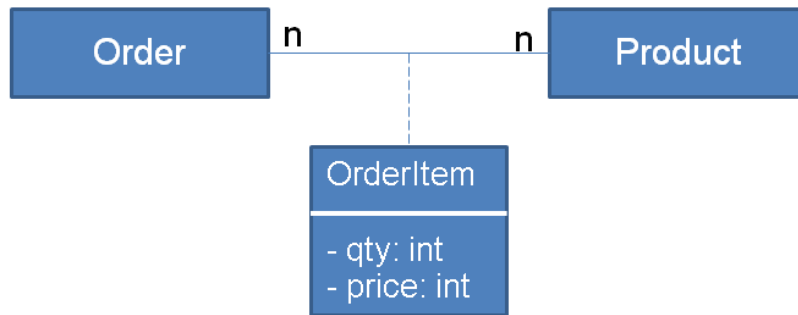
**C. Mô hình quan hệ:****1. Quan hệ 1-n:**

```
class Customer {
    List<AccountBank> accs;
}
class AccountBank {
    Customer cus;
}
```

**2. Quan hệ n-n (không có thuộc tính kết hợp):**

```
class Employee {
    List<Project> pros;
}
class Project {
    List<Employee> ems;
}
```

### 3. Quan hệ n-n (có thuộc tính kết hợp)



```
class Order {
    int orderId;
    List<OrderItem> items;
}
class Product {
    int productId;
    List<OrderItem> items;
}
```

```
class OrderItemId {
    int orderId;
    int productId;
    ...
}
```

```
class OrderItem {
    OrderItemId id;
    Order order;
    Product product;
    int qty;
    int price;
}
```

**III. NỘI DUNG THỰC HÀNH:**

1. Mở **Lab2\_1.java**: Khai báo lớp hình tròn gồm 1 thuộc tính **r**, có 2 hàm dựng không đối số và 1 đối số, phương thức **Get/Set**, phương thức hành vi **tínhCV**, phương thức **toString** in thông tin đối tượng.

```
1 public class Lab2_1 {
2     public static void main(String[] args) {
3         HìnhTron ht;
4         ht = new HìnhTron(10);
5         System.out.println(ht);
6     }
7 }
8 class HìnhTron {
9     private double r;
10    public HìnhTron() {
11    }
12    public HìnhTron(double r) {
13        this.r = r;
14    }
15    public double getR() {
16        return r;
17    }
18    public void setR(double r) {
19        this.r = r;
20    }
21    public double tínhCV() {
22        return 2 * Math.PI * r;
23    }
24    public String toString() {
25        return "HìnhTron: r=" + r + ";CV=" + tínhCV();
26    }
27 }
```

**Yêu cầu:**

- Chạy chương trình, quan sát kết quả.
- Hãy viết thêm phương thức hành vi **tínhDT** tính diện tích hình tròn, chỉnh sửa code trong phương thức **toString** để hiển thị thêm diện tích.

**Bài tập 1:**

- Viết chương trình khai báo lớp **HìnhChuNhat** gồm 2 thuộc tính dài, rộng, có 2 hàm dựng không đối số và 2 đối số, phương thức **Get/Set**, phương thức hành vi **tínhCV**, **tínhDT** phương thức **toString** in thông tin đối tượng.



2. Mở **Lab2\_2.java**: Chương trình khai báo danh sách đối tượng **SinhVien**, in danh sách sinh viên.

```
1  import java.util.ArrayList;
2  import java.util.List;
3  public class Lab2_2 {
4      public static void main(String[] args) {
5          List<SinhVien> dssv = new ArrayList<>();
6          dssv.add(new SinhVien("111", "Bach", 7));
7          dssv.add(new SinhVien("222", "Kha", 7.5));
8          dssv.add(new SinhVien("333", "Duc", 6.5));
9          dssv.add(new SinhVien("444", "Nghia", 3));
10         dssv.add(new SinhVien("555", "Tam", 9));
11         for(int i = 0; i<dssv.size(); i++)
12             System.out.println(dssv.get(i));
13     }
14 }
15 class SinhVien {
16     private String mssv;
17     private String hoten;
18     private double dtb;
19     public SinhVien() {
20     }
21     public SinhVien(String mssv, String hoten, double dtb) {
22         this.mssv = mssv;
23         this.hoten = hoten;
24         this.dtb = dtb;
25     }
26     public String getMssv() {
27         return mssv;
28     }
29     public void setMssv(String mssv) {
30         this.mssv = mssv;
31     }
32     public String getHoten() {
33         return hoten;
34     }
35     public void setHoten(String hoten) {
36         this.hoten = hoten;
37     }
38     public double getDtb() {
39         return dtb;
40     }
41     public void setDtb(double dtb) {
42         this.dtb = dtb;
43     }
44     @Override
45     public String toString() {
46         return "SinhVien{" + "mssv=" + mssv +
47             ", hoten=" + hoten + ", dtb=" + dtb + '}';
48     }
49 }
```

### Yêu cầu:

- Chạy chương trình, quan sát kết quả.
- Hãy viết thêm code tìm điểm trung bình lớn nhất trong danh sách.

### Bài tập 2:

- Khai báo lớp **NhanVien** gồm 3 thuộc tính: **hoten**, **ngaycong**, **luongcb**. Hai hàm dựng không đối số và 3 đối số, phương thức **Get/Set**, phương thức hành vi **getLuong() = ngaycong x luongcb**, phương thức **toString** in thông tin nhân viên.
- Trong chương trình chính, khai báo danh sách các đối tượng **NhanVien**, in danh sách nhân viên, tính trung bình lương toàn nhân viên.

### Bài tập làm thêm: (làm ở nhà)

- Cài đặt bài tập 2 bằng **Set** và **Map**.

**3. Lab2\_3:** Cho sơ đồ lớp như hình

- Mở **Lab2\_3.java**: Khai báo sơ đồ lớp như hình trên. Trong chương trình chính, tạo và in danh sách đối tượng Catalog và các đối tượng Book tương ứng.

```
1  import java.util.*;
2  public class Lab2_3 {
3      public static void main(String[] args) {
4          List<Catalog> catalogs = new ArrayList<>();
5          Catalog cata1, cata2, cata3;
6          cata1 = new Catalog(1, "Java");
7          cata1.addBook(new Book(1, "Java can ban", 5000));
8          cata1.addBook(new Book(2, "Java nang cao", 7000));
9
10         cata2 = new Catalog(2, "Mobile");
11         cata2.addBook(new Book(3, "Lap trinh Android", 10000));
12         cata2.addBook(new Book(4, "Swish", 12000));
13
14         cata3 = new Catalog(3, "Web");
15         cata3.addBook(new Book(5, "JSF", 6000));
16
17         catalogs.add(cata1);
18         catalogs.add(cata2);
19         catalogs.add(cata3);
20
21         for(Catalog catalog:catalogs) {
22             System.out.println("\n" + catalog);
23             for(Book b:catalog.getBooks())
24                 System.out.println(b);
25         }
26     }
27 }
```

```
28
29  class Catalog {
30      private int id;
31      private String name;
32  +   public Catalog() {...2 lines }
34  +   public Catalog(int id, String name) {...4 lines }
38  +   public int getId() {...3 lines }
41  +   public void setId(int id) {...3 lines }
44  +   public String getName() {...3 lines }
47  +   public void setName(String name) {...3 lines }
50  @Override
51  +   public String toString() {...3 lines }
54
55  //Mot doi tuong Catalog co nhieu doi tuong Book
56  private List<Book> books = new ArrayList<>();
57  -   public List<Book> getBooks() {
58  |       return books;
59  |   }
60  -   public void setBooks(List<Book> books) {
61  |       this.books = books;
62  |   }
63  -   public void addBook(Book b) {
64  |       books.add(b);
65  |       b.setCatalog(this);
66  |   }
67  -   public void removeBook(Book b) {
68  |       books.remove(b);
69  |       b.setCatalog(null);
70  |   }
71  }
```

```
72
73  class Book {
74      private int id;
75      private String title;
76      private double price;
77      + public Book() {...2 lines }
79      + public Book(int id, String title, double price) {...5 lines }
84      + public int getId() {...3 lines }
87      + public void setId(int id) {...3 lines }
90      + public String getTitle() {...3 lines }
93      + public void setTitle(String title) {...3 lines }
96      + public double getPrice() {...3 lines }
99      + public void setPrice(double price) {...3 lines }
102  @Override
103  + public String toString() {...3 lines }
106
107      //Mot doi tuong Book thuoc mot doi tuong Catalog
108      private Catalog catalog;
109      - public Catalog getCatalog() {
110          |     return catalog;
111          | }
112      - public void setCatalog(Catalog catalog) {
113          |     this.catalog = catalog;
114          | }
115  }
```

**Yêu cầu:**

- Chạy chương trình, quan sát kết quả.
- Viết thêm code như sau:
  - Thêm 2 đối tượng Book vào đối tượng *catalog* Web.
  - Xóa đối tượng Book có id là 1 trong *catalog* Java.

**Bài tập 4:** Cho sơ đồ lớp như hình

- Khai báo sơ đồ lớp trên. Viết chương trình tạo và in ra danh sách **Department** và danh sách **Employee** tương ứng.

**IV. BÀI TẬP LÀM THÊM:****1. Tạo lớp `Student` gồm 5 thuộc tính:**

- ✓ `firstName` (String)
- ✓ `lastName` (String)
- ✓ `exam1` (double);
- ✓ `exam2` (double);
- ✓ `exam3` (double);

- Khai báo hàm dựng để khởi tạo các thuộc tính cho đối tượng **`Student`**
- Tạo phương thức **`get`** và **`set`** để truy cập các thuộc tính.
- Viết phương thức **`getAverage()`** để tính điểm trung bình của 3 môn.
- Viết chương trình chính (**`StudentTest`**) sử dụng lớp **`Student`** như sau:
  - Tạo lớp **`Student`** với giá trị khởi tạo ("Kha", "Ho", 83, 90, 95)
  - In thông tin đối tượng theo mẫu sau:  
"Chao ban **Kha Ho**, Diem trung binh cua ban la: **89**"
  - Cập nhật điểm cho đối tượng như sau: **`exam1`** = 93, **`exam2`** = 91, **`exam3`** = 86
  - In lại thông tin điểm trung bình sau khi đã cập nhật

**2. Tạo lớp `NgayThang` gồm 3 thuộc tính:**

- ✓ `Ngay` (int)
- ✓ `Thang` (int)
- ✓ `Nam` (int)

- Khai báo hàm dựng để khởi tạo các thuộc tính cho đối tượng **`NgayThang`**
- Tạo phương thức **`get`** và **`set`** để truy cập các thuộc tính.
- Viết phương thức **`kiemtra()`** để kiểm tra ngày tháng có hợp lệ không
- Viết phương thức **`In()`** trả về kiểu String như sau:
  - Nếu ngày tháng hợp lệ, định dạng như sau: `Ngay/Thang/Nam`
  - Ngược lại, trả về "Ngày tháng không hợp lệ!"
- Viết chương trình chính (**`NgayThangTest`**) sử dụng lớp **`NgayThang`** như sau:
  - Tạo lớp **`NgayThang`** với giá trị khởi tạo (11, 12, 2015)
  - In thông tin ngày tháng vừa tạo: 11/12/2015
  - Cập nhật điểm cho đối tượng như sau: **`Ngay`** = 30, **`Thang`** = 2
  - In lại thông tin ngày tháng năm sau khi đã cập nhật:
    - "Ngày tháng không hợp lệ!"

3. Tạo một lớp được gọi là HoaDon của một cửa hàng phần cứng lưu lại thông tin món hàng được bán tại các cửa hàng. Một HoaDon phải bao gồm bốn thông tin như sau:

- Tên món hàng (kiểu String)
- Mô tả (kiểu String)
- Số lượng mặt hàng được mua (kiểu int)
- Giá cho mỗi mặt hàng (kiểu double)

Lớp HoaDon nên có một hàm dựng khởi tạo giá trị ban đầu. Cung cấp các phương thức get và set cho mỗi thuộc tính của HoaDon. Ngoài ra, cung cấp một phương thức có tên getThanhTien() tính số tiền phải trả của HoaDon đó. Số lượng và giá của một mặt hàng phải là số >0, ngược lại thiết lập bằng 0.

Tạo lớp KháchHang lưu thông tin của khách hàng bao gồm các thuộc tính:

- Mã khách hàng
- Họ tên
- Địa chỉ
- Email

**Một khách hàng có thể có nhiều HoaDon.**

Viết một ứng dụng thử nghiệm tên QLHH tạo đối tượng khách hàng và hóa đơn của khách hàng đó.

4. Tạo một lớp được gọi là DuAn của một công ty lưu lại thông tin dự án công ty đang làm. Thông tin DuAn như sau:

- Mã dự án (kiểu String)
- Tên dự án (kiểu String)
- Ngày bắt đầu (kiểu Date)
- Ngày kết thúc (kiểu Date)
- Số người tối đa tham gia dự án

Lớp DuAn có một hàm dựng khởi tạo giá trị ban đầu. Cung cấp các phương thức get và set cho mỗi thuộc tính của DuAn. Đối với thuộc tính ngày bắt đầu và ngày kết thúc

Tạo lớp NhanVien gồm các thông tin:

- MSNV
- Họ tên
- Lương

**DuAn có thể có nhiều NhanVien tham gia, và NhanVien có thể tham gia nhiều DuAn**

**Ràng buộc:** Số người tham gia dự án không vượt quá số người tối đa tham gia dự án, mỗi nhân viên chỉ được tham gia tối đa 3 dự án.

Viết một ứng dụng thử nghiệm tên DuAnTest tạo 4 đối tượng DuAn, 10 đối tượng NhanVien. Cho các nhân viên tham gia dự án và kiểm tra các ràng buộc trên.