

Tuần 2: XÂY DỰNG LỚP, TẠO ĐỐI TƯỢNG



Mục tiêu

1. Nắm vững cấu trúc của một lớp:
 - Thuộc tính (property)
 - Hành vi (behavior, method)
2. Phạm vi truy xuất thành phần của lớp trong cùng một gói
 - Public: dùng chung tại mọi vị trí
 - Private: chỉ được truy xuất trong phạm vi lớp đó
3. Tạo đối tượng, truyền thông điệp cho đối tượng (gọi thực thi một phương thức)

Bài tập

Bài 1: Xây dựng lớp học sinh, biết rằng mỗi học sinh có:

- Thành phần dữ liệu: mã số, họ tên, điểm trung bình
 - Phương thức: set(), get(), input(), output(), rank() - xếp loại cho học sinh theo dtb.
- Viết lớp Demo1 chứa phương thức main():

- Tạo một đối tượng học sinh
- Nhập thông tin cho học sinh
- Xuất thông tin cùng xếp loại của học sinh
- Xem họ tên của đối tượng học sinh vừa tạo và thay bằng một họ tên khác (chuỗi họ tên mới nhập từ bàn phím).

Bài 2: Viết lớp Demo2 chứa phương thức main():

- Tạo một danh sách các đối tượng học sinh
- Nhập thông tin cho danh sách học sinh
- In danh sách học sinh

Bài 3: Sử dụng bài 1, tiếp tục xây dựng lớp DSHOCSINH (danh sách học sinh) gồm các thuộc tính:

- Mảng các đối tượng học sinh
- Số lượng học sinh

Và phương thức:

- Nhập danh sách
- In danh sách
- Sắp xếp danh sách giảm dần theo điểm trung bình của học sinh

Viết lớp Demo4 chứa phương thức main():

- Tạo một đối tượng danh sách học sinh
- Nhập thông tin cho danh sách học sinh
- In danh sách học sinh đã được sắp thứ tự.

Bài 4: Xây dựng lớp Mang gồm có các Thành phần dữ liệu:

- Thành phần dữ liệu: Số phần tử của mảng, mảng chứa các số nguyên, kích thước của mảng
- Phương thức:
 - input(), output()
 - Tính trị trung bình của các số lẻ
 - Tìm phần tử lớn nhất của mảng
 - Sắp xếp mảng theo thứ tự tăng dần

Viết lớp Demo3 chứa phương thức main():

- Tạo một đối tượng mảng
- Thực hiện các phương thức cho đối tượng vừa tạo.

Hướng dẫn

Bài 1:

1. Xây dựng lớp HOCSINH

```
import java.io.*;

class HOCSINH{

}
```

Viết các Property và các Method bên trong lớp

- Khai báo các Properties có phạm vi truy xuất là *private* (để bảo mật dữ liệu, không cho truy xuất bên ngoài lớp).

Cú pháp: `private kiểu_dữ_liệu tên_biến;`

```
private int maso;
private String hoten;
private float dtb;
```

- Để truy xuất được các thuộc tính từ bên ngoài lớp, ta cung cấp các phương thức lấy giá trị (get) và gán giá trị (set) cho thuộc tính

```
//thiết lập giá trị cho Properties
public void setMaSo( int ms){
    maso=ms;
}
public void setHoTen(String hoten){
    this.hoten=hoten;
}
//tự làm với dtb
```

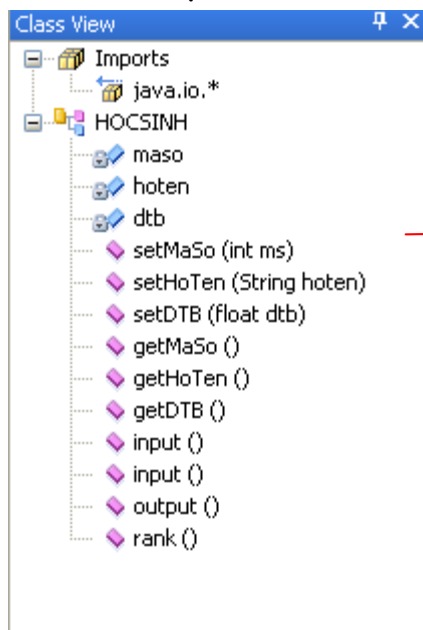
*Trùng tên, không phân biệt được nên phải dùng **this** (**this.hoten** chỉ thuộc tính họ tên của lớp đang xét)*

```
//lấy giá trị của properties
public int getMaSo() {
    return maso;
}
//tương tự với họ tên và điểm trung bình
```

- Viết phương thức *input()* để nhập thông tin của 1 HOCSINH
- Viết phương thức *output()* để xuất thông tin của lớp HOCSINH
- Viết phương thức *rank()* để xếp loại cho học sinh theo điểm trung bình

```
//xếp hạng học sinh theo đtb
public void rank() {
    if (dtb < 5)
        System.out.println("Xep loại yeu");
    else
        if (dtb >= 5 && dtb < 7)
            System.out.println("Xep loại trung binh");
        else
            //...tự viết tiếp
}
```

- Sau khi viết các thuộc tính và phương thức, vào menu *View* → *Class View*, ta thấy cửa sổ Class View hiện ra bên trái như sau:



*Phát sinh theo source code
của chương trình*

- Trong cùng file Demo1.java đang làm, tiếp tục xây dựng lớp Demo1 chứa phương thức main như sau:

```
//=====
class Demo1{
    public static void main (String[] args) {
        HOCSINH hs=new HOCSINH(); //tạo một đối tượng HOCSINH hs
        hs.input(); //gọi phương thức nhập cho đối tượng hs
        hs.output(); //gọi phương thức xuất cho đối tượng hs
        hs.rank(); //gọi phương thức xếp loại cho đối tượng hs
    }
}
```

- Xem và đổi tên của đối tượng học sinh vừa tạo, sử dụng phương thức getHoten() và setHoten() như sau (viết tiếp trong hàm main):

```
System.out.println("Ho ten cua hs vua nhap:" + hs.getHoten());
//nhập chuỗi họ tên mới
String htmoi="";
//bạn tự nhập...:D
//thay họ tên đã có của học sinh bằng chuỗi họ tên mới nhập vào
hs.setHoTen(htmoi);
//xem lại họ tên của học sinh sau khi sửa
System.out.println("Ho ten cua hs sau khi sua:" + hs.getHoten());
```

- Vào menu *View* → *Class View*, cửa sổ Class View lúc này có thêm lớp Demo1

Bài 2: Xây dựng lớp Demo2 chứa phương thức main() như sau:

```
//=====
class Demo2{
    public static void main (String[] args) {

        //khai báo mảng, khởi tạo bộ nhớ cho mảng chứa tối đa 20 hs
        HOCSINH[] dshs=new HOCSINH[20];
        int n=0;
        //nhập số lượng học sinh, nhập n
        //...tự viết

        //nhập thông tin cho từng học sinh dshs[i]
        for(int i=0; i<n; i++){
            dshs[i]=new HOCSINH(); //khởi tạo đối tượng HOCSINH thứ i
            dshs[i].input();
        }

        //xuất thông tin từng học sinh
        //...tương tự, tự viết

    }
}
```

Trong file lúc này có 2 lớp chứa hàm main(), chương trình sẽ ưu tiên chạy hàm main() ở lớp nào nằm trước (theo thứ tự từ trên xuống). Do đó, để chạy hàm main() từ lớp Demo2 trước thì *cut* và *paste* nó trên Demo1 hoặc đóng lớp Demo1 lại (dùng /* */)

❖ **Lưu ý:** Nếu trong file có một lớp khai báo public thì tên file phải trùng với tên lớp đó. (Không nên khai báo lớp là public).

Bài 3:

Sử dụng bài 1, tiếp tục xây dựng lớp DSHOCSINH:

```
class DSHOCSINH{
    //các thuộc tính
    private HOCSINH ds[]; //ds là mảng 1 chiều, mỗi phần tử lưu 1 hs
    private int sl; //số lượng phần tử của mảng

    //các phương thức
    public void nhapds(){
        //nhập số lượng học sinh: nhập sl
        //...tự code

        ds=new HOCSINH[sl]; //cấp phát bộ nhớ cho mảng ds
        //nhập thông tin cho từng người trong ds
        for(int i=0; i<sl; i++)
        {
            //phần tử thứ i trong mảng là 1 đối tượng học sinh nên phải khởi tạo đối tượng (cấp phát ô nhớ)
            ds[i]=new HOCSINH();
            ds[i].Input(); //nhập dữ liệu cho học sinh thứ i
        }
    }
    public void xuatds(){
        System.out.println("Danh sach hoc sinh la:\n");
        for(int i=0; i<sl; i++)
            ds[i].Output();
    }
    public void sapxep(){
        //bạn tự code - good luck !^!
    }
}
```

Viết lớp DEMO4 chứa phương thức main():

```
class DEMO4{
    public static void main (String[] args) {
        DSHOCSINH a=new DSHOCSINH();
        a.nhapds();
        a.xuatds();
        a.sapxep();
    }
}
```

Bài 4: Làm tương tự