

BÀI 2. TẠO CLASS

A. MỤC TIÊU:

- Xác định được đối tượng và lớp.
- Cài đặt được lớp với các phương thức khởi tạo và một số phương thức đơn giản, các hàm xử lý cơ bản, hàm static.
- Xây dựng được các phương thức: get/set, phương thức tính toán.
- Gọi được các đối tượng trong hàm chương trình.

B. DỤNG CỤ - THIẾT BỊ THỰC HÀNH CHO MỘT SV:

STT	Chủng loại – Quy cách vật tư	Số lượng	Đơn vị	Ghi chú
1	Computer	1	1	

C. NỘI DUNG THỰC HÀNH

1. Tóm tắt lý thuyết

1.1 Kiến thức cần nhớ

➤ **Lớp:** Một lớp là một kiểu cấu trúc mở rộng, đó là một kiểu mẫu chung cho các đối tượng thuộc cùng một loại. Như vậy, thành phần của lớp gồm cấu trúc dữ liệu mô tả các đối tượng trong lớp và các phương thức (còn gọi là hàm, hành vi, thao tác) mà mỗi biến đối tượng của lớp đều có.

Lớp được xem như một khuôn mẫu (template) của đối tượng (object).

Trong lớp bao gồm các thuộc tính của đối tượng (properties) và các phương thức (methods) tác động lên các thuộc tính.

➤ **Đối tượng:** Đối tượng được xây dựng từ lớp nên được gọi là thể hiện của lớp (class instance).

➤ **Cú pháp khai báo lớp**

```
[<từ khóa truy cập>] class <TÊN LỚP>
{
    <thành phần thuộc tính>;
    <thành phần phương thức>;
}
```

➤ **Cú pháp xây dựng phương thức: giống khai báo hàm nhưng được khai báo bên trong một lớp**

```
[<Từ khóa truy cập>] <kiểu trả về> <tên phương thức>([<ds tham số>])
{
    // thân phương thức
}
```

➤ **Cú pháp khai báo thuộc tính: khai báo giống như khai báo biến nhưng nằm bên trong một lớp**

```
[<Từ khóa truy cập>] <kiểu dữ liệu> <tên thuộc tính>;
```

➤ **Một số từ khóa chỉ phạm vi thông dụng**

- *public*: được sử dụng bên trong và bên ngoài lớp
- *private*: chỉ được sử dụng bên trong lớp
- *protected*: được sử dụng bên trong lớp và lớp con của nó
- *default*: được sử dụng bên trong lớp con và các lớp trong cùng một gói (package)

➤ **Cú pháp khai đối tượng:**

<Tên lớp> <Tên đối tượng>= new <Tên lớp>();

➤ **Cú pháp gọi và sử dụng các thành phần của lớp**

<Tên đối tượng>.<Tên thuộc tính>;

<Tên đối tượng>.<Tên phương thức>([<danh sách tham số>]);

1.2 Giới thiệu bài tập mẫu

Bài.1. Viết chương trình tính chu vi và diện tích hình tròn khi biết bán kính.

Phân tích: Cần xây dựng lớp Hình tròn có:

- Thuộc tính: bán kính (double)
- Phương thức:
 - Phương khởi tạo;
 - Phương get/set cho thuộc tính bán kính;
 - Phương thức nhập, chuvi, dientich;

Xây dựng lớp lớp **HìnhTron** như sau:

```
public class HTron {
    int r;
    public int getR() {
        return r;
    }
    public void setR(int r) {
        this.r = r;
    }
    public HTron() {
        this.r = 0;
    }
    public HTron(int r) {
        this.r = r;
    }
    public HTron(HTron a) {
        super();
        this.r = a.r;
    }
    public double tinhCV() {
        return Math.PI*r*2;
    }
    public double tinhDT() {
        return Math.PI*r*r;
    }
}
```

```

    }
    public void xuất() {
        System.out.println("Hình tron co CV: " + tinhCV() + " va
DT: " + tinhDT());
    }
}

```

1.3. Thành phần static (tĩnh)

- Thuộc tính tĩnh hay biến tĩnh (biến static) có thể được sử dụng để tham chiếu thuộc tính chung của tất cả đối tượng (mà không là duy nhất cho mỗi đối tượng).
- Thuộc tính này thuộc lớp chứ không thuộc riêng đối tượng của lớp.
- Có từ khóa **static** trước kiểu dữ liệu.
- Cách truy cập: <Tên Lớp>.<Thành viên tĩnh>
- Phương thức tĩnh là phương thức có từ khóa static trước kiểu dữ liệu trả về của phương thức đó khi khai báo phương thức.
- Trong thân của phương thức tĩnh chỉ được phép truy xuất những thành viên tĩnh.
- Các phương thức static chỉ có thể truy cập vào các thuộc tính static và chỉ có thể gọi các phương thức static trong cùng lớp.

1.2. Nạp chồng phương thức (Overload)

- Trong 1 Class có 2 hay nhiều hàm **trùng tên, trùng kiểu hàm (void/kiểu trả về)** nhưng **khác nhau về số lượng tham số hoặc khác nhau về kiểu dữ liệu tham số** thì được gọi là hàm chồng (nạp chồng phương thức).
- Trong một class, các hàm khởi tạo chính là nạp chồng phương thức.

Bài 1. Xây dựng class Student cho các sv trường HUFI có các thông tin: MSSV, họ tên, tên trường, số lượng SV.

Viết các phương thức:

- Properties (get/set)
- Hàm khởi tạo
- Hàm nhập xuất thông tin.

```

import java.util.Scanner;
public class Student{
    private int id;
    private String name;
    private static String college = "HUFI";

    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
}

```

```

    public void setName(String name) {
        this.name = name;
    }
    public static String getCollege() {
        return college;
    }
    public static void setCollege(String college) {
        Student.college = college;
    }
//Hàm khởi tạo cũng là nạp chồng phương thức - overload
//Hàm khởi tạo không tham số
    public Student()
    {
        id=0; name="Nguyen Van An";
    }
//Hàm khởi tạo có tham số
    public Student(int id,String name){
        this.id = id;
        this.name = name;
    }
//Hàm khởi tạo sao chép
    public Student(Student s) {
        id = s.id;
        name = s.name;
    }
    public void nhap()
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Moi nhap ID: ");
        id = sc.nextInt();
        System.out.println("Moi nhap ho ten: ");
        name = sc.nextLine();
    }

    public void xuat() {
        System.out.println("id: " + id + " - " + "name: " + name +
"college: " + college);
    }
    private static void thayDoiTruong(String college) {
        Student.college = college;
    }
}

```

- Xây dựng hàm main trong file Excute.java

```

public static void main(String args[]){
    System.out.println(Student.getCollege());
    Student a = new Student();
}

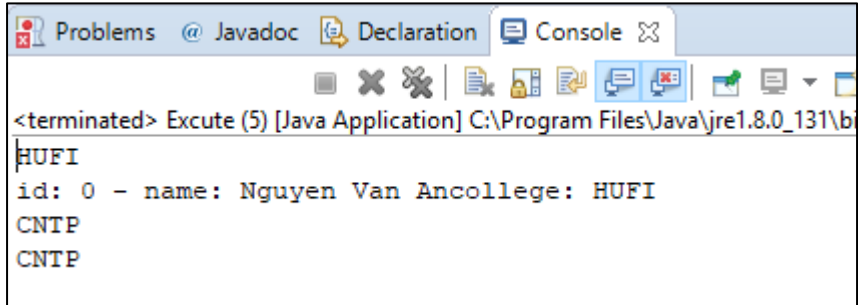
```

```

a.xuat();
Student.setCollege("CNTP");
System.out.println(Student.getCollege());
Student b = new Student();
System.out.println(Student.getCollege());
}

```

Kết quả chương trình



```

<terminated> Excute (5) [Java Application] C:\Program Files\Java\jre1.8.0_131\bin
HUFU
id: 0 - name: Nguyen Van An
college: HUFU
CNTP

```

2. Bài tập thực hành

Tạo lớp và gọi tất cả các hàm đã viết vào hàm main để thực thi:

Bài 1. Xây dựng lớp HCN với phương thức tính chu vi và diện tích.

Bài 2. Xây dựng lớp TamGiac có 2 phương thức tính chu vi và diện tích của hình tam giác.

Bài 3. Xây dựng lớp Nhân viên chứa các thông tin:

- Mã số, họ tên (kiểu chuỗi).
- Số ngày công (số nguyên) (giá trị > 0)
- Xếp loại (char (A,B,C)). Kết quả xếp loại thi đua dựa vào qui định:
 - + Số ngày công > 26 : loại A
 - + $26 \geq \text{Số ngày công} \geq 22$: loại B
 - + Số ngày công < 22 : loại C
- Lương ngày (200.000 đồng): áp dụng cho tất cả các nhân viên → thành phần tĩnh (static).

Hãy xây dựng thêm các phương thức sau:

- Phương thức get/set cho từng thuộc tính có kiểm tra ràng buộc theo yêu cầu nêu trên.
- Nhập/xuất thông tin nhân viên.
- Phương thức tính lương (số ngày công * lương ngày).
- Phương thức tính thưởng: nếu xếp loại A thì thưởng 5% lương, loại B thưởng 2% lương và loại C không thưởng

Bài 4. Xây dựng Lớp Nước giải khát gồm các thông tin:

- Tên hàng (string)
- Đơn vị tính (string)
- Số lượng (int) (>0)
- Đơn giá (float) (>0)
- Thuế VAT (float): áp dụng chung cho các loại nước giải khát và giá trị có thể đổi theo thời gian.

Viết các hàm sau:

- Khai báo các thuộc tính cần thiết và hàm khởi tạo có tham số. Lưu ý: đơn vị tính chỉ nhận 1 trong 4 giá trị: “kết”, “thùng”, “chai”, “lon”. Nếu đơn vị tính không thuộc 1 trong 4 tham số trên thì gán đơn vị tính là “kết”.
- Hàm khởi tạo, hàm hủy
- Các hàm xử lý khác
 - + Xây dựng phương thức tính thành tiền dựa vào đơn vị tính (DVT) như sau:
 - DVT = “kết” hoặc “thùng” -> thành tiền = số lượng * đơn giá * thuế VAT
 - DVT = “chai” -> thành tiền = số lượng * (đơn giá / 20) * thuế VAT
 - DVT = “lon” -> thành tiền = số lượng * (đơn giá / 24) * thuế VAT
 - + Hàm nhập/xuất thông tin mặt hàng giải khát.
- Viết hàm main để kiểm tra các thành phần của lớp nước giải khát.

3. Bài tập về nhà

Bài 5. Xây dựng lớp Time chứa các thành phần sau:

- Các thuộc tính giờ, phút, giây (cần kiểm tra giá trị nhập vào của giờ, phút và giây: $0 \leq \text{giờ} \leq 23$; $0 \leq \text{phút} \leq 59$; $0 \leq \text{giây} \leq 59$)
- 3 hàm khởi tạo.
- Nhập/xuất thời gian theo dạng 24 giờ.
- Nhập/xuất thời gian theo dạng 12 giờ (cần có thêm thông tin AM và PM).
- Hàm kiểm tra giờ có hợp lệ không? Biết rằng giờ hợp lệ là giờ theo hệ thống đồng hồ.
- Viết hàm tăng/giảm giờ (hàm overload):
 - + void tanggio (int sogiay); → tăng giờ thông thường theo dạng giờ 24h
 - + void tanggio (int sogiay, string kieuGio); → kết quả ra dạng giờ 24 tiếng nếu kiểu giờ là 24, kết quả dạng 12h kèm AM/PM nếu kiểu giờ là 12.
 - + tương tự cho 2 hàm giảm giờ.

Bài 6. Xây dựng lớp hình chữ nhật (HCN) có các thành phần sau:

- Các thuộc tính mô tả chiều dài, chiều rộng.
- 3 hàm khởi tạo: không có và có tham số.
- Hàm tính chu vi.
- Hàm tính diện tích.
- Hàm tính đường chéo.
- Hàm nhập/xuất thông tin hình chữ nhật.

- Hàm thay đổi kích thước hình chữ nhật (overload):
 + void changeSize (int tx, int ty, int kiểu); //kích thước (chiều dài, chiều rộng) HCN sẽ tăng lên thêm tx và ty nếu kiểu == 1, ngược lại chúng sẽ giảm tx, ty nếu kiểu ==0.
 + void changeSize (HCN a, int kiểu); // kích thước HCN cộng thêm kích thước HCN a nếu kiểu==1, ngược lại chúng sẽ giảm nếu kiểu==0.
- Viết chương trình nhập vào kích thước 1 HCN, xuất ra chu vi, diện tích và đường chéo.
- Nhập vào các kích thước để thay đổi HCN

Bài 7.

Xây dựng lớp chứa thông tin của các vận động viên chạy đua trong 1 cuộc đua maraton, gồm những thuộc tính và thành phần sau:

- Mã số, họ tên, số áo, thời gian bắt đầu (kiểu Giờ của bài 5), thời gian kết thúc (kiểu Giờ của bài 5).
- Ngoài ra còn có thành tích chuẩn (1:30:00): thời gian qui định thành tích phải <= thành tích chuẩn này, nếu chạy chậm hơn thì coi như không đạt trong cuộc thi. Giá trị này áp dụng cho tất cả các vận động viên.
- 3 hàm khởi tạo.
- Hàm nhập/xuất thông tin vận động viên.
- Hàm tính thành tích (=thời gian kết thúc – thời gian bắt đầu), đổi ra dạng giờ:phút:giây
- Hàm kiểm tra thời gian nhập vào bắt đầu và kết thúc có hợp lệ không? Giờ hợp lệ là giờ thuộc hệ thống giờ trên đồng hồ (loại 24h).

Viết chương trình nhập/xuất thông tin vận động viên. Lưu ý: không nhập thành tích vì thuộc tính này lấy kết quả từ thời gian bắt đầu và kết thúc.

--HẾT--