

## Session 6

### Classes and Objects

#### Phần I - Thực hiện trong 120 phút

##### 1.1 Mục tiêu

- ✓ Hiểu vững quy tắc và cấu trúc khai báo lớp(class).
- ✓ Quy tắc khai báo tên biến, hàm.
- ✓ Khai báo biến, hàm có tham số, kiểu giá trị trả về, phạm vi truy cập cơ bản.
- ✓ Hiểu được cách khởi tạo và giá trị của biến đối tượng.
- ✓ Biết truy cập và lấy dữ liệu của biến, hàm trong class.
- ✓ Hiểu rõ khái niệm constructor của class, khai báo constructor mặc định và constructor có tham số.
- ✓ Hiểu vững phạm vi truy cập của biến, cách khai báo để tối ưu hóa mã nguồn.
- ✓ Khai báo Java-doc.

##### 1.2 Thực hiện

**Bài thực hành 1:** Cài đặt lớp SinhVien gồm các thuộc tính (rollNumber, fullName, className).

Cài đặt hai constructors, các phương thức set/get cho các thuộc tính của lớp.

Cài đặt hàm inputData() để nhập thông tin cho mỗi sinh viên.

Cài đặt hàm displayData để hiển thị thông tin của sinh viên.

Khai báo hàm main, sau đó khai báo và khởi tạo một đối tượng của lớp SinhVien, gọi hàm để nhập và hiển thị thông tin của sinh viên.

Bước 1: Tạo clas SinhVien và khai báo các thuộc tính theo yêu cầu

```
package demo.jp1.lab04;

import java.util.Scanner;

/**
 *
 * @author HAITHANH
 */
public class SinhVien {
    private String rollNumber;
    private String fullName;
    private String className;

    //Bước 2: Cài đặt 2 constructors
    /**
     * Constructor không có tham số
     */
    public SinhVien() {
    }

    /**
     * Constructor có tham số
     * @param rollNumber
     * @param fullName
     * @param className
     */
    public SinhVien(String rollNumber, String fullName, String className) {
        this.rollNumber = rollNumber;
        this.fullName = fullName;
        this.className = className;
    }

    //Bước 3: Cài đặt các hàm get/set cho các thuộc tính
    /**
     * Hàm getXXX() để lấy về giá trị của thuộc tính
     * @return
     */
    public String getRollNumber() {
        return rollNumber;
    }

    /**
     * Hàm setXXX() để gán giá trị cho thuộc tính
     * @param rollNumber
     */
    public void setRollNumber(String rollNumber) {
```

```
        this.rollNumber = rollNumber;
    }

    public String getFullName() {
        return fullName;
    }

    public void setFullName(String fullName) {
        this.fullName = fullName;
    }

    public String getClassName() {
        return className;
    }

    public void setClassName(String className) {
        this.className = className;
    }

    //Bước 4: Cài đặt hàm inputData()
    public void inputData() {
        Scanner nhap = new Scanner(System.in);
        System.out.println("Nhap vao ma sinh vien:");
        rollNumber = nhap.nextLine();
        System.out.println("Nhap vao ho ten sinh vien:");
        fullName = nhap.nextLine();
        System.out.println("Nhap vao ten lop hoc:");
        className = nhap.nextLine();
    }

    //Bước 5: Cài đặt hàm displayData()
    public void displayData() {
        System.out.println("Sinh vien:");
        System.out.println("\tMa sinh vien: "+rollNumber);
        System.out.println("\tHo ten: "+fullName);
        System.out.println("\tLop: "+className);
    }
}
```

Bước 7: Tạo class TestSinhVien, khai báo, khởi tạo và sử dụng đối tượng của lớp SinhVien

```
package demo.jp1.lab04;

/**
 *
```

```
* @author HAITHANH
*/
public class TestSinhVien {
    /**
     * Khai báo hàm main()
     * @param args
     */
    public static void main(String[] args) {
        SinhVien objSV; //Khai báo 1 đối tượng của lớp SinhVien

        objSV = new SinhVien(); //Khởi tạo đối tượng này theo constructor không có
        tham số

        objSV.inputData(); // Gọi hàm để nhập dữ liệu

        objSV.displayData(); //Gọi hàm để hiển thị dữ liệu
    }
}
```

**Bài thực hành 2:** Cài đặt lớp SanPham gồm các thuộc tính: Mã sản phẩm, tên sản phẩm, nhà sản xuất, năm sản xuất, giá bán.

Cài đặt 2 constructors, các phương thức get/set cho các thuộc tính của lớp

Cài đặt hàm input() để nhập dữ liệu, hàm display() để hiển thị dữ liệu.

Cài đặt lớp Demo trong có hàm main và thực hiện công việc:

- Khai báo 2 đối tượng của lớp SanPham
- Đối tượng thứ nhất khởi tạo qua constructor không có tham số
- Đối tượng thứ hai khởi tạo qua constructor có tham số
- Gọi hàm input() cho đối tượng thứ nhất.
- Gọi hàm display() cho cả 2 đối tượng để hiển thị thông tin ra màn hình.

Hướng dẫn:

Bước 1: Tạo class SanPham, khai báo các thuộc tính

```
package demo.jp1.lab04;

import java.util.Scanner;

/**
```

```
*
* @author HAITHANH
*/
public class SanPham {

    //khai báo các thuộc tính
    private String maSP;
    private String tenSP;
    private String nhaSX;
    private int namSX;
    private float giaBan;

    /**
     * Constructor không có tham số
     */
    public SanPham() {
    }

    //Bước 2: Cài đặt 2 constructors
    public SanPham(String maSP, String tenSP, String nhaSX, int namSX, float
giaBan) {
        this.maSP = maSP;
        this.tenSP = tenSP;
        this.nhaSX = nhaSX;
        this.namSX = namSX;
        this.giaBan = giaBan;
    }

    //Bước 3: Cài đặt các hàm get/set cho các thuộc tính của lớp
    public String getMaSP() {
        return maSP;
    }

    public void setMaSP(String maSP) {
        this.maSP = maSP;
    }

    public String getTenSP() {
        return tenSP;
    }

    public void setTenSP(String tenSP) {
        this.tenSP = tenSP;
    }

    public String getNhaSX() {
```

```
        return nhaSX;
    }

    public void setNhaSX(String nhaSX) {
        this.nhaSX = nhaSX;
    }

    public int getNamSX() {
        return namSX;
    }

    public void setNamSX(int namSX) {
        this.namSX = namSX;
    }

    public float getGiaBan() {
        return giaBan;
    }

    public void setGiaBan(float giaBan) {
        this.giaBan = giaBan;
    }

    //Bước 4: Cài đặt hàm input()
    public void input(){
        Scanner sc = new Scanner(System.in);
        System.out.println("Nhap ma san pham:");
        maSP = sc.nextLine();
        System.out.println("Nhap ten san pham:");
        tenSP = sc.nextLine();
        System.out.println("Nhap ten nha san xuất:");
        nhaSX = sc.nextLine();
        System.out.println("Nhap nam san xuất:");
        namSX = Integer.parseInt(sc.nextLine());
        System.out.println("Nhap gia ban:");
        giaBan = Float.parseFloat(sc.nextLine());
    }

    //Bước 5: Cài đặt hàm display()
    public void display(){
        System.out.println("San pham:");
        System.out.println("\tMa san pham:"+maSP);
        System.out.println("\tTen san pham:"+tenSP);
        System.out.println("\tNha san xuất:"+nhaSX);
        System.out.println("\tNam san xuất:"+namSX);
        System.out.println("\tGia ban:"+giaBan);
    }
}
```

Bước 6: Tạo class Demo và khai báo hàm main, khai báo 2 đối tượng của lớp SanPham và thực thi theo các yêu cầu của đề bài

```
package demo.jp1.lab04;

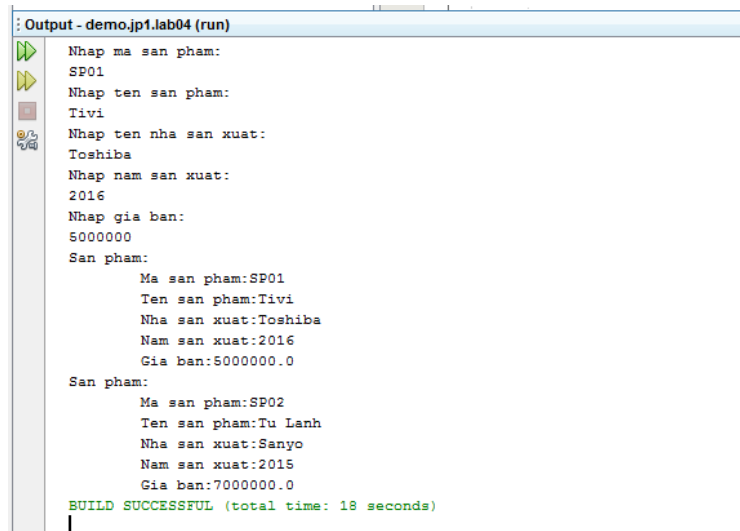
/**
 *
 * @author HAITHANH
 */
public class Demo {
    public static void main(String[] args) {
        SanPham sp1, sp2; //Khai báo 2 đối tượng của lớp SanPham

        sp1 = new SanPham(); //Khởi tạo đối tượng sp1 qua constructor không có tham số
        sp2 = new SanPham("SP02", "Tu Lanh", "Sanyo", 2015, 7000000); //Khởi tạo đối tượng sp2 qua constructor có tham số

        sp1.input(); //Gọi hàm nhập cho đối tượng sp1

        sp1.display();
        sp2.display();
    }
}
```

Kết quả thực hiện:



```
Output - demo.jp1.lab04 (run)
Nhap ma san pham:
SP01
Nhap ten san pham:
Tivi
Nhap ten nha san xuat:
Toshiba
Nhap nam san xuat:
2016
Nhap gia ban:
5000000
San pham:
    Ma san pham:SP01
    Ten san pham:Tivi
    Nha san xuat:Toshiba
    Nam san xuat:2016
    Gia ban:5000000.0
San pham:
    Ma san pham:SP02
    Ten san pham:Tu Lanh
    Nha san xuat:Sanyo
    Nam san xuat:2015
    Gia ban:7000000.0
BUILD SUCCESSFUL (total time: 18 seconds)
```

**Bài thực hành 2:** Khởi tạo lớp Customer, khai báo các thuộc tính (id, name, address, age, salary), các phương thức input(), viewData(), calSalary() có tham số truyền vào là biến bonus kiểu double hoặc int - tính lương thêm thưởng.

Bước 1: Tạo class Customer, khai báo các thuộc tính.

```
package demo.jp1.lab04;

import java.util.Scanner;

/**
 *
 * @author minhvt
 *
 */
public class Customer {

    //khai báo các thuộc tính
    private int id;
    private String name;
    private String address;
    private int age;
    private double salary;

    //Bước 2: Cài đặt 2 constructors
    //Constructor mặc định
    public Customer() {

    }

    /**
     * Constructor có 5 tham số
     */
    public Customer(int id, String name, String address, int age, double salary) {
        this.id = id;
        this.name = name;
        this.address = address;
        this.age = age;
        this.salary = salary;
    }

    //Bước 3: Cài đặt các hàm get/set cho các thuộc tính của lớp
    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 String getAddress() {
        return address;
    }

    public void setAddress(String address) {
        this.address = address;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }

    public double getSalary() {
        return salary;
    }

    public void setSalary(double salary) {
        this.salary = salary;
    }

    //Bước 4: Cài đặt phương thức input().
    void input() {
        Scanner input = new Scanner(System.in);
        System.out.println("ID: ");
        this.id = input.nextInt();
        input.nextLine(); // Giải phóng ký tự \n
        System.out.println("Tên: ");
        this.name = input.nextLine();
        System.out.println("Tuổi: ");
        this.age = input.nextInt();
        input.nextLine(); // Giải phóng ký tự \n
        System.out.println("Địa chỉ: ");
        this.address = input.nextLine();
        System.out.println("Lương: ");
```

```
this.salary = input.nextDouble();
}

//Bước 5: Cài đặt phương thức viewData()
void viewData() {
    System.out.println("Customer{" + "id=" + id
        + ", name=" + name
        + ", address=" + address
        + ", age=" + age
        + ", salary=" + salary + '}');
}

//Bước 5: Viết 2 phương thức tính lương nhận tham số truyền vào là int
//hoặc double. Đối với hàm nhận kiểu int thì giá trị trả về của hàm sẽ
//bằng salary + salary*(bonus/100), đối với hàm nhận kiểu double thì giá
//trị trả về sẽ bằng salary+bonus.
double calSalary(int bonus) {
    return salary + salary * (bonus / 100);
}

/**
 * Hàm tính toán và trả về lương thực của nhân viên (thêm thưởng)
 *
 * @param bonus: số tiền mặt thưởng
 * @since 06/2015
 * @see Lương thực lĩnh
 */
double calSalary(double bonus) {
    return salary + bonus;
}
}
```

**Bài thực hành 3:** Tạo class tên MainClass có hàm main(), trong hàm main() khởi tạo 2 đối tượng customer, truy cập vào các biến của đối tượng gán giá trị nhập từ bàn phím. Viết mã nguồn in ra thông tin của nhân viên trong hàm viewData() của class Customer. Nhập tiền thưởng của nhân viên là 15%, gọi hàm và in ra kết quả. Nhập tiền thưởng của nhân viên là 690.000đ rồi gọi hàm và in ra kết quả .

Bước 1: Tạo MainClass có hàm main() và thêm code như dưới đây

```
package demo.jp1.lab04;
```

```
/**
 *
```

```
* @author minhvt
*/
public class MainClass {
    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        //Bước 1: Khai báo và khởi tạo 2 đối tượng Customer bằng constructor mặc định.
        Customer nhanvien_1 = new Customer();
        Customer nhanvien_2 = new Customer();

        //Bước 2: Gọi hàm input() để nhập dữ liệu cho 2 đối tượng trên.
        System.out.println("Nhập dữ liệu Nhân viên 1");
        nhanvien_1.input();
        System.out.println("Nhập dữ liệu Nhân viên 2");
        nhanvien_2.input();

        //Bước 3: Viết mã để in ra thông tin của Customer. In thông tin lương.
        System.out.println("\nThông tin các nhân viên vừa nhập:");
        nhanvien_1.viewData();
        nhanvien_2.viewData();

        // Thưởng cho nhân viên 1 số tiền 15% lương cơ bản
        System.out.printf("Tiền lương nhân viên 1: %.0f", nhanvien_1.calSalary(15));

        // Thưởng cho nhân viên 2 số tiền 690.000đ
        System.out.printf("Tiền lương nhân viên 2: %.0f",
            nhanvien_2.calSalary(690000.0));
    }
}
```

## Phần II - Bài tập tự làm

### Bài 1

- Viết class HoaDonBanhMy trong đó tạo thông tin về một hóa đơn bán bánh mỳ gồm các trường dữ liệu:

*Mã hóa đơn*

*Ngày lập hóa đơn*

*Tên Khách hàng*

*Địa chỉ khách (Địa chỉ giao hàng)*

*Số lượng bánh cần giao*

*Giá bán 1 chiếc bánh*

*Tổng tiền hàng*

*Tiền khuyến mãi*

*Tổng tiền phải thanh toán*

2. Viết hàm tínhTienKmai :

Nếu mua  $\geq 10$  cái : giá mỗi cái bánh sẽ giảm đi 10%

Nếu mua  $\geq 100$  : giá tổng hóa đơn giảm đi 20% (Không giảm giá bánh nữa)

3. Tạo lớp QuanLyHoaDon

a. Nhập thông tin hóa đơn

b. In hóa đơn.

**Bài 2:** Tạo class có tên là Person và có các thuộc tính sau đây:

private String name;

private String address;

private double salary;

+ Cài đặt đầy đủ 2 constructors, các phương thức set/get cho các thuộc tính của lớp.

+ Cài đặt hàm input() để nhập dữ liệu, hàm display() để hiển thị dữ liệu

Thêm 1 class khác

+ Tạo hàm main trong đó khai báo và khởi tạo hai đối tượng của lớp Person. Gọi hàm nhập và hàm hiển thị thông tin cho hai đối tượng này.

**Bài 3:** Tạo class XeMay có các thuộc tính sau:

String ten

String nhaSanXuat

int soLuongBanhXe

+ Cài đặt đầy đủ 2 constructors, các phương thức set/get cho các thuộc tính của lớp.

+ Cài đặt hàm input() để nhập dữ liệu, hàm display() để hiển thị dữ liệu

Sửa lại

+ Tạo hàm main trong đó khai báo và khởi tạo ba đối tượng của lớp Person. Gọi hàm nhập và hàm hiển thị thông tin cho ba đối tượng này.