

## Ví dụ minh họa lớp SinhVien

Xây dựng lớp **SinhVien** với các yêu cầu:

- ✚ **Thuộc tính:** Mã sinh viên (String), Họ tên (String), tuổi (int), điểm trung bình (double), giới tính (boolean).
- ✚ **Phương thức:** khởi tạo, nhập, xuất, getter(phương thức lấy), setter(phương thức thiết lập).

Thực hiện các công việc sau:

- ✚ Nhập danh sách sinh viên
- ✚ Xuất danh sách sinh viên
- ✚ Xuất ra danh sách các sinh viên có tuổi > 25
- ✚ Xuất ra danh sách các sinh viên nữ
- ✚ Sắp xếp các sinh viên theo thứ tự điểm trung bình không giảm.

## Hướng dẫn chi tiết

```
public class SinhVien {
    private String maSv;
    private String hoTen;
    private int tuoi;
    private double diemTb;
    private boolean gioiTinh;

    public SinhVien() {
        // TODO Auto-generated constructor stub
    }

    public SinhVien(String maSv, String hoTen, int tuoi, double diemTb, boolean gioiTinh) {
        this.maSv = maSv;
        this.hoTen = hoTen;
        this.tuoi = tuoi;
        this.diemTb = diemTb;
        this.gioiTinh = gioiTinh;
    }

    public String getMaSv() {
        return maSv;
    }
}
```

Figure 1: Khai báo lớp, thuộc tính và phương thức khởi tạo.

```
public String getMaSv() {  
    return maSv;  
}  
  
public void setMaSv(String maSv) {  
    this.maSv = maSv;  
}  
  
public String getHoTen() {  
    return hoTen;  
}  
  
public void setHoTen(String hoTen) {  
    this.hoTen = hoTen;  
}  
  
public int getTuoi() {  
    return tuoi;  
}  
  
public void setTuoi(int tuoi) {  
    this.tuoi = tuoi;  
}
```

Figure 2: Phương thức get, set cho các thuộc tính maSv, hoTen, tuoi.

```

public double getDiemTb() {
    return diemTb;
}

public void setDiemTb(double diemTb) {
    this.diemTb = diemTb;
}

public boolean getGioiTinh() {
    return gioiTinh;
}

public void setGioiTinh(boolean gioiTinh) {
    this.gioiTinh = gioiTinh;
}

```

Figure 3: Phương thức get, set cho các thuộc tính diemTb, gioiTinh.

```

public void nhap() {
    Scanner sc = new Scanner(System.in);
    System.out.print("Nhap vao ma sv: ");
    this.maSv = sc.nextLine();
    System.out.print("Nhap vao ho ten sv: ");
    this.hoTen = sc.nextLine();
    System.out.print("Nhap vao tuoi sv: ");
    this.tuoi = sc.nextInt();
    System.out.print("Nhap vao diem tb sv: ");
    this.diemTb = sc.nextDouble();
    System.out.print("Nhap vao gioi tinh sv: ");
    this.gioiTinh = sc.nextBoolean();
}

```

Figure 4: Phương thức nhập thông tin cho sinh viên.

```

@Override
public String toString() {
    return "SinhVien [maSv=" + this.maSv + ", hoTen=" + this.hoTen + ", tuoi=" + this.tuoi +
        ", diemTb=" + this.diemTb + ", gioiTinh=" + this.gioiTinh + "]";
}

public void xuat() {
    System.out.println(this.toString());
}

public void copy(SinhVien sv) {
    this.maSv = sv.getMaSv();
    this.hoTen = sv.getHoTen();
    this.tuoi = sv.getTuoi();
    this.diemTb = sv.getDiemTb();
    this.gioiTinh = sv.getGioiTinh();
}

```

Figure 5: Phương thức in thông tin sinh viên ra màn hình và phương thức copy để sao chép dữ liệu của 2 sinh viên cho nhau.

```

public static void main(String[] args) {
    int n; //So lượng sinh viên
    Scanner sc = new Scanner(System.in);
    SinhVien sv;
    ArrayList<SinhVien> listSv = new ArrayList<SinhVien>();

    System.out.println("Nhap vao so luong sinh vien: ");
    n = sc.nextInt();

    for(int i = 0; i < n; i++)
    {
        System.out.println("----- Sinh vien " + i + "-----");
        sv = new SinhVien();
        sv.nhap();
        listSv.add(sv);
    }

    for(int i = 0; i < listSv.size(); i++)
    {
        listSv.get(i).xuat();
    }
}

```

Figure 6: Hàm main của lớp SinhVien để chạy thử chương trình. Thực hiện công việc nhập danh sách sinh viên và xuất danh sách vừa nhập ra màn hình.

```
<terminated> SinhVien [Java Application] C:\Program Files\Java\jdk1.8.0_261\bin\javaw.exe (Nov 13, 2020, 10:06:51 AM - 10:07:30 AM)
Nhap vao so luong sinh vien: 2
----- Sinh vien 0-----
Nhap vao ma sv: sv001
Nhap vao ho ten sv: nguyen van a
Nhap vao tuoi sv: 23
Nhap vao diem tb sv: 8.3
Nhap vao gioi tinh sv: true
----- Sinh vien 1-----
Nhap vao ma sv: sv002
Nhap vao ho ten sv: vu thi trang
Nhap vao tuoi sv: 26
Nhap vao diem tb sv: 9.5
Nhap vao gioi tinh sv: false
SinhVien [maSv=sv001, hoTen=nguyen van a, tuoi=23, diemTb=8.3, gioiTinh=true]
SinhVien [maSv=sv002, hoTen=vu thi trang, tuoi=26, diemTb=9.5, gioiTinh=false]
```

Figure 7: Kết quả của chương trình sau khi chạy hàm main.

```
System.out.println("----- Cac Sinh vien co tuoi > 25 -----");
for(int i = 0; i < listSv.size(); i++)
{
    if(listSv.get(i).getTuoi() > 25)
    {
        listSv.get(i).xuat();
    }
}
```

Figure 8: Thực hiện in ra màn hình các sinh viên có tuổi lớn hơn 25.

```
System.out.println("----- Cac Sinh vien co gioi tinh la nu -----");
for(int i = 0; i < listSv.size(); i++)
{
    if(listSv.get(i).getGioiTinh() == false)
    {
        listSv.get(i).xuat();
    }
}
```

Figure 9: Thực hiện in ra màn hình các sinh viên có giới tính là nữ.

```

System.out.println("----- Sinh vien theo tt diem tb khong giam -----");
for(int i = 0; i < listSv.size() - 1; i++)
{
    for(int j = i + 1; j < listSv.size(); j++)
    {
        if(listSv.get(i).getDiemTb() > listSv.get(j).getDiemTb())
        {
            SinhVien trungGian = new SinhVien();
            trungGian.copy(listSv.get(i));
            listSv.get(i).copy(listSv.get(j));
            listSv.get(j).copy(trungGian);
        }
    }
}
for(int i = 0; i < listSv.size(); i++)
{
    listSv.get(i).xuat();
}

```

Figure 10: Thực hiện sắp xếp các sinh viên theo thứ tự điểm tb không giảm và in ra màn hình.