

## LAB 2: LỚP NÂNG CAO

### MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng

- ✓ Sử dụng static, final để định nghĩa các hằng số
- ✓ Sử dụng kỹ thuật lớp nặc danh để tạo đối tượng
- ✓ Thiết kế giao diện
- ✓ Xử lý sự kiện

### PHẦN I

#### BÀI 1 (2 ĐIỂM)

Tạo lớp Weekday chứa các hằng số là các ngày trong tuần theo hướng dẫn sau:

```
public class Weekday{  
    public static final int Sunday = 0;  
    public static final int Monday = 1;  
    public static final int Tuesday = 2;  
    public static final int Wednesday = 3;  
    public static final int Thursday = 4;  
    public static final int Friday = 5;  
    public static final int Saturday = 6;  
}
```

Tạo lớp CheckDay chứa phương thức main() cho phép nhập một số từ 0 đến 6 và đưa ra thông báo:

- ✓ “Ngày cuối tuần” nếu là số 0 hoặc số 1
- ✓ “Ngày trong tuần” nếu là các số còn lại

Hướng dẫn:

```
int day = scanner.nextInt();  
if(day == Weekday.Sunday || day == Weekday.Saturday){
```

```
        System.out.println("Ngày cuối tuần");
    }
    else{
        ...
    }
}
```

## **BÀI 2 (2 ĐIỂM)**

Tạo một lớp NhanVien như hướng dẫn sau

```
public abstract class Employee {
    private String fullname;
    public Employee(String fullname){
        this.fullname = fullname;
    }
    public abstract double getSalary();
    public void print(){
        System.out.println("Fullname: " + this.fullname);
        System.out.println("Salary: " + this.getSalary());
    }
}
```

Tạo lớp EmployeeManager chứa phương thức main() và sử dụng kỹ thuật lớp nặc danh để tạo 2 đối tượng e1 và e2 từ lớp Employee và override phương thức getSalary() để tính lương cho các đối tượng e1 và e2.

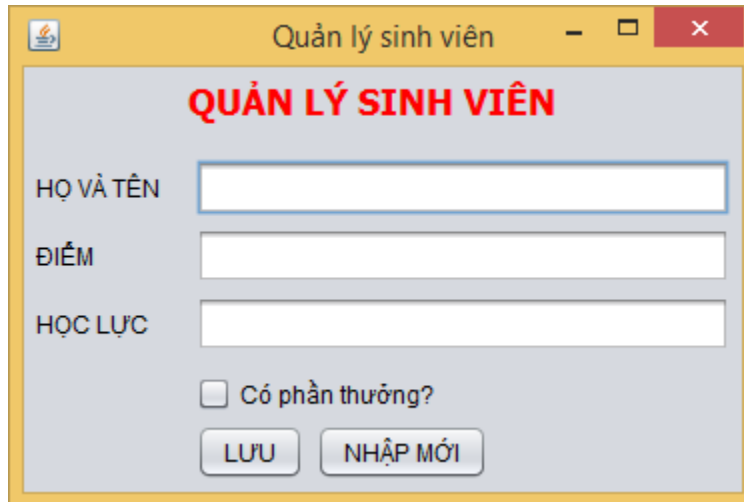
Hướng dẫn:

```
Employee e1 = new Employee(null) {
    @Override
    public double getSalary() {
        return 7000000;
    }
};
e1.print();
```

## PHẦN II

### BÀI 3 (2 ĐIỂM)

Tạo một cửa sổ và thiết kế giao diện quản lý sinh viên như hình sau



Đặt tên các thành phần giao diện theo qui ước: lblXxx, txtXxx, chkXxx, btnXxx

Trong đó ô nhập học lực để hiển thị học lực được tính toán dựa vào số điểm nên không cho phép nhập vào.

Bổ sung lệnh setLocationRelativeTo(null); vào cuối hàm tạo của cửa sổ để làm cho cửa sổ luôn hiển thị giữa màn hình:

```
public StudentUI() {
    initComponents();
    setLocationRelativeTo(null);
}
```

### BÀI 4 (2 ĐIỂM)

Tạo lớp Student để quản lý thông tin sinh viên như sau:

```
public class Student {
    public String name;
    public double marks;
```

```
public String getGrade() {
    if (this.marks < 3) {
        return "Kém";
    }
    if (this.marks < 5) {
        return "Yếu";
    }
    if (this.marks < 6.5) {
        return "Trung bình";
    }
    if (this.marks < 7.5) {
        return "Khá";
    }
    if (this.marks < 9) {
        return "Giỏi";
    }
    return "Xuất sắc";
}

public boolean isBonus() {
    return this.marks >= 7.5;
}
}
```

Viết mã xử lý sự kiện khi click chuột vào các nút

[NHẬP MỚI]: xóa trắng các ô nhập trên form và bỏ check [Có phần thưởng] bằng cách sử dụng các phương thức sau.

```
txtXxx.setText("");
chk.setSelected(false);
```

[LƯU]: Tạo đối tượng sinh viên và hiển thị học lực đồng thời check lên [Có phần thưởng] nếu sinh viên có phần thưởng

```
Student sv = new Student();  
sv.name = txtHoTen.getText();  
sv.marks = Double.parseDouble(txtDiem.getText());  
txtHocLuc.setText(sv.getGrade());  
chkThuong.setSelected(sv.isBonus());  
JOptionPane.showMessageDialog(this, "Đã hoàn tất, chúc mừng bạn!");
```

**BÀI 5 (2 ĐIỂM)**

Giảng viên cho thêm