

JORDY MEJIA EXAMEN

Falto acabar la clase Laboratorio y el Main ☹ y no alcance a ejecutarlo ;(

Main

```
import javax.swing.JOptionPane;

public class Main {

    public static void main(String[] args) {
        int nLabos =
Integer.parseInt(JOptionPane.showInputDialog("Ingrese el numero de
laboratorios: "));
        String tipoLab = JOptionPane.showInputDialog("Ingrese el tipo de
los laboratorios: ");
        int capacidad =
Integer.parseInt(JOptionPane.showInputDialog("Ingrese la capacidad del
laboratorio: "));
        int nEquipos =
Integer.parseInt(JOptionPane.showInputDialog("Ingrese el numero de equipos de
cada laboratorio: "));
        Laboratorio L1 = new Laboratorio(nLabos, tipoLab, capacidad,
nEquipos);

        L1.establecerEquipos();
        int nLab = Integer.parseInt(JOptionPane.showInputDialog("Ingrese
el numero de laboratorio al que le quiera establecer los equipos: "));
        L1.establecerEquipos(nLab);
        L1.mostrarEquipos(nLab);

    }

}
```

Clase Laboratorio //Creaba las clases

```
import javax.swing.JOptionPane;

public class Laboratorio implements LabMets{
    private String horario[][][];
    private int nLabs;
    private String tipo;
    private String nombreL[];
    private int capacidad;
    private int nEquipos;
    String equipos[][];
    public Laboratorio(int nLabs, String tipo, int capacidad, int
nEquipos) {
        super();
        this.nLabs = nLabs;
        this.tipo = tipo;
        this.capacidad = capacidad;
        this.nEquipos = nEquipos;
        nombreL = new String[nLabs];
    }
}
```

```

        equipos = new String[nLabs][nEquipos];
    }
    public int getCapacidad() {
        return capacidad;
    }
    public void setCapacidad(int capacidad) {
        this.capacidad = capacidad;
    }
    @Override
    public void establecerEquipos() {
        for(int i = 0; i<nLabs;i++) {
            for(int j = 0; j<nEquipos;i++) {
                equipos[i][j] = null;
            }
        }
    }
    public void establecerEquipos(int lab) {
        lab = lab-1;
        if(lab >= 1 && lab < nLabs) {
            for(int i = 0; i<nEquipos; i++) {
                equipos[lab][i] =
JOptionPane.showInputDialog("Ingrese el equipo " + (i+1) + "");
            }
        } else {
            System.out.print("\nIngrese una laboratorio existente");
        }
    }
    @Override
    public void establecerHorarios(int lab, int dia, int hora) {
        if(lab >= 1 && lab < nLabs) {
            if(dia >= 1 && dia <= 5) {
                if(hora >= 0 && hora <= 5) {
                    horario[lab][dia][hora] = "";
                }
            }
        } else {
            System.out.print("\nIngrese una laboratorio existente");
        }
    }
    @Override
    public void mostrarEquipos(int lab) {
        lab = lab-1;
        if(lab >= 1 && lab < nLabs) {
            System.out.println("\n\tEquipos del laboratorio");
            for(int i = 0; i<nEquipos; i++) {
                System.out.print(equipos[lab][i] + " | ");
            }
        } else {
            System.out.print("\nIngrese una laboratorio existente");
        }
    }
    @Override
    public void establecerNombres() {
        for(int i = 0; i<nLabs;i++) {
            nombreL[i] = null;
        }
    }
    public void establecerNombres(int nLab) {
        nLab = nLab-1;
    }

```

```

        if(nLab >= 1 && nLab < nLabs) {
            // nombreL[nLab]= JOptionPane.showInputDialog("Ingrese el
equipo " + (i+1) + "");
        } else {
            System.out.print("\nIngrese una laboratorio existente");
        }
    }
    @Override
    public void mostrarNombres() {
        System.out.print("\n\tLaboratorios");
        for(int i = 0; i < nombreL.length; i++) {
            System.out.print(
                "\n" + (i+1) + ". " + nombreL[i]
            );
        }
    }

    @Override
    public void mostrarHorarios() {
        // TODO Auto-generated method stub
    }
    //horarios= new String[][];
}

```

Interface LabMets

```

public interface LabMets {
    public void establecerEquipos();
    public void establecerEquipos(int lab);
    public void mostrarEquipos(int lab);
    public void establecerNombres();
    public void mostrarNombres();
    public void establecerHorarios(int lab, int dia, int hora);
    public void mostrarHorarios();
}

```

Clase Personal

```

public class Personal {
    private String nombre;
    private int edad;
    public Personal(String nombre, int edad) {
        super();
        this.nombre = nombre;
        this.edad = edad;
    }
    public String getNombre() {
        return nombre;
    }
    public void setNombre(String nombre) {
        this.nombre = nombre;
    }
}

```

```

        public int getEdad() {
            return edad;
        }
        public void setEdad(int edad) {
            this.edad = edad;
        }
    }
}

```

Clase Profesor

```

public class Profesor extends Personal implements Datos{
    private String materia, titulo;

    public Profesor(String nombre, int edad, String materia, String
    titulo) {
        super(nombre, edad);
        this.materia = materia;
        this.titulo = titulo;
    }

    public String getMateria() {
        return materia;
    }

    public void setMateria(String materia) {
        this.materia = materia;
    }

    public String getTitulo() {
        return titulo;
    }

    public void setTitulo(String titulo) {
        this.titulo = titulo;
    }

    @Override
    public void mostrarDatos() {
        System.out.print(
            "\n\tDatos - Profesor" +
            "\nNombre: " + getNombre() +
            "\nEdad: " + getEdad() +
            "\nMateria: " + this.materia +
            "\nTitulo: " + this.titulo
        );
    }
}

```

Clase Encargado

```

public class Encargado extends Personal implements Datos{
    private String laboratorio;

    public Encargado(String nombre, int edad, String laboratorio) {
        super(nombre, edad);
        this.laboratorio = laboratorio;
    }
}

```

```

    }

    @Override
    public void mostrarDatos() {
        System.out.print(
            "\n\tDatos - Encargado" +
            "\nNombre:      " + getNombre() +
            "\nEdad:          " + getEdad() +
            "\nLaboratorio:  " + this.laboratorio
        );
    }
}

```

Clase estudiante

```

public class Estudiantes extends Personal implements Datos{
    private String carrera;

    public Estudiantes(String nombre, int edad, String carrera) {
        super(nombre, edad);
        this.carrera = carrera;
    }

    public String getCarrera() {
        return carrera;
    }

    public void setCarrera(String carrera) {
        this.carrera = carrera;
    }

    @Override
    public void mostrarDatos() {
        System.out.print(
            "\n\tDatos - Estudiantes" +
            "\nCarrera:          " + getCarrera() +
            "\nEdad promedio:  " + getEdad()
        );
    }
}

```

Interface Datos

```

public interface Datos {
    public void mostrarDatos();
}

```