

EXAMEN FINAL DE SIS103

Nombre: Gonzales Suyo Franz Reinaldo

C.U. 111-500

Carrera: Ing. Ciencias de la Computación

1.-Implementar una jerarquía Empleado de cualquier tipo de empresa que sea familiar; debe tener dos niveles: herencia de miembros dato, así como métodos y redefinición de métodos. Los métodos deben calcular salarios, despidos, promociones, altas, jubilaciones, etcétera; escribir un programa que cree objetos de tipo empleado y realice operaciones polimórficas.

Clase Empleado

```
public abstract class Empleado {

    private String nombre;
    private int edad;
    private double salario;

    public Empleado(String nombre, int edad, double salario) {
        this.nombre = nombre;
        this.edad = edad;
        this.salario = salario;
    }

    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;
    }

    public double getSalario() {
        return salario;
    }

    public void setSalario(double salario) {
        this.salario = salario;
    }
}
```

```

    }

    public void calcularSalario() {

        System.out.println("El salario de " + nombre + " es: " + salario);
    }

    public void despedir() {
        System.out.println("El empleado " + nombre + " ha sido despedido.");
    }

    public void promocion(double incremento) {
        salario += incremento;
        System.out.println("El empleado " + nombre + " ha sido
promovido nuevo salario de: " + salario);
    }

    public void alta() {
        System.out.println("El empleado " + nombre + " ha sido dado de alta
en la empresa.");
    }

    public void jubilar() {
        System.out.println("El empleado " + nombre + " se ha jubilado.");
    }
}

```

Clase EmpleadoMedioTiempo

```

public class EmpleadoMedioTiempo extends Empleado {
    public EmpleadoMedioTiempo(String nombre, int edad, double salario) {
        super(nombre, edad, salario);
    }

    @Override
    public void calcularSalario() {
        // Podemos calcular el salario para empleados de medio tiempo
        double salario = getSalario();
        double salarioProporcional = salario / 2;
        System.out.println("El salario de " + getNombre() + " de medio
tiempo es: " + salarioProporcional);
    }
}

```

Clase EmpleadoTiempoCompleto

```

public class EmpleadoTiempoCompleto extends Empleado {

```

```

    public EmpleadoTiempoCompleto(String nombre, int edad, double salario) {
        super(nombre, edad, salario);
    }

    @Override
    public void calcularSalario() {
        // Podemos calcular el salario para empleados de tiempo completo
        double salario = getSalario();
        double salarioBonos = salario + 500;
        System.out.println("El salario del empleado " + getNombre() + " de
tiempo completo es: " + salarioBonos);
    }
}

```

Clase Empresa

```

public class Empresa {
    public static void main(String[] args) {

        Empleado empleado1 = new EmpleadoTiempoCompleto("Ramiro", 30, 2200);
        Empleado empleado2 = new EmpleadoMedioTiempo("Carlos", 25, 5000);

        // Damos de alta a los empleados
        empleado1.alta();
        empleado2.alta();

        // Calculamos los salarios de los empleados
        empleado1.calcularSalario();
        empleado2.calcularSalario();

        // Promociones de los empleados
        empleado1.promocion(300);
        empleado2.promocion(150);

        // Calculamos los salarios de los empleados
        empleado1.calcularSalario();
        empleado2.calcularSalario();

        // Despedimos y jubilamos a los empleados
        empleado1.despedir();
        empleado2.jubilacion();
    }
}

```

Prueba de ejecución

```
PS C:\GONZALES\SIS103\Parciales\ExamenFinal> javac GestionAlumnos.java Alumno.java Asignatura.java Profesor.java
PS C:\GONZALES\SIS103\Parciales\ExamenFinal> java Empresa
El empleado Ramiro ha sido dado de alta en la empresa.
El empleado Carlos ha sido dado de alta en la empresa.
El salario del empleado Ramiro de tiempo completo es: 2700.0
El salario de Carlos de medio tiempo es: 2500.0
El empleado Ramiro ha sido promovido nuevo salario de: 2500.0
El empleado Carlos ha sido promovido nuevo salario de: 5150.0
El salario del empleado Ramiro de tiempo completo es: 3000.0
El salario de Carlos de medio tiempo es: 2575.0
El empleado Ramiro ha sido despedido.
El empleado Carlos se ha jubilado.
PS C:\GONZALES\SIS103\Parciales\ExamenFinal>
```

2.-Realizar una aplicación para que cada profesor de la universidad gestione las fichas de sus alumnos; un profesor imparte una asignatura con distintos grupos de alumnos; éstos pueden ser presenciales o a distancia. Al comenzar el curso, se entrega al profesor un listado con los alumnos por cada asignatura; escribir un programa de tal forma que el listado de alumnos se introduzca por teclado y se den de alta calificaciones de exámenes y prácticas realizadas; se podrán obtener listados de calificaciones una vez realizados los exámenes y porcentajes de aprobados. En el diseño de clases encontrar operaciones polimórficas.

Clase Alumno

```
public class Alumno {
    private String nombre;
    private String tipoAlumno; // Este es de tipoAlumno presencial o a
    distancia
    private double[] calificaciones;
    private int nroCalificaciones;

    public Alumno(String nombre, String tipoAlumno, int maxCalificaciones) {
        this.nombre = nombre;
        this.tipoAlumno = tipoAlumno;
        this.calificaciones = new double[maxCalificaciones];
        this.nroCalificaciones = 0;
    }

    public String getNombre() {
        return nombre;
    }

    public String getTipoAlumno() {
        return tipoAlumno;
    }

    public void agregarCalificacion(double calificacion) {

        if (nroCalificaciones < calificaciones.length) {
            calificaciones[nroCalificaciones] = calificacion;
            nroCalificaciones++;
        }
    }
}
```

```

        } else {
            System.out.println("No se pueden agregar más calificaciones para
el alumno: " + nombre);
        }
    }

    public double calcularPromedio() {
        if (nroCalificaciones == 0) {
            return 0;
        }
        double suma = 0;
        for (int i = 0; i < nroCalificaciones; i++) {
            suma += calificaciones[i];
        }
        double promedio = suma / nroCalificaciones;
        return promedio;
    }

    public void listarCalificaciones() {
        System.out.print("Calificaciones de los Alumnos: ");
        for (int i = 0; i < nroCalificaciones; i++) {
            System.out.print(calificaciones[i] + " ");
        }
        System.out.println();
    }
}

```

Clase Asignatura

```

public class Asignatura {

    private String nombre;
    private Alumno[] alumnos;
    private int numeroAlumnos;

    public Asignatura(String nombre, int maxAlumnos) {
        this.nombre = nombre;
        this.alumnos = new Alumno[maxAlumnos];
        this.numeroAlumnos = 0;
    }

    public String getNombre() {
        return nombre;
    }

    public void agregarAlumno(Alumno alumno) {

        if (numeroAlumnos < alumnos.length) {

```

```

        alumnos[numeroAlumnos] = alumno;
        numeroAlumnos++;
    } else {
        System.out.println("No se pueden agregar más alumnos a la
asignatura " + nombre);
    }
}

public void listarCalificaciones() {
    System.out.println("Listado de calificaciones para la asignatura: "
+ nombre);
    for (int i = 0; i < numeroAlumnos; i++) {
        Alumno alumno = alumnos[i];

        System.out.print(alumno.getNombre() + " (" +
alumno.getTipoAlumno() + "): ");
        alumno.listarCalificaciones();
        System.out.println("Promedio: " + alumno.calcularPromedio());
    }
}

public void porcentajeAprobados(double notaAprobatoria) {
    int aprobados = 0;
    for (int i = 0; i < numeroAlumnos; i++) {
        if (alumnos[i].calcularPromedio() >= notaAprobatoria) {
            aprobados++;
        }
    }
    double porcentaje = ((double) aprobados / numeroAlumnos) * 100;
    System.out.println("Porcentaje de aprobados en " + nombre + ": " +
porcentaje + "%");
}
}

```

Clase Profesor

```

public class Profesor {
    private String nombre;
    private Asignatura[] asignaturas;
    private int numeroAsignaturas;

    public Profesor(String nombre, int maxAsignaturas) {
        this.nombre = nombre;
        this.asignaturas = new Asignatura[maxAsignaturas];
        this.numeroAsignaturas = 0;
    }

    public void agregarAsignatura(Asignatura asignatura) {

```

```

        if (numeroAsignaturas < asignaturas.length) {
            asignaturas[numeroAsignaturas] = asignatura;
            numeroAsignaturas++;
        } else {
            System.out.println("No se pueden agregar más asignaturas para el
profesor " + nombre);
        }
    }

    public void listarCalificaciones() {
        System.out.println("Listado de calificaciones para el profesor: " +
nombre);
        for (int i = 0; i < numeroAsignaturas; i++) {
            asignaturas[i].listarCalificaciones();
        }
    }

    public void porcentajeAprobados(String nombreAsignatura, double
notaAprobatoria) {
        for (int i = 0; i < numeroAsignaturas; i++) {

            if (asignaturas[i].getNombre().equals(nombreAsignatura)) {

                asignaturas[i].porcentajeAprobados(notaAprobatoria);
                return;
            }
        }
        System.out.println("Asignatura no encontrada.");
    }
}

```

Clase GestionAlumnos

```

import java.util.Scanner;

public class GestionAlumnos {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Ingrese el nombre del profesor: ");
        String nombreProfesor = scanner.nextLine();
        Profesor profesor = new Profesor(nombreProfesor, 10);

        System.out.println("Ingrese el nombre de la asignatura: ");
        String nombreAsignatura = scanner.nextLine();
        Asignatura asignatura = new Asignatura(nombreAsignatura, 30);
        profesor.agregarAsignatura(asignatura);
    }
}

```

```

        System.out.println("Ingrese el número de alumnos:");
        int numAlumnos = Integer.parseInt(scanner.nextLine());

        // Recorremos a ingresar todos los alumnos
        for (int i = 0; i < numAlumnos; i++) {

            System.out.println("Ingrese el nombre del alumno " + (i + 1) +
":");

            String nombreAlumno = scanner.nextLine();

            System.out.println("El alumno es presencial o a distancia?
(P/D):");

            String tipoAlumno = scanner.nextLine();

            // String tipo = "";
            // if(tipoAlumno == "P"){
            //     tipo = "Presencial";
            // }else{
            //     tipo = "Distancia";
            // }

            // Alumno alumno = new Alumno(nombreAlumno, tipo, 10);

            Alumno alumno = new Alumno(nombreAlumno,
tipoAlumno.equalsIgnoreCase("P") ? "Presencial" : "Distancia", 10);

            asignatura.agregarAlumno(alumno);

            System.out.println("Ingrese el número de calificaciones del
alumno " + nombreAlumno + ":");
            int numCalificaciones = Integer.parseInt(scanner.nextLine());

            for (int j = 0; j < numCalificaciones; j++) {
                System.out.println("Ingrese la calificación " + (j + 1) +
":");

                double calificacion =
Double.parseDouble(scanner.nextLine());
                alumno.agregarCalificacion(calificacion);
            }
        }

        profesor.listarCalificaciones();

        System.out.println("Ingrese la nota de aprobación: ");
        double notaAprobatoria = Double.parseDouble(scanner.nextLine());

        profesor.porcentajeAprobados(nombreAsignatura, notaAprobatoria);

        scanner.close();
    }
}

```


Prueba de ejecución

```
PS C:\GONZALES\SIS103\Parciales\ExamenFinal>
PS C:\GONZALES\SIS103\Parciales\ExamenFinal> javac GestionAlumnos.java Alumno.java Asignatura.java Profesor.java
PS C:\GONZALES\SIS103\Parciales\ExamenFinal> java GestionAlumnos
Ingrese el nombre del profesor:
Carlos
Ingrese el nombre de la asignatura:
Matematicas
Ingrese el número de alumnos:
2
Ingrese el nombre del alumno 1:
mario
El alumno es presencial o a distancia? (P/D):
D
Ingrese el número de calificaciones del alumno mario:
1
Ingrese la calificación 1:
67
Ingrese el nombre del alumno 2:
Juan
El alumno es presencial o a distancia? (P/D):
P
Ingrese el número de calificaciones del alumno Juan:
1
Ingrese la calificación 1:
78
Listado de calificaciones para el profesor: Carlos
Listado de calificaciones para la asignatura: Matematicas
mario (Distancia): Calificaciones de los Alumnos: 67.0
Promedio: 67.0
Juan (Presencial): Calificaciones de los Alumnos: 78.0
Promedio: 78.0
Ingrese la nota de aprobación:
51
Porcentaje de aprobados en Matematicas: 100.0%
PS C:\GONZALES\SIS103\Parciales\ExamenFinal> █
```