

UNIVERSIDAD NACIONAL SAN CRISTOBAL DE HUAMANGA

FACULTAD DE INGENIERIA DE MINAS, GEOLOGIA Y CIVIL

ESCUELA PROFESIONAL DE INGENIERIA DE SISTEMAS



Real, Pontificia y Nacional
1677

Laboratorio 6

DOCENTE : M.sC. Fredy Barrientos Espillco

CURSO : Programación Orientada a Objetos

SIGLA : IS - 142

TEMA : INFORME N° 05

SEMESTRE : 2019 - II

ESTUDIANTE : ORÉ CHÁVEZ, Edwin

CODIGO : 27075141

AYACUCHO - PERÚ

(2019)

1. Introducción

La programación orientada a objetos es una “filosofía”, un modelo de programación, con su teoría y su metodología, que conviene conocer y estudiar, antes de nada. Un lenguaje orientado a objetos es un lenguaje de programación que permite el diseño de aplicaciones orientadas a objetos. Dicho esto, lo normal es que toda persona que vaya a desarrollar aplicaciones orientadas a objetos aprenda primero la “filosofía” (o adquiera la forma de pensar) y después el lenguaje, porque “filosofía” sólo hay una y lenguajes muchos. En este documento veremos brevemente los conceptos básicos de la programación orientada a objetos desde un punto de vista global, sin particularizar para ningún lenguaje de programación específico.

2. Objetivos

2.1. Objetivo General

Definir Clases y crear objetos, métodos y constructores en Java.

2.2. Objetivos Específicos

- Definir las clases con sus respectivos atributos y métodos en Java
- Crear objetos de la clase en Java
- Crear métodos en Java
- Crear constructores en Java

3. Requisitos

Para el desarrollo del siguiente laboratorio necesitamos tener conocimientos previos de:

- Manejo de variables, cadenas y operadores en Java.
- Manejo de estructuras de control en Java.

4. Desarrolle La Tarea Académica (TA05) consiste en los siguientes ejercicios:

4.1 Implemente una clase de nombre Estudiante con los siguientes atributos privados: código, apellidos, especialidad, nota1, nota2, nota3, nota4. Considere un método para calcular el promedio, sin considerar la nota menor.

Solución

En el programa Netbeans creamos un proyecto de nombre TA05_Edwin_Ore y su respectivo paquete TA05 y dentro de ella creamos la clase Estudiante y asignamos el código correspondiente de acuerdo al análisis del problema. Que a continuación se muestra en la captura de pantalla.

Los comentarios señalados dentro de la codificación dan detalle de cada proceso

```
package TA05;

import java.util.Scanner;

/**
 *
 * @author luudwin
 */
public class Estudiante {
    // declaramos las variables codigo, nombre y especialidad como string privado,

    private String codigo;
    private String nombre;
    private String especialidad;

    // utilizando metodos get y set para cada variable privado
    public String getCodigo() {
        return codigo;
    }
    public void setCodigo(String codigo) {
        this.codigo = codigo;
    }
    public String getNombre() {
        return nombre;
    }
    public Estudiante(String codigo, String nombre, String especialidad) {
        super();
        this.codigo = codigo;
        this.nombre = nombre;
        this.especialidad = especialidad;
    }
    public void setNombre(String nombre) {
        this.nombre = nombre;
    }
    public String getEspecialidad() {
        return especialidad;
    }
    public void setEspecialidad(String especialidad) {
        this.especialidad = especialidad;
    }
    // creamos el metodo main

    public static void main(String[] args) {

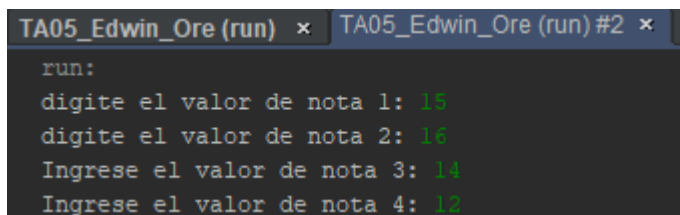
        // Importamos la clase Scanner para leer valores por entrada
        // Creamos un objeto entrada para almacenar los datos ingresados
        Scanner entrada = new Scanner(System.in);
        /* declarando nota1, nota2 ,nota3, nota4 de tipo double
        asi como la variable promedio y variable menor nota
        */
        double menor, nota1, nota2, nota3, nota4;
        double promedio;
        // pedimos al usuario ingresar las cuatro notas por teclado
        System.out.print("digite el valor de nota 1: ");
        nota1 = entrada.nextDouble();
        System.out.print("digite el valor de nota 2: ");
        nota2 = entrada.nextDouble();
        System.out.print("Ingrese el valor de nota 3: ");
        nota3 = entrada.nextDouble();
        System.out.print("Ingrese el valor de nota 4: ");
        nota4 = entrada.nextDouble();
```

```

// Utilizamos estructuras de control "if" para conparar cada nota ingresada
menor=notal;
if(menor>nota2)
    menor=nota2;
if(menor>nota3)
    menor=nota3;
if(menor>nota4)
    menor=nota4;
/* La siguiente instuccion es para calcular el promedio de las notas
sin considerar la menor nota.
*/
promedio=(notal+nota2+nota3+nota4-menor)/3;
// finalmente mostra mos el resultado en pantalla
System.out.println("La menor nota es : " + menor);
System.out.println("El promedio es: : " + promedio);
}
}

```

Al momento de ejecutar, este programa pedirá digitar las cuatro notas del estudiante las cuales serán almacenadas en un objeto de nombre “Entrada” que previamente importó la clase Scanner

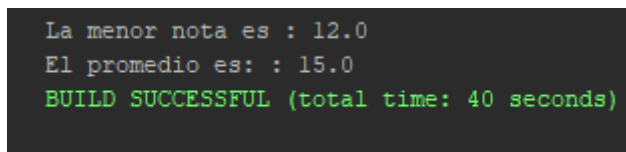


```

TA05_Edwin_Ore (run) x TA05_Edwin_Ore (run) #2 x
run:
digite el valor de nota 1: 15
digite el valor de nota 2: 16
Ingrese el valor de nota 3: 14
Ingrese el valor de nota 4: 12

```

Al momento de ejecutar nos mostrara por pantalla la nota menor y el promedio generado (sin considerar la nota menor)



```

La menor nota es : 12.0
El promedio es: : 15.0
BUILD SUCCESSFUL (total time: 40 seconds)

```

4.2 ¿Cuáles serían los atributos de la clase Ventana (de ordenador)? ¿cuáles serían los métodos? Piensa en las propiedades y en el comportamiento de una ventana de cualquier programa.

- De La clase Ventana (se puede identificar los siguientes atributos)
 - Color
 - Ancho
 - Largo
 - Ubicación
 - Visibilidad
- Como los métodos de la clase Ventana se puede mencionar los siguientes
 - Maximizar

- Minimizar
- Cerrar
- Restaurar
- Mover

```
package TA05;

/**
 *
 * @author luudwin
 */
public class Ventana {

    // declarando los atributos de la ventana del ordenador

    String Color;
    String Nombre;
    float ancho;
    float largo;
    // tambien podemos considerar como atributos la "ubicacion" la "visibilidad"

    // en cuanto a sus metodos se puede considerar los siguientes:

    public void maximizarVentana() {
    }
    public void minimizarVentana() {
    }
    public void cerrarVentana() {
    }
    public void restaurarVentana() {
    }
    public void moverVentana() {
    }

}
```

4.3 Implemente una clase de nombre Empleado con los siguientes atributos: código, nombre, área laboral, sueldo, horas extras, afiliación a una AFP. Además, considere atributos de valores comunes para todos los empleados, para los porcentajes de descuento por afiliación a una AFP (11% del sueldo), por afiliación al sistema nacional de pensiones (13% del sueldo) y por salud (3% del sueldo). Considere método de cálculo para el monto de horas extras (sueldo básico/240 * horas extras), monto de los descuentos por AFP, por SNP, por salud, monto total de descuentos, sueldo total, sueldo neto.

Realizando el código del ejercicio

```

package TA05;

/* creamos la clase empleados e implementamos los procedimientos y metodos
   a calcular

*/
public class Empleado {
    // Declarando Variables;
    String codigo;
    String nombre;
    String area;
    String seguro;
    double horasExtras;
    double sueldo;

    // valores comunes para todos los objetos
    private final double snp = 0.13;
    private final double essalud = 0.3;
    private final double afp = 0.11;

    // constructor
    public Empleado (String codigo, String nombre, String area, String seguro,
        double horasExtras, double sueldo){

        this.codigo = codigo;
        this.nombre = nombre;
        this.area = area;
        this.seguro = seguro;
        this.sueldo = sueldo;
        this.horasExtras = horasExtras;
    }

    // asignando metodos para realizar el calculo
    // calculando el monto extra
    public double montoExtra(){
        return sueldo*horasExtras/240;
    }

    // calculando el monto del seguro
    public double montoSeguro(){
        if(seguro.equalsIgnoreCase("AFP"))
            return sueldo*afp;
        if(seguro.equalsIgnoreCase("SNP"))
            return sueldo*snp;
        return 0;
    }

    // calculando el monto de Essalud;

    public double montoEssalud(){
        return sueldo*essalud;
    }

    // calculando el monto de los descuentos
    public double montoDescuento(){
        return montoSeguro()+ montoEssalud();
    }

    // calculando el monto del sueldo bruto
    public double sueldoBruto(){
        return sueldo + montoExtra();
    }

    // calculando el sueldo neto
    public double sueldoNeto(){
        return sueldoBruto()-montoDescuento();
    }
}

```

Creamos una clase PruebaEmpleado y creamos dentro de ella un objeto empleado1

```
package TA05;

public class PruebaEmpleado {
    //utilizamos el metodo Principal main
    public static void main(String[] args) {

        /* y luego creamos un objeto Empleado1 ingresamos los parametros
           de acuerdo al constructor creado
        */
        Empleado empleado1 = new Empleado("1234", "Edwin ", "Sistemas",
            "afp", 10.5, 1500);

        System.out.println(" el empleado " + empleado1.nombre +
            "trabaja en el area de " + empleado1.area + " gana un sueldo"
            + " de: S/." + empleado1.sueldo + " ");

        System.out.println("El monto de la horas extras es: S/. "
            + empleado1.montoExtra());
        System.out.println("El monto del Seguro del empleado es: S/. "
            + empleado1.montoSeguro());
        System.out.println("El monto de Essaud del empleado es: S/. "
            + empleado1.montoEssaud());
        System.out.println("total descuento es: S/. "
            + empleado1.montoDescuento());

        System.out.println(" el sueldo neto es: S/. "
            + empleado1.sueldoNeto());

    }
}
```

El cual nos muestra el siguiente es el resultado

```
Output - TA05_Edwin_Ore (run)

run:
 el empleado Edwin trabaja en el area de Sistemas gana un sueldo de: S/.1500.0
El monto de la horas extras es: S/. 65.625
El monto del Seguro del empleado es: S/. 165.0
El monto de Essaud del empleado es: S/. 450.0
total descuento es: S/. 615.0
 el sueldo neto es: S/. 950.625
BUILD SUCCESSFUL (total time: 0 seconds)
```

5 Conclusiones

En este laboratorio aprendemos que para entender cómo funciona el paradigma de la programación orientada a objetos es necesario ver un programa como una colección de objetos que interactúan entre sí enviándose mensajes y cambiando su estado durante la ejecución.

Resolver un problema bajo el paradigma de la programación orientada a objetos implica determinar y caracterizar los diferentes objetos que intervienen en el problema, definir sus propiedades y métodos y ponerlos a interactuar.