

Segundo entregable POO

Estudiante: Jhonatan David Asprilla Arango

CC: 1018222341

Usuario UNAL: jasprilla

Repositorio Github: <https://github.com/jdaar/POO-UNAL>

Punto 1

Cap 3. Numeral 18

```
package entregable_dos;

import java.math.BigDecimal;
import java.util.Scanner;

class Empleado {
    public int codigo;
    public String[] nombres;
    // Se utiliza el tipo BigDecimal en vez de float o double ya que es peligroso
    // operar cantidades que necesitan una alta precision con estos tipos (como en este
    // caso, el dinero)
    // @see https://docs.oracle.com/cd/E19957-01/806-3568/ncg_goldberg.html
    public BigDecimal horas_trabajadas;
    public BigDecimal precio_hora;
    public BigDecimal retencion_fuente;

    public Empleado(int codigo, String[] nombres, BigDecimal horas_trabajadas,
        BigDecimal precio_hora, BigDecimal retencion_fuente) {
        this.codigo = codigo;
        this.nombres = nombres;
        this.horas_trabajadas = horas_trabajadas;
        this.precio_hora = precio_hora;
        this.retencion_fuente = retencion_fuente;
    }

    public String toString() {
        BigDecimal salario_bruto = horas_trabajadas.multiply(precio_hora);
        return String.format("El empleado %s con el codigo %d tiene un salario
bruto de %s y neto de %s", String.join(" ", nombres), codigo,
salario_bruto.toString(),
```

```

salario_bruto.subtract(salario_bruto.multiply(retencion_fuente)).toString());
    }
}

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

        System.out.print("Introduzca el codigo del empleado: ");
        int codigo = scanner.nextInt();

        // Para consumir el newline (\n) que no fue consumido por scanner.nextInt,
        esto es para evitar que el siguiente scanner.nextLine capture unicamente este
        caracter en vez de esperar por el input del usuario
        scanner.nextLine();

        System.out.print("Introduzca los nombres del empleado separados por
espacio: ");
        String[] nombres = scanner.nextLine().split(" ");

        System.out.print("Introduzca las horas trabajadas por el empleado: ");
        BigDecimal horas_trabajadas = new BigDecimal(scanner.nextLine());

        System.out.print("Introduzca el precio de la hora del empleado: ");
        BigDecimal precio_hora = new BigDecimal(scanner.nextLine());

        System.out.print("Introduzca el porcentaje de retencion en la fuente (el
valor debe estar entre 0 y 1): ");
        BigDecimal retencion_fuente = new BigDecimal(scanner.nextLine());

        scanner.close();

        Empleado empleado = new Empleado(codigo, nombres, horas_trabajadas,
precio_hora, retencion_fuente);
        System.out.println(empleado.toString());
    }
}

```

Cap 3. Numeral 19

```

package entregable_dos;

import java.math.BigDecimal;
import java.util.Scanner;

```

```

class Empleado {
    public int codigo;
    public String[] nombres;
    // Se utiliza el tipo BigDecimal en vez de float o double ya que es peligroso
    // operar cantidades que necesitan una alta precision con estos tipos (como en este
    // caso, el dinero)
    // @see https://docs.oracle.com/cd/E19957-01/806-3568/ncg_goldberg.html
    public BigDecimal horas_trabajadas;
    public BigDecimal precio_hora;
    public BigDecimal retencion_fuente;

    public Empleado(int codigo, String[] nombres, BigDecimal horas_trabajadas,
        BigDecimal precio_hora, BigDecimal retencion_fuente) {
        this.codigo = codigo;
        this.nombres = nombres;
        this.horas_trabajadas = horas_trabajadas;
        this.precio_hora = precio_hora;
        this.retencion_fuente = retencion_fuente;
    }

    public String toString() {
        BigDecimal salario_bruto = horas_trabajadas.multiply(precio_hora);
        return String.format("El empleado %s con el codigo %d tiene un salario
bruto de %s y neto de %s", String.join(" ", nombres), codigo,
salario_bruto.toString(),
salario_bruto.subtract(salario_bruto.multiply(retencion_fuente)).toString());
    }
}

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

        System.out.print("Introduzca el codigo del empleado: ");
        int codigo = scanner.nextInt();

        // Para consumir el newline (\n) que no fue consumido por scanner.nextInt,
        // esto es para evitar que el siguiente scanner.nextLine capture unicamente este
        // caracter en vez de esperar por el input del usuario
        scanner.nextLine();

        System.out.print("Introduzca los nombres del empleado separados por
espacio: ");
        String[] nombres = scanner.nextLine().split(" ");

        System.out.print("Introduzca las horas trabajadas por el empleado: ");

```

```

        BigDecimal horas_trabajadas = new BigDecimal(scanner.nextLine());

        System.out.print("Introduzca el precio de la hora del empleado: ");
        BigDecimal precio_hora = new BigDecimal(scanner.nextLine());

        System.out.print("Introduzca el porcentaje de retencion en la fuente (el
valor debe estar entre 0 y 1): ");
        BigDecimal retencion_fuente = new BigDecimal(scanner.nextLine());

        scanner.close();

        Empleado empleado = new Empleado(codigo, nombres, horas_trabajadas,
precio_hora, retencion_fuente);
        System.out.println(empleado.toString());
    }
}

```

Cap 3. Numeral 20

```

package entregable_dos;

import java.util.Scanner;

// Para evitar problemas por nombramiento ambiguo en el paquete
class TrianguloEjercicioTres {
    public double lado_uno;
    public double lado_dos;
    public double lado_tres;

    public TrianguloEjercicioTres(double lado_uno, double lado_dos, double
lado_tres) {
        this.lado_uno = lado_uno;
        this.lado_dos = lado_dos;
        this.lado_tres = lado_tres;
    }

    public double calcular_perimetro() {
        return lado_uno + lado_dos + lado_tres;
    }

    public double calcular_semiperimetro() {
        return calcular_perimetro() / 2;
    };

    public double calcular_area() {

```

```

// @see https://es.wikipedia.org/wiki/F%C3%B3rmula_de_Her%C3%B3n
double s = this.calcular_semiperimetro();
return Math.sqrt(s * (s - lado_uno) * (s - lado_dos) * (s - lado_tres));
};
}

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

        System.out.print("Introduzca la longitud del lado 1: ");
        double lado_uno = scanner.nextDouble();

        System.out.print("Introduzca la longitud del lado 1: ");
        double lado_dos = scanner.nextDouble();

        System.out.print("Introduzca la longitud del lado 1: ");
        double lado_tres = scanner.nextDouble();

        scanner.close();

        TrianguloEjercicioTres triangulo = new TrianguloEjercicioTres(lado_uno,
lado_dos, lado_tres);

        System.out.printf(
            "El perimetro del triangulo es: %f\nEl area del triangulo es:
%f\nEl semiperimetro del triangulo es: %f",
            triangulo.calcular_perimetro(), triangulo.calcular_area(),
            triangulo.calcular_semiperimetro());
    }
}

```

Cap 4. Numeral 7

```

package entregable_dos;

import java.util.Scanner;

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

        System.out.print("Introduzca el valor A: ");
        double a = scanner.nextDouble();
    }
}

```

```

        System.out.print("Introduzca el valor B: ");
        double b = scanner.nextDouble();

        scanner.close();

        System.out.println(a < b ? "B es mayor que A" : "A es mayor que B");
    }
}

```

Cap 4. Numeral 10

```

package entregable_dos;

import java.math.BigDecimal;
import java.util.Scanner;

class Estudiante {
    public static BigDecimal umbral = new BigDecimal("2000000");

    public int numero_inscripcion;
    public String[] nombres;
    public BigDecimal patrimonio;
    public int estrato;

    public Estudiante(int numero_inscripcion, String[] nombres, BigDecimal
patrimonio, int estrato) {
        this.numero_inscripcion = numero_inscripcion;
        this.nombres = nombres;
        this.patrimonio = patrimonio;
        this.estrato = estrato;
    }

    public BigDecimal calcular_pago_matricula() {
        return patrimonio.compareTo(Estudiante.umbral) > 0 && estrato > 3 ? new
BigDecimal("50000").add(patrimonio.multiply(new BigDecimal("0.03"))) : new
BigDecimal("50000");
    }
}

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

        System.out.print("Introduzca el numero de inscripcion: ");
        int numero_inscripcion = scanner.nextInt();
    }
}

```

```

scanner.nextLine();

System.out.print("Introduzca los nombres separados por espacio: ");
String[] nombres = scanner.nextLine().split(" ");
System.out.print("Introduzca el patrimonio: ");
BigDecimal patrimonio = new BigDecimal(scanner.nextLine());
System.out.print("Introduzca el estrato: ");
int estrato = scanner.nextInt();

scanner.close();

Estudiante estudiante = new Estudiante(numero_inscripcion, nombres,
patrimonio, estrato);

System.out.printf("El estudiante con numero de inscripcion %s y nombre %s
debe pagar $%s en matricula\n", estudiante.numero_inscripcion, String.join(" ",
estudiante.nombres), estudiante.calcular_pago_matricula().toString());
}
}

```

Cap 4. Numeral 11

```

package entregable_dos;

import java.util.Scanner;

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

        System.out.print("Introduzca el primer numero: ");
        int a = scanner.nextInt();
        System.out.print("Introduzca el segundo numero: ");
        int b = scanner.nextInt();
        System.out.print("Introduzca el tercer numero: ");
        int c = scanner.nextInt();

        scanner.close();

        int biggest = a > b ? (a > c ? a : c) : (b > c ? b : c);

        System.out.printf("El mayor numero entre %d, %d y %d es %d\n", a, b, c,
biggest);
    }
}

```

```
}  
}
```

Cap 4. Numeral 12

```
package entregable_dos;  
  
import java.math.BigDecimal;  
import java.util.Scanner;  
  
class Trabajador {  
    String[] nombres;  
    int horas_trabajadas;  
    BigDecimal valor_hora;  
  
    public Trabajador(String[] nombres, int horas_trabajadas, BigDecimal  
valor_hora) {  
        this.nombres = nombres;  
        this.horas_trabajadas = horas_trabajadas;  
        this.valor_hora = valor_hora;  
    }  
  
    public BigDecimal calcular_salario() {  
        return this.valor_hora.multiply(BigDecimal.valueOf(horas_trabajadas));  
    }  
}  
  
public class ejercicio_siete {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
  
        System.out.print("Introduzca los nombres separados por espacio: ");  
        String[] nombres = scanner.nextLine().split(" ");  
        System.out.print("Introduzca las horas trabajadas: ");  
        int horas_trabajadas = scanner.nextInt();  
  
        scanner.nextLine();  
  
        System.out.print("Introduzca el valor de la hora: ");  
        BigDecimal valor_hora = new BigDecimal(scanner.nextLine());  
  
        scanner.close();  
  
        Trabajador trabajador = new Trabajador(nombres, horas_trabajadas,  
valor_hora);
```



```

        System.out.printf("El trabajador con nombre %s devenga $%s\n",
String.join(" ", trabajador.nombres), trabajador.calcular_salario().toString());
    }
}

```

Cap 4. Numeral 13

```

package entregable_dos;

import java.math.BigDecimal;
import java.util.Scanner;

enum Bola {
    BLANCA("blanca", BigDecimal.valueOf(0)),
    VERDE("verde", BigDecimal.valueOf(0.1)),
    AMARILLA("amarilla", BigDecimal.valueOf(0.25)),
    AZUL("azul", BigDecimal.valueOf(0.5)),
    ROJA("roja", BigDecimal.valueOf(1));

    public String nombre;
    public BigDecimal descuento;

    private Bola(String nombre, BigDecimal descuento) {
        this.nombre = nombre;
        this.descuento = descuento;
    }
}

public class ejercicio_ocho {
    public static void main(String[] args) {
        Bola[] bolas = { Bola.BLANCA, Bola.VERDE, Bola.AMARILLA, Bola.AZUL,
Bola.ROJA };

        Scanner scanner = new Scanner(System.in);

        System.out.print("Introduzca el valor de la compra: ");
        BigDecimal valor_compra = new BigDecimal(scanner.nextLine());
        System.out.print("Introduzca el color de la bola: ");
        String color_bola = scanner.nextLine().strip().toLowerCase();

        scanner.close();

        for (Bola bola : bolas) {
            if (bola.nombre.equals(color_bola)) {

```

```

        System.out.printf("El valor a pagar es $%\n",
valor_compra.subtract(valor_compra.multiply(bola.descuento)));
        System.exit(0);
    }
}

System.out.println("No introdujo un valor valido para el color de la
bola");
System.exit(1);
}
}

```

Cap 4. Numeral 14

```

package entregable_dos;

import java.math.BigDecimal;
import java.util.Scanner;

class Departamento {
    public BigDecimal ventas;
    public BigDecimal salario_por_empleado;

    public Departamento(BigDecimal ventas, BigDecimal salario_por_empleado) {
        this.ventas = ventas;
        this.salario_por_empleado = salario_por_empleado;
    }

    // Si despejamos la desigualdad planteada por el problema
    // Que en este caso seria  $0.3(v_1+v_2+v_3) \geq v_n$  (donde  $v_1, v_2, v_3, \dots, v_n$ .  $n =$ 
    //  $\{1, 2, 3\}$ )
    // Veremos que podemos cambiar esta ecuacion para que tome la siguiente forma
    //  $0.33(v_2+v_3) \geq (1-0.33)(v_1)$  para  $v_1$ 
    //  $0.33(v_1+v_3) \geq (1-0.33)(v_2)$  para  $v_2$ 
    //  $0.33(v_1+v_2) \geq (1-0.33)(v_3)$  para  $v_3$ 
    // Lo que nos permite definir la funcion como un metodo de  $v_n$ 
    // Esto es muy util para la mantenibilidad del codigo ya que podemos definir
    // Si se hace el aumento como un atributo de la clase
    public Boolean se_hace_aumento(Departamento[] otros_departamentos) {
        BigDecimal valor_a_comparar = BigDecimal.valueOf(0);
        for (Departamento departamento : otros_departamentos) {
            valor_a_comparar =
valor_a_comparar.add(departamento.ventas.multiply(BigDecimal.valueOf(0.33)));
        }
        return

```

```

this.ventas.multiply(BigDecimal.valueOf(0.77)).compareTo(valor_a_comparar) > 0;
    }
}

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

        Departamento[] departamentos = new Departamento[3];

        BigDecimal ventas;
        BigDecimal salario;
        for (int x = 1; x < 4; x += 1) {
            System.out.printf("Introduce las ventas del departamento %d: ", x);
            ventas = new BigDecimal(scanner.nextLine());
            System.out.printf("Introduce el salario de los empleados del
departamento %d: ", x);
            salario = new BigDecimal(scanner.nextLine());

            departamentos[x - 1] = new Departamento(ventas, salario);
        }

        // 0.33(v2+v3) >= (1-0.33)(v1) para v1
        System.out.printf("El salario para los empleados del departamento 1 es
$%s\n",
            (departamentos[0].se_hace_aumento(new Departamento[] {
departamentos[1], departamentos[2] })))
        ?
        departamentos[0].salario_por_empleado.add(departamentos[0].salario_por_empleado.multiply(BigDecimal.valueOf(0.2)))
            : departamentos[0].salario_por_empleado.toString());
        // 0.33(v1+v3) >= (1-0.33)(v2) para v2
        System.out.printf("El salario para los empleados del departamento 2 es
$%s\n",
            (departamentos[1].se_hace_aumento(new Departamento[] {
departamentos[0], departamentos[2] })))
        ?
        departamentos[1].salario_por_empleado.add(departamentos[1].salario_por_empleado.multiply(BigDecimal.valueOf(0.2)))
            : departamentos[1].salario_por_empleado.toString());
        // 0.33(v1+v2) >= (1-0.33)(v3) para v3
        System.out.printf("El salario para los empleados del departamento 3 es
$%s\n",
            (departamentos[2].se_hace_aumento(new Departamento[] {
departamentos[0], departamentos[1] })))
        ?

```

```

departamentos[2].salario_por_empleado.add(departamentos[2].salario_por_empleado.multiply(BigDecimal.valueOf(0.2)))
                                : departamentos[2].salario_por_empleado.toString());

    scanner.close();
}
}

```

Cap 4. Numeral 15

```

package entregable_dos;

import java.util.Scanner;

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

        System.out.print("Introduzca el primer numero: ");
        float a = scanner.nextInt();
        System.out.print("Introduzca el segundo numero: ");
        float b = scanner.nextInt();
        System.out.print("Introduzca el tercer numero: ");
        float c = scanner.nextInt();
        System.out.print("Introduzca el cuarto numero: ");
        float d = scanner.nextInt();

        scanner.close();

        // Se revisa si el primer valor es el diferente y de no ser el caso se va
        // revisando de uno en uno para identificar
        String indice_diferente = (b == d && b == c && b == d) ? "primer" : a != b
? "segundo" : (b != c ? "tercer": "cuarto");
        float valor_diferente = (b == d && b == c && b == d) ? a : (a != b ? b : (b
!= c ? c : d));

        System.out.printf("Se identifico que el %s numero que se introdujo fue el
diferente con el valor %2f", indice_diferente, valor_diferente);
    }
}

```

Cap 4. Numeral 22

```

package entregable_dos;

```

```

import java.math.BigDecimal;
import java.util.Scanner;

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

        System.out.print("Introduzca el nombre del empleado: ");
        String nombre = scanner.nextLine();

        System.out.print("Introduzca el salario por hora del empleado: ");
        BigDecimal salario_hora = new BigDecimal(scanner.nextLine());

        System.out.printf(salario_hora.compareTo(BigDecimal.valueOf(450000)) > 0 ?
"Nombre del empleado: %s. Salario del empleado: %s\n" : "Nombre del empleado:
%s\n", nombre, salario_hora.toString());

        scanner.close();
    }
}

```

Cap 4. Numeral 23

```

package entregable_dos;

import java.util.Scanner;

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

        System.out.print("Introduzca el valor de A: ");
        double a = scanner.nextInt();
        System.out.print("Introduzca el valor de B: ");
        double b = scanner.nextInt();
        System.out.print("Introduzca el valor de C: ");
        double c = scanner.nextInt();

        scanner.close();

        // nuevamente, si se deseara real precision de la solucion es recomendable
        usar BigDecimal
        double discriminante = Math.pow(b, 2) - 4 * a * c;

        if (discriminante < 0) {

```

```

        System.out.printf("La ecuacion %.2fx^2+%.2fx+%.2f = 0 no tiene
soluciones", a, b, c);
    } else {
        double solucion_1 = (-1 * b + Math.sqrt(discriminante)) / (2 * a);
        double solucion_2 = (-1 * b - Math.sqrt(discriminante)) / (2 * a);
        System.out.printf("Las soluciones de la ecuacion %.2fx^2+%.2fx+%.2f = 0
son x=%.2f, x=%.2f", a, b, c, solucion_1, solucion_2);
    }
}
}
}

```

Cap 4. Numeral 24

```

package entregable_dos;

import java.util.Scanner;

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

        System.out.print("Introduzca el primer valor: ");
        double a = scanner.nextInt();
        System.out.print("Introduzca el segundo valor: ");
        double b = scanner.nextInt();
        System.out.print("Introduzca el tercer valor: ");
        double c = scanner.nextInt();

        scanner.close();

        System.out.printf("El valor mas grande entre [%.2f, %.2f, %.2f] es %.2f",
a, b, c, a > b ? a : (b > c ? b : c));
    }
}

```

Punto 2

Codigo

```

package entregable_dos;

enum TipoTriangulo {
    ISOSCELES("isosceles"),
    ESCALENO("escaleno"),

```

```
EQUILATERO("equilatero");

public String nombre_tipo;

private TipoTriangulo(String nombre_tipo) {
    this.nombre_tipo = nombre_tipo;
}
}

interface Figura {
    public double calcular_area();
    public double calcular_perimetro();
}

class Circulo implements Figura {
    public double radio;

    public Circulo(double radio) {
        this.radio = radio;
    }

    public double calcular_area() {
        return Math.PI * Math.pow(radio, 2);
    }

    public double calcular_perimetro() {
        return 2 * Math.PI * radio;
    }
}

class TrianguloRectangulo implements Figura {
    public double base;
    public double altura;

    public TrianguloRectangulo(double base, double altura) {
        this.base = base;
        this.altura = altura;
    }

    public double calcular_area() {
        return base * altura / 2;
    }

    public double calcular_perimetro() {
        return base + altura + this.calcular_hipotenusa();
    }
}
```

```

    public TipoTriangulo determinar_tipo() {
        return base == altura ? (altura == this.calcular_hipotenusa() ?
TipoTriangulo.EQUILATERO : TipoTriangulo.ISOSCELES) : TipoTriangulo.ESCALENO;
    }

    public double calcular_hipotenusa() {
        return Math.sqrt(Math.pow(base, 2) + Math.pow(altura, 2));
    }
}

class Rectangulo implements Figura {
    protected double base;
    protected double altura;

    public Rectangulo(double base, double altura) {
        this.base = base;
        this.altura = altura;
    }

    public double calcular_area() {
        return this.base * this.altura;
    }

    public double calcular_perimetro() {
        return this.base * 2 + this.altura * 2;
    }

    // Se opto por usar getters y setters para delegar la definicion
    // de los valores de base y altura, de esta manera podemos definir cuadrado
    // a partir de rectangulo
    // Getters y setters
    public double get_base() {
        return base;
    }
    public void set_base(double base) {
        this.base = base;
    }
    public double get_altura() {
        return altura;
    }
    public void set_altura(double altura) {
        this.altura = altura;
    }
}

```



```

class Cuadrado extends Rectangulo {
    private double lado;

    public Cuadrado(double lado) {
        super(lado, lado);
        this.lado = lado;
    }

    // Getters y setters
    public double get_lado() {
        return lado;
    }
    public void set_lado(double lado) {
        this.altura = lado;
        this.base = lado;
        this.lado = lado;
    }
}

public class ejercicio_catorce {
    public static void main(String[] args) {
        Circulo figura1 = new Circulo(2);
        Rectangulo figura2 = new Rectangulo(1,2);
        Cuadrado figura3 = new Cuadrado(3);
        TrianguloRectangulo figura4 = new TrianguloRectangulo(3,5);
        System.out.printf("El área del círculo es = %.2f\n",
figura1.calcular_area());
        System.out.printf("El perímetro del círculo es = %.2f\n\n",
figura1.calcular_perimetro());
        System.out.printf("El área del rectángulo es = %.2f\n",
figura2.calcular_area());
        System.out.printf("El perímetro del rectángulo es = %.2f\n\n",
figura2.calcular_perimetro());
        System.out.printf("El área del cuadrado es = %.2f\n",
figura3.calcular_area());
        System.out.printf("El perímetro del cuadrado es = %.2f\n\n",
figura3.calcular_perimetro());
        System.out.printf("El área del triángulo es = %.2f\n",
figura4.calcular_area());
        System.out.printf("El perímetro del triángulo es = %.2f\n",
figura4.calcular_perimetro());
        System.out.printf("El triangulo es de tipo = %s",
figura4.determinar_tipo().nombre_tipo);
    }
}

```

Diagrama

