

## **Trabajo Práctico N° 9:** **Módulo Objetos (Herencia).**

### **Ejercicio 1.**

*(a) Incluir la clase Triángulo a la jerarquía de figuras vista (carpeta tema4). Triángulo debe heredar de Figura todo lo que es común y definir su constructor y sus atributos y métodos propios. Además, se debe redefinir el método toString.*

*(b) De igual manera, incluir la clase Círculo a la jerarquía de figuras.*

*(c) Añadir, a la representación String, el valor del perímetro. Pensar: ¿qué método toString se debe modificar: el de cada subclase o el de Figura?*

*(d) Añadir el método despintar que establece los colores de la figura a línea “negra” y relleno “blanco”. Pensar: ¿dónde se debe definir el método: en cada subclase o en Figura?*

*(e) Realizar un programa que instancie un triángulo y un círculo. Mostrar en consola la representación String de cada uno. Probar el funcionamiento del método despintar.*

Ver proyecto “TP9” de Java.

**Ejercicio 2.**

*Se quiere representar a los empleados de un club: jugadores y entrenadores.*

- *Cualquier empleado se caracteriza por su nombre, sueldo básico y antigüedad.*
- *Los jugadores son empleados que se caracterizan por el número de partidos jugados y el número de goles anotados.*
- *Los entrenadores son empleados que se caracterizan por la cantidad de campeonatos ganados.*

*(a) Implementar la jerarquía de clases declarando atributos, métodos para obtener/modificar su valor y constructores que reciban los datos necesarios.*

*(b) Cualquier empleado debe responder al mensaje calcularEfectividad. La efectividad del entrenador es el promedio de campeonatos ganados por año de antigüedad, mientras que la del jugador es el promedio de goles por partido.*

*(c) Cualquier empleado debe responder al mensaje calcularSueldoACobrar. El sueldo a cobrar es el sueldo básico más un 10% del básico por cada año de antigüedad y, además:*

- *Para los jugadores: si el promedio de goles por partido es superior a 0,5, se adiciona un plus de otro sueldo básico.*
- *Para los entrenadores: se adiciona un plus por campeonatos ganados (\$5.000 si ha ganado entre 1 y 4 campeonatos; \$30.000 si ha ganado entre 5 y 10 campeonatos; \$50.000 si ha ganado más de 10 campeonatos).*

*(d) Cualquier empleado debe responder al mensaje toString, que devuelve un String que lo representa, compuesto por nombre, sueldo a cobrar y efectividad.*

*(f) Realizar un programa que instancie un jugador y un entrenador. Informar la representación String de cada uno.*

*NOTA: Para cada método a implementar, pensar en qué clase/s se debe definir el método.*

**Ver proyecto “TP9” de Java.**

**Ejercicio 3.**

**(a)** Implementar las clases para el siguiente problema. Una garita de seguridad quiere identificar los distintos tipos de personas que entran a un barrio cerrado. Al barrio, pueden entrar: personas, que se caracterizan por nombre, DNI y edad; y trabajadores, estos son personas que se caracterizan, además, por la tarea realizada en el predio. Implementar constructores, getters y setters para las clases. Además, tanto las personas como los trabajadores, deben responder al mensaje `toString` siguiendo el formato:

- *Personas:* “Mi nombre es Mauro, mi DNI es 11.203.737 y tengo 70 años.”
- *Trabajadores:* “Mi nombre es Mauro, mi DNI es 11.203.737 y tengo 70 años. Soy jardinero.”

**(b)** Realizar un programa que instancie una persona y un trabajador y mostrar la representación de cada uno en consola.

NOTA: Reutilizar la clase *Persona* (carpeta tema2).

Ver proyecto “TP9” de Java.

**Ejercicio 4.**

Un objeto visor de figuras se encarga de mostrar en consola cualquier figura que reciba y también mantiene cuántas figuras mostró. Analizar y ejecutar el siguiente programa y responder: ¿Qué imprime? ¿Por qué?

<pre>public class VisorFiguras {     private int mostradas;      public VisorFiguras(){         mostradas=0;     }      public void mostrar(Figura f){         System.out.println(f.toString());         mostradas++;     }      public int getMostradas() {         return mostradas;     } }</pre>	<pre>public class MainVisorFiguras {     public static void main(String[] args) {         VisorFiguras visor = new VisorFiguras();          Cuadrado c1 = new Cuadrado(10,"Violeta","Rosa");         Rectangulo r= new Rectangulo(20,10,"Azul","Celeste");         Cuadrado c2= new Cuadrado(30,"Rojo","Naranja");          visor.mostrar(c1);         visor.mostrar(r);         visor.mostrar(c2);          System.out.println(visor.getMostradas());     } }</pre>
--	--

Ver proyecto "TP9" de Java.

**Ejercicio 5.**

(a) *Modificar la clase VisorFiguras: ahora, debe permitir guardar las figuras a mostrar (a lo sumo, 5) y también mostrar todas las figuras guardadas. Usar la siguiente estructura:*

<pre>public class VisorFigurasModificado {     private int guardadas;     private int capacidadMaxima=5;     private Figura [] vector;      public VisorFigurasModificado(){         //completar     }      public void guardar(Figura f){         //completar     }      //sigue a la derecha -&gt;</pre>	<pre>    public boolean quedaEspacio(){         //completar     }      public void mostrar(){         //completar     }      public int getGuardadas() {         return guardadas;     } }</pre>
--	--

(b) *Realizar un programa que instancie el visor, guarde dos cuadrados y un rectángulo en el visor y, por último, haga que el visor muestre sus figuras almacenadas.*

Ver proyecto “TP9” de Java.

## **Ejercicio 6.**

*El Servicio Meteorológico Nacional necesita un sistema que permita registrar, para una determinada estación meteorológica, la temperatura promedio mensual de  $N$  años consecutivos a partir de un año  $A$  dado. Además, necesita dos versiones del sistema: una que permita reportar el promedio histórico por años y otra que permita reportar el promedio histórico por meses. Esto se detalla más adelante. De la estación, interesa conocer: nombre y latitud y longitud donde se encuentra. Implementar las clases, constructores y métodos que se considere necesarios para:*

**(a)** *Crear el sistema de registro/reporte, que funcionará en una determinada estación, para  $N$  años consecutivos a partir de un año  $A$ . Iniciar cada temperatura en un valor muy alto.*

**(b)** *Registrar la temperatura de un mes y año recibidos por parámetro. Nota: El mes está en rango 1..12 y el año está en rango  $A..A+N-1$ .*

**(c)** *Obtener la temperatura de un mes y año recibidos por parámetro. Nota: El mes está en rango 1..12 y el año está en rango  $A..A+N-1$ . En caso de no haberse registrado temperatura para ese mes/año, se retorna el valor muy alto.*

**(d)** *Devolver un String que concatena el mes y año en que se registró la mayor temperatura. Nota: Suponer que ya están registradas las temperaturas de todos los meses y años.*

**(e)** *Devolver un String con el nombre de la estación, su latitud y longitud y los promedios mensuales o anuales según corresponda:*

- *La versión del sistema que reporta por años deberá calcular el promedio para cada año (el promedio del año  $X$  se calcula con los datos mensuales de ese año).*
- *La versión del sistema que reporta por meses deberá calcular el promedio para cada mes (el promedio del mes  $M$  se calcula con los datos de todos los años en ese mes).*

*NOTA: Suponer que ya están registradas las temperaturas de todos los meses y años. Utilizar el carácter `\n` para concatenar un salto de línea.*

**(f)** *Realizar un programa principal que cree un Sistema con reporte anual para 3 años consecutivos a partir del 2021, para la estación La Plata (latitud -34,921 y longitud -57,955). Cargar todas las temperaturas (para todos los meses y años). Informar los promedios anuales y el mes y año en que se registró la mayor temperatura. Luego, crear un Sistema con informe mensual para 4 años a partir de 2020, para la estación Mar del Plata (latitud -38,002 y longitud -57,556). Cargar todas las temperaturas (para todos los meses y años). Informar los promedios mensuales y el mes y año en que se registró la mayor temperatura.*

*NOTA: Prestar atención de no violar el encapsulamiento al resolver el ejercicio.*

**Ver proyecto “TP9” de Java.**