

Práctica 4

Herencia

Objetivo. Trabajar con el concepto de herencia y polimorfismo.

Nota: Trabajar sobre la carpeta “tema4” del proyecto

1- Nos piden una aplicación estilo Paint, para ello necesitamos representar figuras geométricas (cuadrados, rectángulos, círculos, triángulos). Todas las figuras tienen color de relleno y color de línea. Además, los triángulos guardan el valor de sus tres lados, los cuadrados el valor de su lado, los círculos el valor del radio, y los rectángulos el valor de la base y de la altura.

Las figuras deben incluir funcionalidad para: crearla a partir de los datos necesarios (constructor), modificar/obtener el valor de los atributos (métodos `get` y `set`), calcular el área y devolverla (método `calcularArea`), calcular el perímetro y devolverlo (método `calcularPerimetro`), y mostrar la representación String de la figura (método `toString`) concatenando toda la información.

A- Analice la jerarquía de figuras (carpeta `tema4`).

B- Incluya la clase `Triángulo` a la jerarquía de figuras. `Triángulo` debe *heredar* de `Figura` todo lo que es común y *definir* su constructor y sus atributos y métodos propios. Además debe *redefinir* el método `toString`.

C- De igual manera, incluya la clase `Círculo` a la jerarquía de figuras.

D- Añada a la representación String el valor del perímetro. Piense ¿qué método `toString` debe modificar: el de cada subclase o el de `Figura`?

E- Añada el método `despintar` que establece los colores de la figura a línea “negra” y relleno “blanco”. Piense ¿dónde debe definir el método: en cada subclase o en `Figura`?

F- Realizar un programa que instancie un triángulo y un círculo. Muestre en consola la representación String de cada uno. Pruebe el funcionamiento del método `despintar`.

2- Queremos 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- Implemente 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.

Taller de Programación 2024 – Módulo POO

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 (5000\$ 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. Informe la representación String de cada uno.

NOTA: para cada método a implementar piense en que clase/s debe definir el método.

3-A- Implemente 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.

Implemente 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 **11203737** y tengo **70** años”
- Trabajadores “Mi nombre es **Mauro**, mi DNI es **11203737** y tengo **70** años. Soy **jardinero**.”

B- Realice un programa que instancie una persona y un trabajador y muestre la representación de cada uno en consola.

NOTA: Reutilice la clase Persona (carpeta tema2).

4- 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 tenga funcionalidad para reportar el promedio histórico por años y otra que tenga funcionalidad para reportar el promedio histórico por meses. *Esto se detalla más adelante.*

De la estación, interesa conocer: nombre, y latitud y longitud donde se encuentra.

Implemente las clases, constructores y métodos que 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**. Inicie 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.**

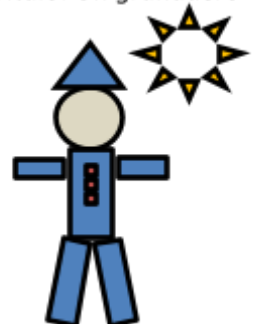
- c) Obtener la temperatura de un mes y año recibidos por parámetro. **Nota: El mes está en rango 1..12 y el año está en rango A..A+N-1. En caso de no haberse registrado temperatura para ese mes/año se retorna el valor muy alto.**
- d) Devolver un String que concatena el mes y año en que se registró la mayor temperatura. **Nota: Suponga 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).
Ej: "La Plata (34,921 S - 57,955 O):
- Año 2020: 23,8 °C;
- Año 2021: 26,1 °C;
- Año 2022: 25,3 °C. "
 - 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).
Ej: "La Plata (34,921 S - 57,955 O):
- Enero: 28,2 °C;
- Febrero: 26,8 °C;
- Marzo: 24.3 °C
- "
- Nota: Suponga que ya están registradas las temperaturas de todos los meses y años. Utilice el carácter \n para concatenar un salto de línea.**
- f) Realice 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). Cargue todas las temperaturas (para todos los meses y años). Informe los promedios anuales, y el mes y año en que se registró la mayor temperatura.
- Luego cree 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). Cargue todas las temperaturas (para todos los meses y años). Informe los promedios mensuales, y el mes y año en que se registró la mayor temperatura.

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

5 A- Queremos representar dibujos. Un dibujo guarda su título y las figuras que lo componen (círculos, triángulos, cuadrados, rectángulos, etc). Piense, con lo visto hasta ahora (no es necesario implementar):

- ¿Dónde almacenará las figuras que componen el dibujo?. ¿Cuántas estructuras se necesitarán?
- ¿Cómo agregará las distintas figuras al dibujo?. ¿Cuántos métodos *agregar* necesita implementar?
- ¿Qué problema surge a medida que aumentan las posibles figuras en la jerarquía?

Título: Un granadero



B- Implemente la clase Dibujo usando un *array genérico de Figuras*. Dicho array puede guardar objetos creados a partir de cualquier subclase de Figura. Siga el molde mostrado.

```
public class Dibujo {
    private String titulo;
    private Figura [] vector;
    private int guardadas;
    private int capacidadMaxima=10;

    //inicia el dibujo, sin figuras
    public Dibujo (String titulo){
        //completar
    }

    //agrega la figura al dibujo
    public void agregar(Figura f){
        //completar
        System.out.println("la figura "+
                           f.toString() +
                           " se ha guardado");
    }

    //calcula el área del dibujo:
    //suma de las áreas de sus figuras
    public double calcularArea(){
        //completar
    }

    //sigue a la derecha ->
}

//imprime el título, representación
//de cada figura, y área del dibujo
public void mostrar(){
    //completar
}

//retorna está lleno el dibujo
public boolean estaLleno() {
    return (guardadas == capacidadMaxima);
}
}

public class MainDibujos {
    public static void main(String[] args) {
        Dibujo d = new Dibujo("un dibujo");

        Cuadrado c1 = new Cuadrado(10,"Violeta","Rosa");
        Rectangulo r= new Rectangulo(20,10,"Azul","Celeste");
        Cuadrado c2= new Cuadrado(30,"Rojo","Naranja");

        d.agregar (c1);
        d.agregar (r);
        d.agregar (c2);

        d.mostrar();
    }
}
```

B- Analice y ejecute programa MainDibujos. Responda: ¿Qué imprime? ¿Por qué?

Piense: ¿qué ventaja tiene esta implementación a medida que aumentan las posibles figuras en la jerarquía? ¿cuál es la ventaja del polimorfismo? ¿dónde se observa en este ejercicio?