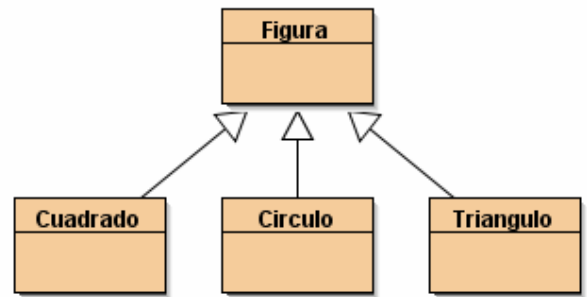


## UD8 POO (II) – Ejercicios B

### Ejercicio 1. Figuras

Escribe un programa con la jerarquía de clases indicada e implementa los atributos y métodos necesarios para que se pueda ejecutar el programa de abajo.



```

public class ProgramaFiguras {
    public static void main(String[] args) {
        ArrayList<Figura> figuras = new ArrayList<Figura>();
        figuras.add(new Circulo(10)); // Radio=10
        figuras.add(new Cuadrado(10)); // Lado=10
        figuras.add(new Triangulo(10,5)); // Base=10, Altura=5;
        for (Figura f: figuras)
            System.out.println("Área: " + f.area());
    }
}
  
```

### Ejercicio 2. Animales

Todos los animales de mi mundo imaginario **cantan**, pero cada animal canta de forma distinta dependiendo de qué tipo sean.

- Todos los **animales** empiezan su canto diciendo su clase de animal, y pueden añadir más cosas al canto según el caso.
- Los **pájaros** añaden “pio pio” al canto. Además, los pájaros **carpinteros** añaden “picomadera” a su canto, y los **gallos** “quiquiriqui”.
- Los **peces** añaden “glu glu” a su canto. Además, los peces **payaso** añaden “chorprecha” a su canto, y los peces **espada** “ríndete”.

Así, por ejemplo, algunos de los cantos de los animales serían:

- Animal => canta “Animal”.
- Pájaro => canta “Pajaro pio pio”.
- Pájaro carpintero => canta “PajaroCarpintero pio pio picomadera”.

1. Escribe la lista de los 7 animales y qué canta cada uno.
2. Haz el diagrama de clases simplificado.
3. Programa todas las clases aplicando herencia para reutilizar código (tienes que utilizar ‘super’ para invocar métodos de superclase siempre que sea posible).

### Ejercicio 3. Cuentas Bancarias

Un banco tiene **cuentas bancarias**. Cada cuenta bancaria tiene un **titular** (nombre completo) y un **saldo** en euros. En todas las cuentas es posible **ingresar** y **retirar** dinero.

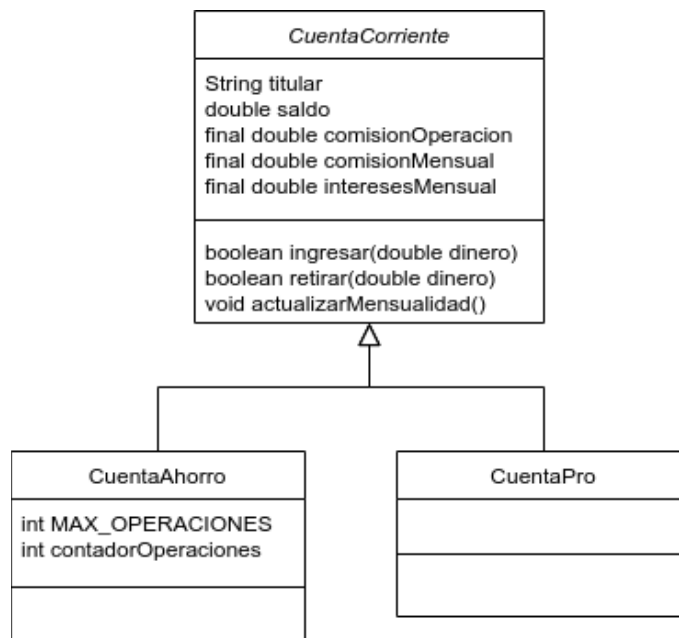
También es posible **actualizar mensualidad** (esto se hace una vez al mes), momento en el que el banco premia las cuentas con una pequeña bonificación por intereses y también cobra unos pocos gastos de mantenimiento.

Existen tres tipos de cuentas bancarias:

- **Cuenta Corriente:** Es la más común para utilizar en el día a día. Se caracteriza por:
  - Permite tener cualquier saldo positivo (nunca negativo).
  - Cada vez que se realiza una operación (ingreso o retirada), cobra 1€ de la cuenta.
  - Al actualizar mensualidad da un 1% de intereses y cobra 5€ de mantenimiento.
- **Cuenta de Ahorro:** Está pensada para ahorrar a largo plazo, no para ser usada a diario. Es igual que la cuenta corriente salvo por un par de diferencias:
  - Solo permite hacer 3 operaciones (ingresar o retirar) al mes.
  - Al actualizar mensualidad da el 10% de intereses y no cobra mantenimiento.
- **Cuenta Pro:** Está pensada para quienes hacen muchas operaciones y no les importa la rentabilidad. Es igual que la cuenta corriente salvo por un par de diferencias:
  - No se cobra por hacer una operación.
  - Al actualizar mensualidad da el 0% de intereses y cobra 100€ de mantenimiento.

Implementa la superclase CuentaCorriente y luego las subclases CuentaAhorro y CuentaPro con herencia. Luego, en una clase Banco con función 'main', haz las siguientes pruebas:

1. Instancia al menos una cuenta de cada tipo y prueba a hacer varios ingresos, retiradas, actualizar saldo, etc. para comprobar que todo funciona. Te será de ayuda añadir el método toString para imprimir la información de las cuentas.
2. Crea un ArrayList<CuentaCorriente> y añade varias cuentas de los tres tipos. Prueba a recorrer el ArrayList llamando a distintos métodos para comprobar que se ejecuta el método correcto de cada cuenta.



## Ejercicio 4. Cooperativa

Las trabajadoras de una cooperativa han decidido organizarse de una forma un poco diferente y más flexible en cuanto a las horas que trabajan y lo que cobran. Por lo general, cada semana pueden decidir cuántas horas trabajan, y cobran a final de semana según las horas trabajadas. Como ellas son las dueñas de la empresa, hacen lo que quieren.

Todas las **trabajadoras** tienen una serie de características comunes:

- Tiene un **nombre** y un **contador de horas trabajadas** esta semana.
- Todas las trabajadoras tienen el mismo **sueldo base por hora trabajada**, que está fijado en 10€ y no puede variar.
- Pueden decidir **trabajar** una cantidad de horas (suma al contador de horas).
- Pueden **cobrar su nómina** a fin de mes (horas trabajadas x sueldo por hora), lo cual devuelve el contador de horas trabajadas a cero.

Algunas de las trabajadoras son especialistas y tienen características especiales:

- **Operarias de fábrica:** Es un trabajo peligroso así que obtienen un plus del 20% en la nómina pero no pueden trabajar más de 40 horas a la semana.
- **Oficinistas:** Es el trabajo más cómodo de todos, pero a cambio pierden flexibilidad. Tienen que trabajar entre 30 y 40 horas a la semana o reciben una penalización del 50% en la nómina.
- **Transportistas:** Se pasan la semana en la carretera. Por eso, pueden trabajar tantas horas como quieran y encima poner la música a tope todo el camino.



Implementa las clases del diagrama de clases de arriba y crea también la clase **Cooperativa** que contenga como único atributo un `ArrayList<Trabajadora>` y los métodos:

- **Cooperativa():** constructor que instancia el `ArrayList<Trabajadora>` trabajadoras.
- **añadirTrabajadora(Trabajadora t):** añade una trabajadora a la lista.
- **void mostrar():** muestra por pantalla la información de todas las trabajadoras.
- **void trabajar(int i, int horas):** hace que la trabajadora en la posición `i` de la lista trabaje las horas indicadas.
- **void trabajar(int horas):** hace que todas las trabajadoras de la lista trabajen las horas indicadas.
- **double cobrarNomina(int i):** hace que la trabajadora en la posición `i` de la lista cobre su nómina. Devuelve la cantidad que ha cobrado.
- **void cobrarNominas():** hace que todas las trabajadoras cobren su nómina. Muestra por pantalla lo que ha cobrado cada una.

Implementa una clase **Programa** con una función 'main' y programa lo siguiente:

1. Instancia una cooperativa.
2. Añade varias trabajadoras (al menos una de cada tipo).
3. Muestra su información.
4. Simula que pasan 4 semanas (con un bucle for). En cada semana:
  1. Haz que todas trabajen un número aleatorio de horas entre 10 y 50.
  2. Haz que todas cobren sus nóminas.
  3. Muestra su información.

Revisa que la información mostrada tiene sentido.