

Actividad 1

Clase Televisión

Piensa que podría tener la clase Televisión, aquello que debería de saber y debería de hacer:



¡Afila tu lápiz!

Rellena en el objeto Televisión lo que debería saber y debería hacer:

Televisión



Variables de instancia

Métodos

Después, por medio de la aplicación <https://www.draw.io/>, representa la clase con sus elementos. Puedes inventar aquello que creas conveniente. No existe una única solución.

Actividad 2

Define una clase que represente un coche. En la definición se debe de incluir:

- el modelo
- el color
- si la pintura es metalizada o no
- la matrícula
- el tipo de coche, que puede ser mini, utilitario, familiar o deportivo.
- el año de fabricación.

- la modalidad del seguro, que puede ser a terceros o a todo riesgo.

Haz un constructor con valores iniciales (los que se deseen), otro constructor con parámetros y un método que se llame *mostrarInfo()* que muestre el modelo y el color del coche.

Haz un diagrama UML y a continuación la implementación.

Actividad 3

Escribe un programa que disponga de una clase para representar las asignaturas de una carrera:

- Una asignatura tiene un nombre, un código numérico y el curso en el que se imparte..
- Los valores iniciales deben de proporcionarse en el constructor.
- La clase debe tener métodos para obtener los valores de los atributos.
- El programa debe de construir un objeto con los siguientes valores: nombre "Matemáticas", código 1017, curso 1.
- A continuación, el programa debe de imprimir los valores del objeto por pantalla.

Haz el diagrama de clases UML y a continuación, implementa la solución en Java.

Actividad 4

En esta actividad crearemos una clase NIF con las siguientes características:

Los atributos serán el número de DNI y la letra.

- La clase contendrá un método privado que calcule la letra del DNI a partir del número.
- El constructor de la clase tendrá un solo parámetro que será el DNI sin letra, en el momento de la inicialización se llamará a la función de cálculo para iniciar la clase.
- Crea un método que devuelva el NIF (DNI+letra).

Para comprobar que la clase funciona de forma correcta, crea una instancia de la clase NIF número de DNI 21679882 y muestra por pantalla el nif. Pruébala también con tu dni

Actividad 5

Queremos declarar una clase que represente un punto en un espacio bidimensional. La clase debe de cumplir como mínimo con los siguientes requerimientos:

- Cada punto se define por sus coordenadas.
- Implementa un constructor con parámetros para asignar valores a las variables de instancia.
- Implementa los *getters* y *setters*.
- Implementa un método que calcule la distancia a otro punto. Para calcular la distancia entre dos puntos.

$$A(x_1, y_1) \quad B(x_2, y_2)$$
$$d(A, B) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

- Implementa un método que visualice las coordenadas de un punto en formato (coordX, coordY).

Una vez implementada la clase Punto, escribe un programa que instancia cuatro puntos: el primero estará situado en el origen (0,0), el segundo situado en (5,3), el tercero en (2,-1) y el cuarto estará situado en medio del segundo y el tercero (consideramos que en medio quiere decir en la mitad de los valores de la coordenada X y en la mitad de los valores de la coordenada Y). Crea un punto en (4,2) y visualiza la distancia del punto al origen de coordenadas.

Actividad 6

Escribe una clase para representar relojes, de forma que se guarde la hora, el minuto y el segundo. La hora podrá almacenar números entre 0 y 23 (se guardaran los datos en formato 24 horas), los minutos, los minutos y segundos 0 y 59. Es decir, la hora será guardada en formato 24 horas.

la clase debe de disponer de los siguientes métodos:

- a) Dos constructores, uno sin parámetros que pondrá la hora del reloj a 0:0:0 y otro al cual se le pasará la hora, minutos y segundos.

b) Un método que muestre la hora, los minutos y los segundos, separados por ":", en formato 24 horas. Es decir, si el reloj marca las 7:30:00, significa que son las 7 de la mañana, y si marca las 19:30:00, significa que son las 7 de la tarde.

c) Un método que muestre la hora, los minutos y los segundos, separados por ":", en formato 12 horas.

Como en el caso del apartado b), si son las 7 de la mañana, el reloj deberá de mostrar las 7:30:00 AM. Si son las 7 de la tarde, el reloj mostrará las 7:30:00 PM.

d) Un método para poner el reloj en hora. Recibirá como parámetros la hora, los minutos y los segundos los pondrá a 0.

e) Un método como el anterior, pero pasándole también los segundos.

g) *Setters* y *getters* para las variables de instancia.

Actividad 7.

Crea una clase Fracción amb métodos para sumar, restar, multiplicar y dividir fracciones. La clase tendrá dos atributos numerador y denominador.

```
public static void main(String[] args) {  
  
    // Se crean 4 fracciones  
  
    Fraccion f1 = new Fraccion(1, 4); // Fracción 1/4  
  
    Fraccion f2 = new Fraccion(1, 2); // Fracción 1/2  
  
    Fraccion f3 = new Fraccion(); // Fracción 0/1  
  
    Fraccion f4 = new Fraccion(4); // Fracción 4/1  
  
    // operaciones aritméticas con esas fracciones  
  
    Fraccion suma = f1.sumar(f2);  
  
    Fraccion resta = f1.restar(f3);  
  
    Fraccion producto = f1.multiplicar(f4);  
  
    Fraccion cociente = f1.dividir(f2);  
  
    //mostrar resultados  
  
    System.out.println(f1 + " + " + f2 + " = " + suma);  
  
    System.out.println(f1 + " - " + f3 + " = " + resta);  
}
```

```
System.out.println(f1 + " * " + f4 + " = " + producto);
```

```
System.out.println(f1 + " / " + f2 + " = " + cociente);
```

Actividad 8

Crea una clase Fecha. La clase contendrá además de constructores, el método toString, deberá tener método para comprobar si la fecha es correcta y otro para modificar la fecha actual por la del día siguiente.

Actividad 9

Crea una clase Persona. La clase contendrá además de un atributo fechaNacimiento que será un objeto de la clase Fecha creada en el ejercicio anterior, el nombre, dni, dirección, apellidos y email. Los apellidos y la dirección pueden no ser facilitados en el momento de la creación, por lo que deberás de crear constructores para ello.

Actividad 10

Crea una clase Cuenta con los métodos ingreso, reintegro y transferencia. Los datos de la cuenta son: número de cuenta, tipo de interés, saldo y cliente al que pertenece que será representado por la clase Persona.

Actividad 11

Diseña e implementa una clase linterna atendiendo a las siguientes características:

- Se deben implementar 3 modos de funcionamiento; *parpadeo rápido*, *luz fija* y *parpadeo lento*.
- Dispondrá de una batería interna no reemplazable.
- Los modos de funcionamiento de la linterna vendrán dados por un objeto enum.

Actividad 12

Escribe un programa para representar el consumo de energía de una instalación eléctrica. Para hacerlo, tendremos una clase que represente los instrumentos conectados en la instalación. Cada instrumento o aparato tiene un consumo eléctrico determinado. Al encenderlo, el consumo de energía total se incrementará en la potencia de ese aparato. Al apagarlo, se decrementará el consumo total. Inicialmente, los aparatos están todos apagados. Además, se desea consultar el consumo total de la instalación.

Hacer un programa que declare tres aparatos eléctricos: una bombilla de 100 watios, un radiador de 2000 watios y una plancha de 1200 watios. El programa imprimirá el

consumo cuando se creen los objetos. Posteriormente, se conectará la bombilla y la plancha, y el programa mostrará el consumo. Después, se apagará la plancha y se encenderá el radiador y se volverá a mostrar el consumo.

Actividad 13

Escribe una clase llamada Elemento, que disponga de un atributo con el número. La clase debe de contener un método llamado contarElementos() que devuelva el número de elementos que se han instanciado.

Actividad 14

Escribe una clase llamada Item, que acepte un nombre en su constructor. Cada item debe de tener un número que servirá como identificador único de cada objeto. La clase Item dispondrá de un método para obtener el identificador y otro para obtener el nombre. Haz un programa de prueba que genere tres Items, "uno", "dos" y "tres" y después escriba los nombres e identificados de cada Item. Piensa en una solución para que el identificador se genere automáticamente, incrementándose en uno cada vez que se cree un Item.