

Boletín Clases

Ejercicio n1

Defina una clase que sirva para representar el estado de una bombilla (encendido o apagado). Defina, asimismo, dos métodos que permitan encender (on) y apagar (off) la luz de la bombilla.

Para probarlo, cree un pequeño método main que cree un objeto de la clase definida, que haga uso de los métodos previamente definidos y vaya mostrando el estado de la bombilla.

Ejercicio n2

Defina una clase que permita almacenar datos sobre los planetas que forman parte del sistema solar. Algunos atributos que puede ser interesante recoger son: la masa del cuerpo, su diámetro, el período de rotación sobre el propio eje, período de traslación alrededor del sol, distancia media recorrida en su órbita, ...

Defina un método, que de un determinado planeta del sistema solar, imprima toda la información de sobre el mismo.

Ejercicio 3

Defina una clase 'Monedero' que permita gestionar la cantidad de dinero de que una persona dispone en un momento dado. La clase deberá tener un constructor que permitirá crear un monedero con una cantidad de dinero inicial y deberá definir un método para meter dinero en el monedero, otro para sacarlo y finalmente, otro para consultar el disponible; solo podrá conocerse la cantidad de dinero del monedero a través de este último método. Por supuesto, no se podrá sacar más dinero del que haya en un momento dado en el monedero.

Para probar el funcionamiento de la clase, escriba un método 'main' con una serie de instrucciones que hagan uso de los métodos definidos.

Ejercicio n 4

Se quiere definir una clase que permita controlar un sintonizador digital de emisoras FM; concretamente, lo que se desea es dotar al controlador de una interfaz que permita subir (up) o bajar (down) la frecuencia (en saltos de 0.5 MHz) y mostrar la frecuencia sintonizada en un momento dado (display). Supondremos que el rango de frecuencias a manejar oscila entre los 80 Mhz y los 108 MHz y que al inicio, el controlador sintoniza a 80 MHz. Si durante una operación de subida o bajada se sobrepasa uno de los dos límites, la frecuencia sintonizada debe pasar a ser la del extremo contrario.

Ejercicio n 5 (Relación entre objetos)

Se desea llevar un control del estado de una cuenta corriente; la cuenta corriente está caracterizada por su saldo y sobre ella se pueden realizar tres tipos de operaciones:

- saldo: devuelve el saldo de la cuenta (puede ser negativo).
- imposición (cantidad): ingresa en la cuenta una cantidad de dinero.
- reintegro (cantidad): saca de la cuenta una determinada cantidad de dinero.
- Traspaso (cantidad, CuentaCorriente): traspasa una cantidad de dinero a otra cuenta recibida por parámetro.

Suponga que la cuenta inicialmente tiene un saldo de cero. Escriba una clase CuentaCorriente que implemente la funcionalidad descrita; escriba un pequeño programa principal para probar su funcionamiento.

Ejercicio 6 (Relación entre objetos)

Defina una clase punto que tendrá dos atributos, de tipo real, x e y, que representarán las coordenadas del punto dentro del plano. Defina un método que tenga como argumento otro objeto de la clase punto y que calcule la distancia entre los dos puntos.

Para probar esta funcionalidad, escriba un método main en el que se creen dos puntos, el primero de los cuales deberá tener las coordenadas 100.0, 200.0 y el segundo deberá tener las coordenadas 400.0, 800.0 y calcule e imprima la distancia entre ambos. $\text{distancia}(p1,p2) = \text{raizcuadrada}(\text{cuadrado}(x2-x1)+\text{cuadrado}(y2-y1))$

Ejercicio 7 (sobrecarga de metodos)

Escriba una clase en la que se defina un método que escriba por pantalla el tipo de argumento que se le pasa como parámetro. Ejemplo: si se le pasa un carácter (char), que imprima algo del tipo "Soy un carácter". Esta clase deberá ser capaz de detectar todos los tipos primitivos de Java.

Ejercicio 8 (Relación entre distintos objetos)

Cree una clase 'AlarmaDeTemperatura' cuyos objetos activen un objeto 'Timbre' cuando la temperatura llegue a un cierto valor. Para probar la alarma simule en el Main un 'Sensor' que utilice la clase 'AlarmaDeTemperatura' y simule variaciones de temperatura y cambios en la temperatura de alarma. La clase 'Timbre' mostrará un mensaje por pantalla cuando se active y otro cuando se desactive.

Ejercicio n.9 (Relación entre distintos objetos)

Crear una clase cubo que contenga información sobre la capacidad y su contenido actual en litros. Se podrá consultar tanto la capacidad como el contenido en cualquier momento. Dotar a la clase de la capacidad de verter el contenido de un cubo en otro (hay que tener en cuenta si el contenido del cubo origen cabe en el cubo destino, si no cabe, se verterá solo el contenido que quepa). Hacer una clase principal para probar el funcionamiento con un par de cubos.

Ejercicio n 10 (Propiedad estática)

Queremos modelar una casa con muchas bombillas, de forma que cada bombilla se puede encender o apagar individualmente. Para ello haremos una clase Bombilla con una variable privada que almacene si está encendida o apagada, así como un método que nos diga si una bombilla concreta está encendida.

Además queremos poner un interruptor general de la luz, tal que si saltan los fusibles, todas las bombillas quedan apagadas. Cuando el fusible se repara, las bombillas vuelven a estar encendidas o apagadas, según estuvieran antes del percance

Ejercicio n 11 (Propiedad estática)

Se desea representar el conjunto de bombillas que hay en una casa, así como la potencia consumida en un momento dado por todas ellas. Para ello, defina una clase 'Bombilla' que tenga al menos dos atributos: uno para identificarla y otro que indique la potencia que consume; también, deberá crear dos métodos que permitan encender y apagar la bombilla. La clase deberá disponer de un mecanismo que permita obtener la potencia consumida por el conjunto de bombillas que estén encendidas en un determinado instante.

Crear otra clase que cree varias bombillas para probar la funcionalidad.

(Podemos ampliar la clase bombilla del ejercicio anterior en vez de crear una nueva)

Ejercicio n.12 (Propiedad estática)

Crear una clase Votacion que permita llevar el control de unas elecciones entre distintos candidatos. De cada candidato se necesita saber el nombre y el número de votos que lleva. Realizar la clase con los atributos necesarios y los métodos que permitan conocer la cantidad de votos de cada candidato, así como el nombre del candidato más votado y cuantos votos ha conseguido. Crear la clase PruebaVotacion donde se creen varios candidatos, se simule unas pequeñas elecciones y se muestre el resultado.

Ejercicio n.13 (array de objetos)

Crear una clase PruebaPlaneta2 que permita crear un array de 5 objetos de la clase Planeta creada en el ejercicio 2. Usar un bucle para generar los objetos pidiendo por teclado el nombre del planeta, y asignar de manera aleatoria valores para la órbita (kilómetros) y traslación (días). Finalmente recorrer el array para mostrar los datos de todos los planetas.

Ejercicio n.14 (array de objetos)

Crear una clase PruebaVotacion2 que permita crear un array de 5 candidatos de la clase Planeta creada en el ejercicio 11. Usar un bucle para generar los objetos pidiendo por teclado el nombre del candidato. Utiliza otro bucle para generar de manera aleatoria 100 votos y finalmente recorrer el array para mostrar los votos de todos los candidatos, así como el candidato ganador.

Ejercicio n.15 (Array estático de objetos)

Se desea representar los coches que hay en un concesionario, de cada coche se almacenara matricula, modelo, motor, precio y color. Se desea saber en todo momento la cantidad de dinero invertida en el concesionario que será la suma del valor de todos los coches. Se desea disponer de un método para insertar un nuevo coche aportando todos los datos del mismo, y otro para obtener los datos de un coche (recibe un entero correspondiente a la posición en el array estático y devuelve un objeto coche). También se tendrá un método para listar por pantalla todos los coches del concesionario mostrando todos sus datos. Realizar una clase con Main para probar todos los métodos de la clase Coche.