

1. Crea una clase Rectangulo. Dispondrá de un constructor que cree un rectángulo partiendo de la base y la altura, de forma que su vértice inferior izquierdo esté en (0,0). La clase también incluirá un método para calcular la superficie y otro que desplace el rectángulo en el plano.

Probar los métodos de la clase creando varios rectángulos y operando con ellos en un programa.

2. Crea una clase llamada Cuenta que tendrá los siguientes atributos: id (número entero) y cantidad (puede tener decimales). Construye los siguientes métodos para la clase:

- Un constructor.
- Setter y getter de id y getter de cantidad.
- `mostrar()`: Muestra los datos de la cuenta.
- `ingresar(cantidad)`: se ingresa una cantidad a la cuenta, si la cantidad introducida es negativa, no se hará nada.
- `retirar(cantidad)`: se retira una cantidad a la cuenta. La cuenta puede estar en números rojos.

Probar los métodos de la clase creando varios objetos y operando con ellos en un programa.

3. Crea una clase NIF que se usará para mantener DNIs con su correspondiente letra. Los atributos serán el número de DNI (entero) y la letra que le corresponde (char). La clase dispondrá de los siguientes métodos:

Constructor sin argumentos que inicialice el nº de DNI a 0 y la letra a espacio en blanco (será un NIF no válido).

Constructor que reciba el DNI y establezca la letra que le corresponde.

Método que nos permita mostrar el NIF (ocho dígitos, un guión y la letra)

La letra se calculará con un método auxiliar (privado) de la siguiente forma: se obtiene el resto de la división entera del número de DNI entre 23 y se usa la siguiente tabla para obtener la letra que corresponde:

0 - T 1 - R 2 - W 3 - A 4 - G 5 - M 6 - Y 7 - F 8 - P 9 - D 10 - X 11 - B 12 - N 13 - J

14 - Z 15 - S 16 - Q 17 - V 18 - H 19 - L 20 - C 21 - K 22 - E

Probar los métodos de la clase creando varios objetos y operando con ellos en un programa.

4. Crea una clase Fecha con atributos para el día, el mes y el año de la fecha. Incluye, al menos, los siguientes métodos:

Constructor sin argumentos con el 1-1-1900 como fecha por defecto

Constructor parametrizado con día, mes y año.

`bisiesto()`: indicará si el año de la fecha es bisiesto o no.

`diasMes()`: devolverá el número de días del mes de la fecha.

valida(): comprobará si la fecha es correcta (entre el 1-1-1900 y el 31-12-2050); si el día no es correcto, lo pondrá a 1; si el mes no es correcto, lo pondrá a 1; y si el año no es correcto, lo pondrá a 1900. Será un método auxiliar (privado). Este método se llamará en el constructor parametrizado

siguiente(): pasará al día siguiente.

anterior(): pasará al día anterior.

igualQue(Fecha): indica si la fecha es la misma que la proporcionada.

menorQue(Fecha): indica si la fecha es anterior a la proporcionada.

mayorQue(Fecha): indica si la fecha es posterior a la proporcionada.

Probar los métodos de la clase creando varios objetos y operando con ellos en un programa.

5. Crea una clase Hora con atributos para las horas, los minutos y los segundos de la hora. Incluye, al menos, los siguientes métodos:

Constructor sin argumentos con el 00:00:00 como hora por defecto.

Constructor parametrizado con horas, minutos y segundos.

Getters y setters para los tres atributos

print(): mostrará la hora por pantalla.

aSegundos(): devolverá el número de segundos transcurridos desde la medianoche.

segundosDesde(Hora): devolverá el número de segundos entre la hora y la proporcionada como argumento.

siguiente(): pasará al segundo siguiente.

anterior(): pasará al segundo anterior.

igualQue(Hora): indica si la hora es la misma que la proporcionada como argumento.

menorQue(Hora): indica si la hora es anterior a la proporcionada como argumento.

mayorQue(Hora): indica si la hora es posterior a la proporcionada como argumento

Probar los métodos de la clase creando varios objetos y operando con ellos en un programa.