




LISTADO DE EJERCICIOS. CLASES Y OBJETOS.

1. Responde a las siguientes preguntas referentes a la inicialización de campos por defecto:
 - a. Si existen campos de tipo int, float, long... sin inicializar, su valor por defecto es: 
 - b. Si existen campos de tipo String sin inicializar, su valor por defecto es: 
 - c. Si existen campos de tipo int[] sin inicializar, su valor por defecto es: 
2. Crea la clase Menu. Mételo en el paquete utiles. Implementa al menos los siguientes métodos y atributos:
 - a. titulo
 - b. opciones
 - c. numOpciones
 - d. gestionar()
 - e. mostrar()
 - f. recogerOpcion()
3. Crea la clase "Circulo" que responda al siguiente comportamiento:
 - a. Un círculo puede crecer. Aumenta su radio.
 - b. Un círculo puede menguar. Decrementa su radio.
 - c. Un círculo me devuelve su área si se la pido.
 - d. Un círculo me dice su estado, por ejemplo "Soy un círculo de radio 0.5 metros. Ocupo un área de 0.7853981633974483 metros cuadrados" (método toString())
4. Crea una clase TestCirculo que cree una instancia de "Circulo", muestre su estado, lo haga crecer 27 veces, averigüe su área, lo haga decrecer 10 veces y muestre su estado final.
5. Crea una clase TestCirculoConMenu que muestre un menú para manipular una instancia de círculo.
6. Modifica la clase "Circulo" si no has tenido en cuenta que el radio no puede ser negativo. Supongamos que si el radio es 0 se convierte en un punto. Entonces el círculo reacciona y dice "Soy un mísero punto sin área"
7. Crea la clase Rectangulo y su respectivo TestRectangulo que nos permita manipular una instancia de rectángulo mediante un menú.
 - a. Un rectángulo no puede tener dimensiones negativas
 - b. Comprueba que el rectángulo no desaparezca (siempre un ancho y un alto mayor que cero).
8. Cuando se convierta en cuadrado, aparecerá un mensaje que diga "Ahora soy un cuadrado"
9. Crea la clase "Gato" que responda al siguiente comportamiento:
 - a. Un gato puede comer. En este caso incrementa su peso y dice "Rico rico..."

- b. Un gato puede jugar. En este caso decrementa su peso y dice “Qué diver...”
 - c. Un gato puede dormir (su estado natural). En este caso dice “Ronroneo. Zzz...”
 - d. Nuestros gatos son habladores. Nos pueden decir por ejemplo “Hola, soy Gardfield de raza Persa. Peso 4.597 kilos”
10. Crea una clase TestGato que cree una instancia de Gato, le dé de comer , juegue con él mucho (7 veces), lo haga dormir y finalmente nos hable.
11. Crea una clase TestGatoConMenu que muestre un menú para que juegue con el gato.
12. Modifica la clase Gato para que si engorda demasiado o adelgaza en extremo, advierta a su amo (“Tengo hambre!!!” o “Voy a reventar!!!”). En caso de llegar a un peso máximo o mínimo, el gato morirá (se saldrá del programa)
13. Crea la Clase Estudiante. Mételo en el paquete ies. Implementa al menos los siguientes métodos y atributos.
- a. nombre
 - b. apellido primero
 - c. apellido segundo
 - d. curso (sólo se admitirán los valores 1ESO, 2ESO, 3ESO, 4ESO, 1BACH, 2BACH, 1DAW, 2DAW)
 - e. setCurso()
 - f. incCurso(), que hará la promoción del alumno dentro de la etapa, que es:
 - f.i. 1ESO ->2ESO->3ESO->4ESO
 - f.ii. 1BACH ->2BACH,
 - f.iii. 1DAW -> 2DAW
14. Implementa la clase TestEstudiante() para probar la clase Estudiante: Mediante un menú se podrá crear un estudiante, borrarlo, mostrarlo y modificarlo. Tendrás al menos las clases:
- a. ies.Estudiante.java,
 - b. testEstudiante.TestEstudiante.java,
 - c. utiles.Menu.java. Deberás crear al menos un menú con las opciones “alta”, “baja”, “mostrar”, “modificar” y “salir”, y otro con las opciones “promocionar” o “cambiar de etapa”
15. Crea la clase Dado de seis caras (valores del 1 al 6). Mételo en el paquete casino. Implementa al menos los métodos
- a. toString().
 - b. lanzar()
16. Implementa la clase TestDado() para probar la clase Dado: Mediante un menú se lanza el dado tantas veces como se quiera, mostrándose su valor. Tendrás al menos las clases:

- a. casino.Dado.java,
 - b. testCasino.TestDado.java,
 - c. utiles.Menu.java. Deberás crear al menos un menú con al menos las opciones “lanzar dado” y “salir”
17. Implementa la clase JuegoDado(). Consiste en lanzar el dado. Ganará aquel que saque el mayor valor. Mediante un menú se lanza el dado tantas veces como se quiera, mostrándose su valor. Tendrás al menos las clases:
- a. casino.Dado.java,
 - b. casino.JuegoDado.java. Has de implementar al menos el método jugar(),
 - c. utiles.Menu.java. Deberás crear al menos un menú con las opciones “jugar” y “salir”
 - d. testCasino.TestJuegoDado.java, donde se creará el juego.