

## Unidad-11: Utilización avanzada de clases.

### ACTIVIDADES - I

1. Implementa la clase caballo. De un caballo queremos conocer su nombre, color, edad, así como las carreras que ha ganado. Además, queremos implementar para esta clase las acciones más comunes de un caballo: cabalgar, relinchar y rumiar. Realiza su implementación en Java y prueba el siguiente código:

```
public class PruebaCaballo {  
  
    public static void main(String[] args) {  
  
        Caballo a = new Caballo("Avra", "marrón moteado", 6, 24);  
        Caballo l = new Caballo("Lykos", "negro", 8, 61);  
  
        System.out.println("Hola, me llamo " + a.getNombre());  
        a.cabalga();  
        a.relincha();  
  
        System.out.println("Hola, yo soy " + l.getNombre());  
        l.rumia();  
        l.cabalga();  
    }  
}
```

2. Crea la clase vehículo, así como las clases bicicleta y coche como subclases de la primera. Para los vehículos queremos conocer los vehículos creados, los kilómetros totales recorridos por todos los vehículos y los kilómetros recorridos por cada vehículo. De las bicicletas nos interesa el número de piñones que tiene y de los coches su cilindrada. Genera un método específico para la bicicleta que indique estás haciendo el caballito y otro para el coche que indique que está quemando las ruedas. Para probarlo te puedes basar en el siguiente menú:

1. Hacer Km con una bicicleta.
  2. Hacer el caballito con una bicicleta
  3. Hacer Km con un coche
  4. Quemar rueda con un coche
  5. Ver km de una bicicleta
  6. Ver km de un coche
  7. Ver km total
  8. Salir
- Elige una opción (1-8):

Puedes tener una cantidad de bicicletas y de coches en una lista de una longitud determinada e incluir en ella uno de esos vehículos. Por ejemplo, de forma aleatoria indicamos si creamos una bicicleta o un coche, al mismo tiempo que pedimos los valores de sus atributos (si fuera necesario claro). Cuando queremos indicar los Km que hace una bicicleta o un coche mostraremos el tipo

elegido para indicarle con que bicicleta o coche voy a recorrer kilómetros. Lo mismo para hacer el caballito o quemar ruedas.

3. Crea la clase fracción y los métodos invertir, simplificar, multiplicar, dividir, sumar y restar.
4. En una pizzeria se necesita saber el tamaño (mediana o familiar), el tipo (margarita, cuatro quesos o funghi) i su estado (pedida o servida). También se quiere almacenar el número total de pizzas que se han pedido y que se han servido. Siempre que se pide (se crea) una pizza su estado es "pedida". Implementa el modelo del sistema si a partir del siguiente código del programa principal, la salida ha de ser la que se muestra a continuación:

```
public class PedidosPizza {
    public static void main(String[] args) {
        Pizza p1 = new Pizza("margarita", "mediana");
        Pizza p2 = new Pizza("funghi", "familiar");
        p2.sirve();
        Pizza p3 = new Pizza("cuatro quesos", "mediana");
        System.out.println(p1);
        System.out.println(p2);
        System.out.println(p3);
        p2.sirve();
        System.out.println("pedidas: " + Pizza.getTotalPedidas());
        System.out.println("servidas: " + Pizza.getTotalServidas());
    }
}
```

pizza margarita mediana, pedida  
pizza funghi familiar, servida  
pizza cuatro quesos mediana, pedida  
esa pizza ya se ha servido  
pedidas: 3  
servidas: 1

Podéis mejora i/o ampliar la aplicación con aquellas funciones que consideréis.

5. Crea la clase tiempo con los métodos suma y resta. Los objetos de la clase tiempo son intervalos de tiempo que necesitan para ser contruidos de las horas, minutos y segundos respectivamente. Por ejemplo:

Tiempo t = new Tiempo(1,20,30);

Crea el método toString para ver los intervalos de tiempo de la forma:

10h 35m 5s

Si se suman por ejemplo 30m 40s y 35m 20s el resultado debería de ser 1h 6m 0s. Realiza un programa de prueba para comprobar que la clase funciona bien.

6. Queremos gestionar la venta de entradas (no numeradas) de Expocoches Málaga que tiene 3 zonas, la sala principal con 1000 entradas disponibles, la zona de compra-venta con 200 entradas disponibles y la zona VIP con 25 entradas disponibles. Hay que controlar que existen entradas antes de venderlas.

El menú del programa debe ser el que se muestra a continuación. Cuando elegimos la opción 2, se nos debe preguntar para qué zona queremos las entradas y cuántas queremos. Lógicamente, el programa debe controlar que no se puedan vender más entradas de la cuenta.

1. Mostrar número de entradas libres
2. Vender entradas
3. Salir