

EJERCICIOS HERENCIA 02/02/2021

EJERCICIO 1

Crea la clase Vehiculo , así como las clases Bicicleta y Coche como subclases de la primera. Para la clase Vehiculo , crea los atributos de clase vehiculosCreados y kilometrosTotales , así como el atributo de instancia kilometrosRecorridos . Crea también algún método específico para cada una de las subclases. Prueba las clases creadas mediante un programa con un menú como el que se muestra a continuación:

VEHÍCULOS

=====

1. Anda con la bicicleta
 2. Haz el caballito con la bicicleta
 3. Anda con el coche
 4. Quema rueda con el coche
 5. Ver kilometraje de la bicicleta
 6. Ver kilometraje del coche
 7. Ver kilometraje total
 8. Salir
- Elige una opción (1-8):

EJERCICIO 2:

Las amebas son seres unicelulares de forma cambiante ya que carecen de pared celular. Fagocitan cualquier cosa que se les pone por delante. Crea la clase Ameba con el atributo peso , un número entero que indica los microgramos que pesa el bicho. Al tratarse de una unidad tan pequeña, no se tienen en cuenta los decimales, será un dato entero. Cuando Dios crea una ameba de la nada – new Ameba() – su peso es de 3 microgramos. Al comer, va incrementando su peso; gasta un microgramo en el proceso de fagocitar y el resto hace que aumente de peso.

Por ejemplo, si come una partícula de 6 microgramos – por ej. miAmeba.come(6) – engordaría 5 microgramos. Una ameba se puede comer a otra ameba. En este caso, sucede lo mismo que anteriormente, se gasta un microgramo en el proceso de fagocitado y el resto lo engorda la ameba que come. Por ejemplo, si una ameba de 7 microgramos se come a una de 4, acaba pesando 10 microgramos. La ameba comida no se destruye sino que se quedaría con un peso de 0 microgramos, una pena de ameba vamos.

Posteriormente, una ameba comida podría recuperarse si ella misma come algo. Nótese que el método come está sobrecargado.

Programa principal:

```
Ameba a1 = new Ameba();
a1.come(2);
System.out.println(a1);
Ameba a2 = new Ameba();
a2.come(4);
System.out.println(a2);
a1.come(a2);
System.out.println(a1);
System.out.println(a2);
a2.come(3);
```

```
System.out.println(a2);
```

Salida:

Soy una ameba y peso 4 microgramos.
Soy una ameba y peso 6 microgramos.
Soy una ameba y peso 9 microgramos.
Soy una ameba y peso 0 microgramos.
Soy una ameba y peso 2 microgramos.

EJERCICIO 3:

La empresa El Corte Inglés nos ha encargado una aplicación para gestionar las tarjetas regalo. Como primer paso para implementar la aplicación, es necesario crear la clase principal. Implementa la clase TarjetaRegalo. Cuando se crea una nueva tarjeta, se le da un saldo y se asigna de forma automática un número de 5 dígitos. Si se intenta gastar más dinero del que tiene la tarjeta, se debe mostrar un mensaje de error. Dos tarjetas regalo se pueden fusionar creando una nueva tarjeta con la suma del saldo que tenga cada una y un nuevo número aleatorio de 5 cifras. Al fusionar dos tarjetas en una, las dos tarjetas originales se quedarían con 0 € de saldo.

Programa principal:

```
TarjetaRegalo t1 = new TarjetaRegalo(100);  
TarjetaRegalo t2 = new TarjetaRegalo(120);  
System.out.println(t1);  
System.out.println(t2);  
t1.gasta(45.90);  
t2.gasta(5);  
t2.gasta(200);  
t1.gasta(3.55);  
System.out.println(t1);  
System.out.println(t2);  
TarjetaRegalo t3 = t1.fusionaCon(t2);  
System.out.println(t1);  
System.out.println(t2);  
System.out.println(t3);
```

Salida:

Tarjeta no 67324 – Saldo 100.00€
Tarjeta no 02788 – Saldo 120.00€
No tiene suficiente saldo para gastar 200.00€
Tarjeta no 67324 – Saldo 50.55€
Tarjeta no 02788 – Saldo 115.00€
Tarjeta no 67324 – Saldo 0.00€
Tarjeta no 02788 – Saldo 0.00€
Tarjeta no 59032 – Saldo 165.55€

EJERCICIO 4:

Se quiere informatizar una biblioteca. Crea las clases Publicacion, Libro y Revista. Las clases deben estar implementadas con la jerarquía correcta. Las características comunes de las revistas y de los libros son el código ISBN, el título, y el año de publicación. Los libros tienen además un atributo prestado. Cuando se crean los libros, no están prestados. Las revistas tienen un número.

La clase Libro debe implementar la interfaz Prestable que tiene los métodos presta, devuelve y estaPrestado.

Programa principal:

```
Libro libro1 = new Libro("123456", "La Ruta Prohibida", 2007);
Libro libro2 = new Libro("112233", "Los Otros", 2016);
Libro libro3 = new Libro("456789", "La rosa del mundo", 1995);
Revista revista1 = new Revista("444555", "Año Cero", 2019, 344);
Revista revista2 = new Revista("002244", "National Geographic", 2003, 255);
System.out.println(libro1);
System.out.println(libro2);
System.out.println(libro3);
System.out.println(revista1);
System.out.println(revista2);
libro2.presta();
if (libro2.estaPrestado()) {
System.out.println("El libro está prestado");
}
libro2.presta();
libro2.devuelve();
if (libro2.estaPrestado()) {
System.out.println("El libro está prestado");
}
libro3.presta();
System.out.println(libro2);
System.out.println(libro3);
```

Salida:

```
ISBN: 123456, título: La Ruta Prohibida, año de publicación: 2007 (no prestado)
ISBN: 112233, título: Los Otros, año de publicación: 2016 (no prestado)
ISBN: 456789, título: La rosa del mundo, año de publicación: 1995 (no prestado)
ISBN: 444555, título: Año Cero, año de publicación: 2019
ISBN: 002244, título: National Geographic, año de publicación: 2003
El libro está prestado
Lo siento, ese libro ya está prestado.
ISBN: 112233, título: Los Otros, año de publicación: 2016 (no prestado)
ISBN: 456789, título: La rosa del mundo, año de publicación: 1995 (prestado)
```