

Evaluación Final
Fundamentos de Programación Orientada a Objetos
II Término 2013-14

Desarrollar clases en Java para controlar lo indicado en el siguiente caso: Se tiene cuatro clases: Conductor, Taxi, Taxímetro y Carrera.

De la clase Taxi se requiere conocer el código de placa (tipo String), el kilometraje (tipo int), el conductor (tipo Conductor) y el taxímetro asignado (tipo Taxímetro).

Del Conductor se tiene la cédula (tipo String), el nombre (tipo String) y número de citas (tipo int).

Del Taxímetro se requiere conocer un número identificador del taxímetro (tipo int), el número de facturas emitidas (tipo int) y las carreras registradas en el taxímetro (vector de objetos tipo Carrera).

De la Carrera se requiere la hora de inicio (tipo String) y hora fin (tipo String) de la carrera, el kilometraje (tipo int) y el valor cobrado (tipo double).

La clase Taxi tiene dos subclases: Normal, Especial. Estas dos clases deben sobrecargar la operación abstracta devolverFactor "public abstract double devolverFactor()" definida en la clase Taxi. El método devolverFactor de la clase Normal devuelve 1 y el de la clase especial devuelve 1.2

Debe cumplir con los principios de encapsulación, herencia y polimorfismo.

Incluir al menos 3 constructores por cada clase. Uno de ellos debe recibir como parámetro un objeto del mismo tipo.

Se debe poder ejecutar el siguiente código:

```
Conductor c1 = new Conductor ("0123456789", "Carlos P ", 3);  
Taximetro tx1 = new Taximetro (101, 0);  
Taxi taxi1 = new Especial ("GTF0234", 1300, c1, tx1);
```

```
Carrera cr1 = new Carrera ("10:30", "10:45", 6); // Debe calcular el valor a cobrar **  
Carrera cr2 = new Carrera (12, 30, 12, 45, 6); // Debe calcular el valor a cobrar **
```

```
tx1.agregarCarrera (cr1); // agrega la carrera al taxímetro instalado en el taxi  
tx1.agregarCarrera (cr2); // agrega la carrera al taxímetro instalado en el taxi
```

```
taxi1.listarCarreras (); // muestra los datos de todas las carreras realizadas por el taxi
```

** El valor a cobrar en la carrera se calcula multiplicando el kilometraje recorrido en la carrera por el valor por Kilometro definido en una constante de clase por el valor devuelto por el método devolverFactor().