

1. Statement of the problem

There is a BlueJ project with an incomplete set of classes and interfaces (which should not be modified). The student must implement the following components to complete the project:

1. Add a new interface *Posicion*. This interface allows us to know the position of an object in the cartesian plane using the following methods:

```
public double obtenerCoordenadaX();  
public double obtenerCoordenadaY();
```
2. Add a new class *Vehiculo*.
 - All cars are vehicles. Therefore, there must exist a relation between class *Vehiculo* and class *Coche* that captures this reality.
 - As it happens with figures, all vehicles have a position. Therefore, this class must properly relate with interface *Posicion*.
 - You must take advantage of inheritance, so that all cars and other kinds of vehicles inherit attributes and use methods of their superclasses.
 - You must set a mechanism to avoid that objects of *Vehiculo* class exist at runtime.
3. Add a class *Comparador* to compare the positions of any two objects that implement interface *Posicion* (read Pag. 7 of Practice 2). Class *Comparador* should store into two attributes two objects of the same class that should be passed as parameters to its constructor. It should have a method to know whether the two stored objects have the same position. You can obtain information about this class looking at the code of *UsoComparador* class. This class should continue working properly if, in the future, we define new types of vehicles.
4. Perform the required changes in the project so that it compiles without errors and all the elements work correctly.

2. Test

If the project is successfully completed, the execution of the *main* method from the *UsoComparador* class produces the following result: *true false*.

3. Available time to solve the problem

The student has 45 minutes to solve the problem.