



Programación Orientada a Objetos

TPO 2 - Herencia y Polimorfismo

Una empresa de videojuegos planea crear la mejor experiencia de carreras del mercado. Una de sus premisas, es que quiere diferenciar el tipo de auto: si es a gas, a nafta o eléctrico.

En su primera versión, sólo necesita implementar autos a gas y autos eléctricos. En la segunda iteración del desarrollo, planean agregar los autos convencionales a nafta.

De todos los autos es necesario saber su color, marca, cantidad de puertas, velocidad máxima, aceleración, kmRecorridos. Sin embargo, si es un auto de gas, es necesario saber la cantidad de gas que puede tener (un entero representando los galeones máximos), además de la cantidad de gas actual; si es un auto eléctrico, necesitamos saber la cantidad máxima de kms que puede recorrer con una sola carga (al 100%), además de la carga actual.

1. Dibuje el diagrama de clases
2. Indique los atributos de cada clase e implemente en Smalltalk. Implemente los constructores.
3. Realice el diagrama de secuencia mensaje **“andar: cantKm”**
 - Debe incrementar la cantidad de km recorridos
 - Si es un auto eléctrico, la carga actual debe bajar teniendo en cuenta el cantKm y el cantMaxKm. Por ejemplo: si cuando tiene 100% puede recorrer hasta 200km, y se invoca el mensaje *andar: 20km*, el porcentaje actual debe ser $100\% - (20km * 100\% / 200km) \rightarrow 100\% - 10\% = 90\%$
 - Si es un auto a gas, se debe restar 1 galeón cada 30km.
 - **Importante:** Para ambos casos, recuerde verificar si **puede avanzar** (por ejemplo, si el auto tiene 10% de energía, no puede andar kms que superen esa energía).
4. Indique el **polimorfismo utilizado** en el punto 3
5. Realice un diagrama de secuencia **alternativo** del punto 3, considerando otro tipo de polimorfismo
6. Implemente en Pharo el punto 3