

# PROGRAMACIÓ ORIENTADA A OBJECTES

## TRIMESTRE 2 – CURS 2021/2022

### PRÀCTICA 2. SESSIÓ 3

**Objectiu:** POO. Classes: **Herència**. **Mètodes abstractes** i redefinició de mètodes i **Polimorfisme**. Ús de l'operador de **càsting**. Ús de l'operador **instanceof**. L'atribut **super**.

**Durada:** Una sessió

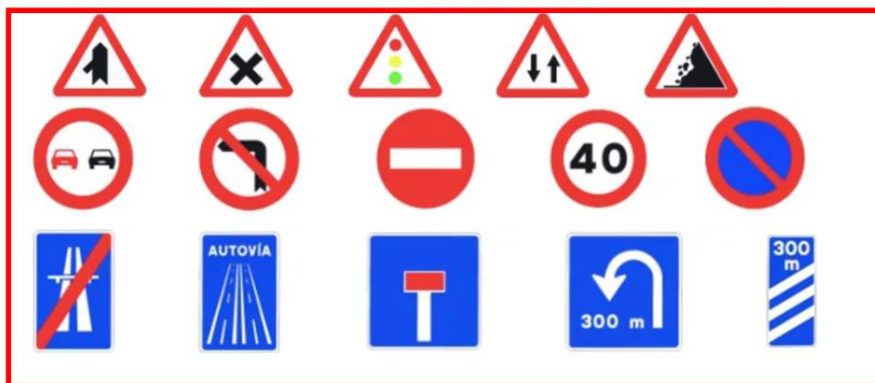
**Lliurament:** Llistat imprès dels fonts i penjar el projecte al Moodle

**Data Lliurament:** Abans de la Pràctica 3

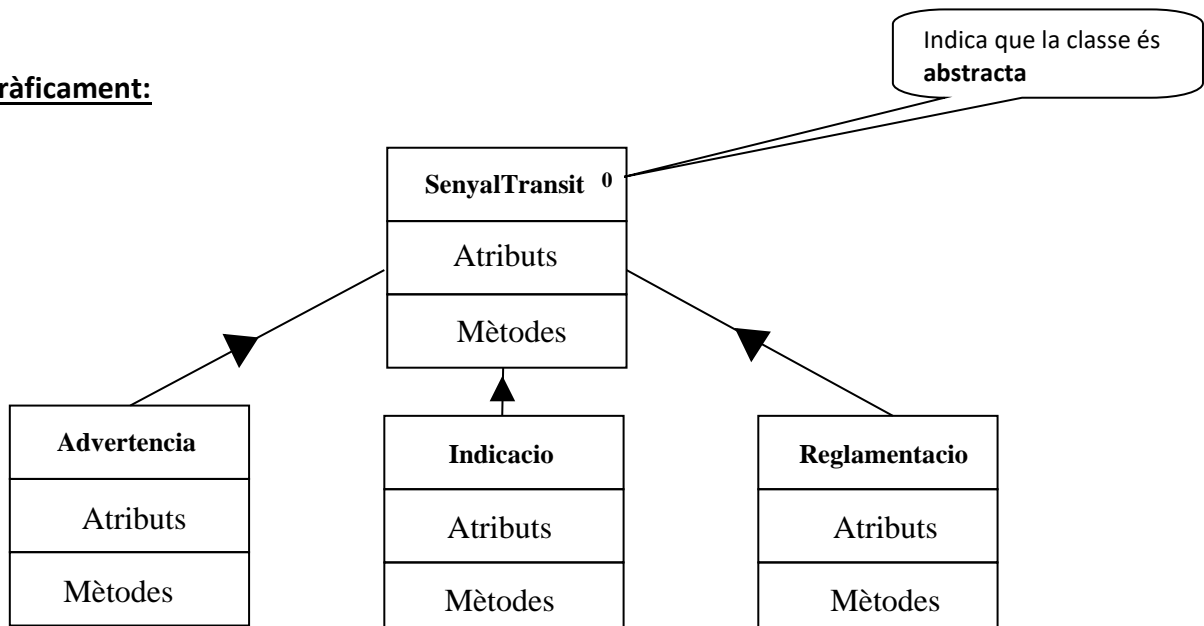
**Important:** **Imprescindible per aprovar la pràctica** seguir les pautes de l'enunciat.

#### Enunciat

L'objectiu és especialitzar la classe SenyalTransit, una subclasse per cada tipus de senyal.



#### Gràficament:



## Primera part: La classe SenyalTransit

### Afegim atributs i mètodes:

- Els senyals de trànsit tenen unes dimensions que poden variar en funció de si han de ser vistes des d'un vehicle en moviment, segons la carretera on estan ubicades, característiques de la circulació i en especial de la velocitat. Tot i això hi ha unes dimensions estàndards:
  - Senyals circulars 50 centímetres de diàmetre
  - Senyals triangulars 70 centímetres de costat
  - Senyals rectangulars 132 x 93 centímetres

La classe ha de oferir una **constant pública de classe** per a cadascuna d'aquestes dimensions (4).

- Tres atributs de classe a la classe *SenyalTransit* per emmagatzemar la quantitat de senyals de cada tipus que s'han creat (advertència, reglamentació o indicació), seran comptadors. Atributs de tipus **enter** amb visibilitat **protected** amb l'objectiu de tenir-hi accés en les seves classes derivades. La resta d'atributs han de mantenir la seva visibilitat de **private**. Determina el lloc adient on fer la inicialització, l'increment es farà en el corresponent constructor de les classes derivades.
- Un mètode públic `getXXX` per cadascun dels atributs de classe comptadors, per poder consultar el seu valor.
- Un mètode que visualitzi a pantalla totes les dades que emmagatzemen els atributs d'instància d'un objecte *SenyalTransit*.

```
public void visualitza()
```

- Aprofitant que totes els senyals tenen una superfície, la classe ha de tenir un **mètode abstracte** que en calculi la seva àrea:

```
public abstract float area()
```

Una conseqüència d'afegir aquest mètode és que provoca que la classe *SenyalTransit* es converteixi en una classe abstracta. I per tant, no se'n podran crear objectes.

## Segona part: les subclasses Advertencia, Indicacio i Reglamentacio

Crear les classes **Advertencia**, **Indicacio** i **Reglamentacio** com a subclasses de la classe *SenyalTransit*.

### Afegir a cada classe derivada:

- Un atribut privat per emmagatzemar l'explicació del significat del senyal:

```
private String significatSenyal;
```
- Un o dos atributs per emmagatzemar les mides del senyal. Pels senyals **rodons** només cal emmagatzemar el seu **diàmetre**, pels **rectangulars** o quadrats la mida de **cada costat** i pels **triangulars** la mida **d'un costat** (són triangles de 3 costats iguals, triangles equilàters).
- Els corresponents mètodes `getXXX` per consultar el valor de cadascun dels atributs afegits.
- Un constructor que rebi com a paràmetres, el codi, la ubicació, l'any de col·locació i la descripció del senyal. **Cada subclasse informarà a la superclasse del seu tipus (imprescindible usar els atributs estàtics de la classe SenyalTransit), que no ha de ser un paràmetre del constructor. El primer atribut propi s'inicialitza amb la**

**informació que arriba via paràmetre. Les mides seran les estàndards, és a dir els valors que emmagatzemen les constants estàtiques de la superclasse.**

**Recordeu que** el mètode haurà d'incrementar el corresponent atribut comptador en funció del tipus de senyal que es crea.

- **Sobrecarrega del constructor**, en aquest cas s'ha de proporcionar com a paràmetre(s) la mida del senyal, és a dir a més dels paràmetres que té el constructor previ arribaran paràmetres envers les dimensions del senyal que es crea. Si el senyal no és del tipus indicació serà un paràmetre més, en cas d'indicació seran 2, alçada i amplada. **Imprescindible** que aquest constructor invoqui al precedent o viceversa amb l'objectiu de reaprofitar codi.
- Implementar el mètode abstracte heretat `public float area()`. Ha de calcular l'àrea del senyal, sabeu que en funció de la forma cal aplicar una fórmula matemàtica u altra.
- **Redefiniu** el mètode `public void visualitza()` per a què també visualitzi el valor dels atributs propis afegits a la classe derivada. **Imprescindible** que la implementació d'aquest mètode invoqui al mètode que té el mateix nom de la superclasse.

### Tercera part: La classe Prova

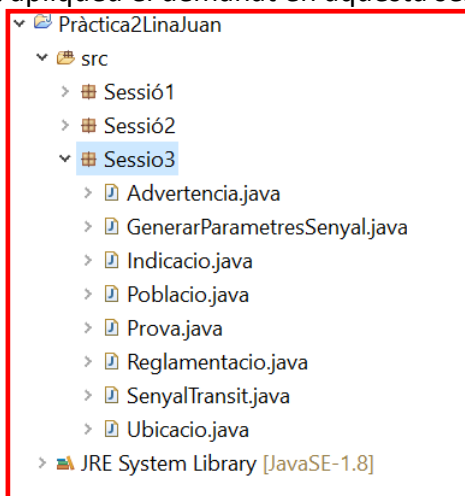
Per provar el codi desenvolupat, escriviu un mètode **main** dins de la classe Prova que ha de posar de manifest el funcionament del **polimorfisme**. Decidiu vosaltres la funcionalitat del programa però com a mínim ha de tenir un mètode ajudant que tingui paràmetres de tipus objecte. Heu de treballar amb mínim 20 senyals, creats el més aleatòriament possible i emmagatzemant les referències en una **taula de referències a objectes SenyalTransit**.

En aquesta sessió, **sí s'haurà de lliurar aquesta classe**.

```
SenyalTransit [] senyals = new SenyalTransit[20];  
// creació dels senyals el més aleatòriament possible
```

### Organització

Dins del mateix projecte creeu un nou paquet corresponent a la sessió 3 i per començar **copieu el mateix contingut que el de la sessió 2**. A les classes d'aquest tercer paquet apliqueu el demanat en aquesta sessió.



### Què se us subministra?

L'enunciat amb les especificacions i línies a seguir per el desenvolupament del disseny de classes.

### Què s'ha de lliurar i com?

S'ha de lliurar la carpeta que conté el projecte Eclipse amb el vostre desenvolupament de la pràctica. La carpeta s'ha de lliurar amb tot el seu contingut i comprimida amb ZIP o RAR.

També s'ha de lliurar un **llistat en paper** del codi desenvolupat. El format de lliurament d'aquest codi ha de seguir el patró indicat en la presentació de l'assignatura: amb portada, índex, número de pàgina, tabulació ... En **aquest llistat** cal que indiqueu:

- **la distribució de la feina entre els dos estudiants. És a dir, el grau de participació de cada membre del grup en la realització d'aquesta activitat.**
- **Si el programa no funciona cal que indiqueu quina/es parts no funcionen explicant que és el que passa.**

### On s'ha de lliurar?

El lliurament del projecte es farà a través de la plataforma Moodle i no s'acceptarà cap altra via. Feu atenció a la data i hora límit.

**El lliurament en paper es farà directament a la professora a l'inici de la primera sessió de la pràctica 3.**

### Quan s'ha de lliurar?

El lliurament es podrà fer fins el **dia indicat a sota**. Tingueu present que a partir d'aquesta hora el sistema bloquejarà, de manera automàtica, la possibilitat de lliurament.

**Lliurament Moodle → 24 Febrer a les 23:50h**

**Lliurament en paper → 25 de Febrer inici sessió de pràctiques**

### Pauta de correcció:

En aquesta pràctica de tres sessions resoldreu un únic enunciat que s'anirà ampliant sessió a sessió.

**Aquesta sessió 3 de la pràctica 2 té un pes del 20% de la qualificació total de la pràctica 2.**

### Valoració (sobre 10 punts):

- 1.- Escriptura de les **classes derivades (7.5 punts)**
- 2.- Modificacions classe **SenyalTransit (1.5 punt)**
- 3.- La classe **Prova (1 punt)**

### La correcció valorarà:

- 1.- **La descomposició funcional** aplicada, tant per un reaprofitament del codi com per la claredat de la solució mostrada. És imprescindible que cada tasca amb una funcionalitat pròpia tingui el seu propi procediment. Reaprofitament de codi.
- 2.- L'aplicació dels **esquemes** adients (recorregut i cerca)
- 3.- **Eficiència** en quan a no fer càlculs repetits amb unes mateixes dades.

La qualificació dependrà dels ítems indicats prèviament. **Obligatòriament heu de seguir les pautes de programació** donades en l'escriptura del programa. La vostra solució s'ha d'ajustar a l'enunciat de l'exercici.