

PROGRAMACIÓ ORIENTADA A OBJECTES

TRIMESTRE 2 – CURS 2021/2022

PRÀCTICA 2. SESSIÓ 1

Objectius: POO. **Classes:** que encapsulen atributs i mètodes.
Instanciació d'objectes. Ús de **this**. Modificadors **final** i **static**.
Ús de **null**.

Durada: Tres sessions

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.

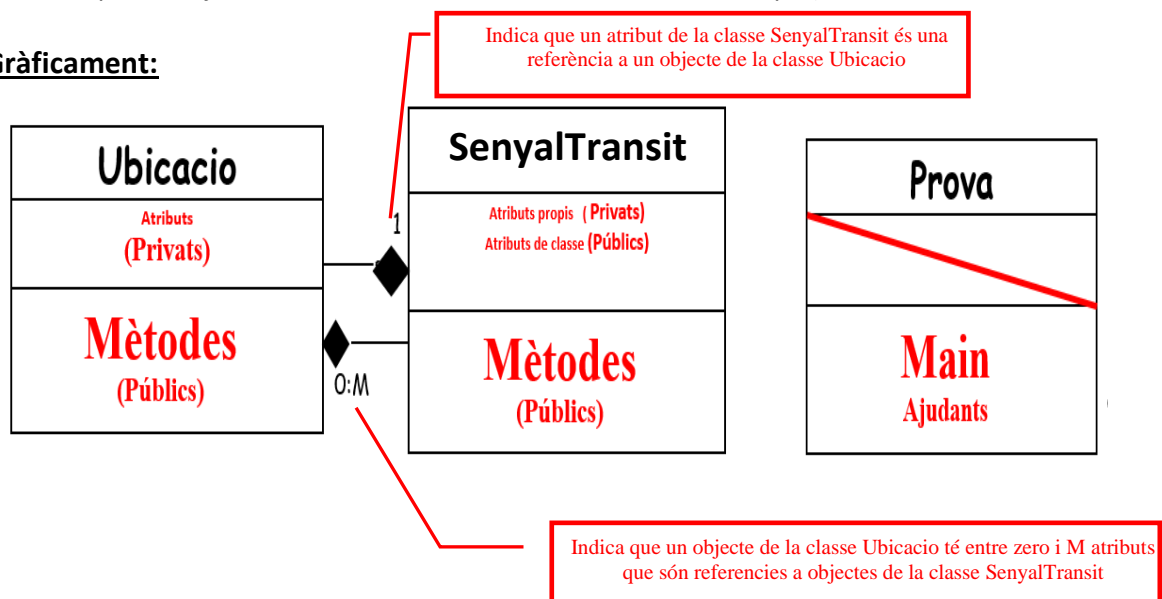
Treballarem al llarg de les tres sessions el mateix enunciat. Així com s'avanci en les sessions s'haurà d'ampliar i fer modificacions respecte el treballat a la sessió prèvia. Per tant, és imprescindible finalitzar el demanat en una sessió per poder continuar en la següent.

Enunciat

Cal fer un programa per a gestionar els senyals verticals de trànsit d'una gran ciutat. Com tots sabem els senyals de trànsit estableixen un llenguatge general per a tots els conductors. Aquest llenguatge té com a objectiu advertir, informar i reglamentar el comportament dels conductors segon indica la llei de Seguretat Vial i el Reglament General de Circulació. L'Ajuntament de la població, concretament la consellaria d'Obres i Serveis i Infraestructures juntament amb la Policia Local volen disposar d'una aplicació informàtica per controlar la ubicació dels senyals verticals que hi ha a la ciutat i el seu estat de conservació. Un senyal pot estar ubicat a la via pública, en aquest cas es vol saber en quina adreça o bé al dipòsit de la policia municipal o de l'ajuntament.

En aquesta primera sessió se'ns demana d'implementar tres classes. Dues d'elles estaran relacionades mitjançant una relació d'agregació (continença) en ambdós sentits, i la tercera contindrà únicament el main i procediments ajudants per provar i testejar el correcte funcionament del codi desenvolupat (aquesta classe no s'haurà de lliurar, l'estudiant l'ha de fer servir per testejar el correcte funcionament del codi desenvolupat).

Gràficament:



Important: els noms dels atributs i mètodes que estan explicitats en l'enunciat no es poden canviar. Els noms genèrics sí que els heu de canviar i donar-los el nom més adient. No podeu afegir ni atributs ni mètodes públics en cap de les classes i cal respectar la visibilitat indicada. Si no s'indica el contrari els atributs han de ser privats i els mètodes públics.

Primera part: la classe *Ubicacio*

En aquesta primera part de la pràctica cal definir la **classe** *Ubicacio*. Una ubicació ve caracteritzada per uns atributs i mètodes que cal definir en una classe tal com es descriu a continuació.

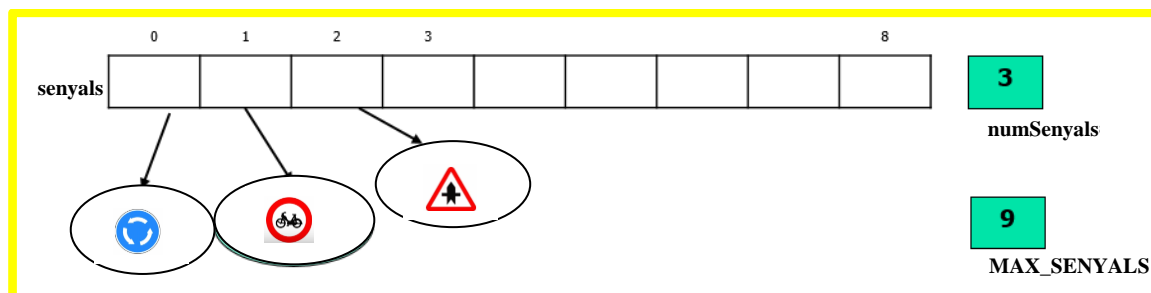
Pel que fa als atributs, un objecte *Ubicacio* té tres atributs, tots ells privats, preparats per emmagatzemar:

- **Atribut 1:** El nom de la via pública.
- **Atribut 2:** El número.
- **Atribut 3:** Per indicar si aquesta ubicació és o no una cruïlla.

I finalment altres atributs per emmagatzemar tots els senyals verticals de trànsit que estan en aquella ubicació, en concret:

- **Atribut 4:** l'anomenarem **senyals** que serà un vector que emmagatzemarà els senyals de trànsit que estan situats en aquesta ubicació, contindrà les referències a objectes de la **classe** *SenyalTransit* (descrita més endavant). **Totes les posicions plenes de la taula seran posicions contigües, així doncs les buides estaran també consecutives al final del magatzem.**
- **Atribut 5:** l'anomenarem **numSenyals**, de tipus enter que indicarà el número de senyals que hi ha en aquella ubicació. **Emmagatzemarà la dimensió real del vector senyals.**
- **Atribut 6:** anomenat **MAX_SENYALS** serà un atribut **constant** que s'inicialitzarà en la construcció i no podrà variar, i que representarà el número màxim de senyals verticals que pot haver a la ubicació. **Emmagatzemarà la dimensió declarada del vector.**

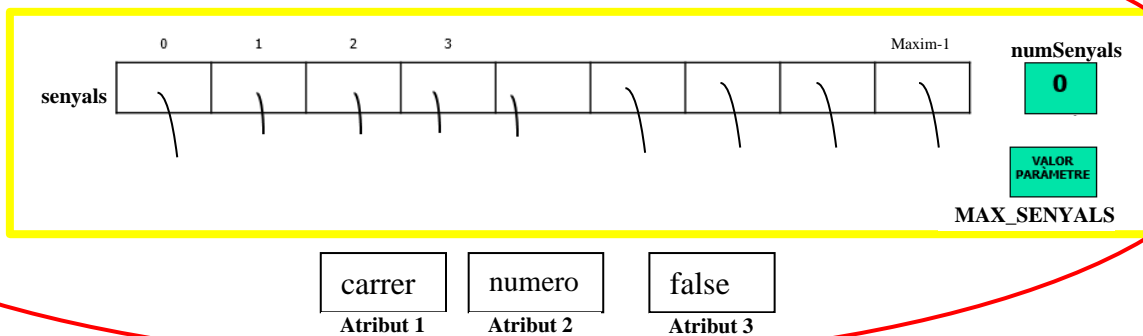
Exemple: Ubicació amb capacitat per a 9 senyals i que en aquest moment, en aquest exemple, en té 3:



Pel que fa als mètodes, un objecte de la classe *Ubicacio* disposa de:

- Un **mètode constructor** amb els paràmetres corresponents a les dades dels atributs de l'objecte que s'està construint, i que assignarà a cadascun dels atributs mencionats anteriorment. Els atributs corresponents a l'emmagatzematge de senyals s'han d'inicialitzar de la següent manera: la taula **senyals** s'ha de crear però ha de romandre buida i el **numSenyals** s'ha d'inicialitzar a 0 indicant que de moment la ubicació no té cap senyal. La resta d'atributs s'inicialitzaran amb el corresponent paràmetre excepte l'atribut que emmagatzema si és o no cruïlla que serà fals.

```
public Ubicacio(int Maxim, String carrer, int numero)
//MAX_SENYALS s'inicialitzarà amb el valor del primer paràmetre
```



- Un objecte *Ubicacio* té mètodes **getXXX** per consultar alguns dels atributs. Concretament:

```
public String getAtribut1();
public int getAtribut2();
public boolean getAtribut3();
public int getAtribut5();
public int getAtribut6();
```

- Un objecte *Ubicacio* només té un mètode set per canviar el contingut de l'**Atribut 3**. (recordeu que en la construcció sempre es construeix amb un valor de fals).

```
public void setAtribut3();
```

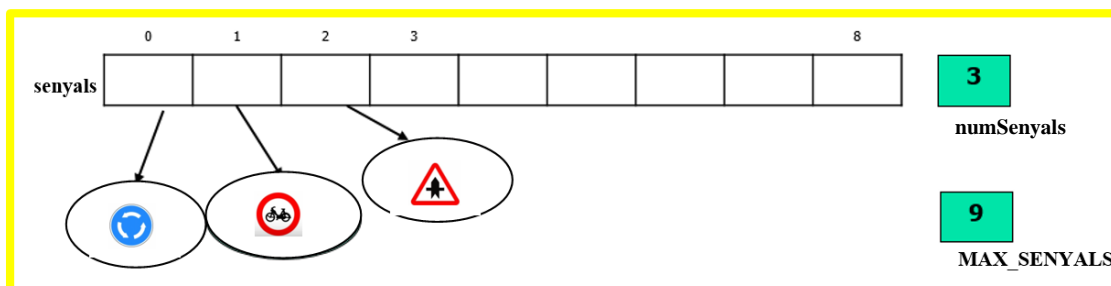
El mètode no té paràmetre, si l'atribut 3 té una valor de false el posa a true i si el té a true a false.

- Un mètode per poder obtenir un dels senyals que conté la ubicació. El mètode retorna la referència al senyal que ocupa la posició ièssima dins de l'atribut magatzem. El prototipus del mètode és:

```
public SenyalTransit getSenyal(int quin);
```

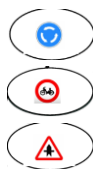
El paràmetre indica l'índex que ocupa el senyal dins del magatzem. Si en la posició indicada no hi ha cap senyal el mètode retorna una referència nul·la. També si la posició indicada en el paràmetre no existeix en el vector.

Exemple: Per la següent ubicació:



Les invocations:

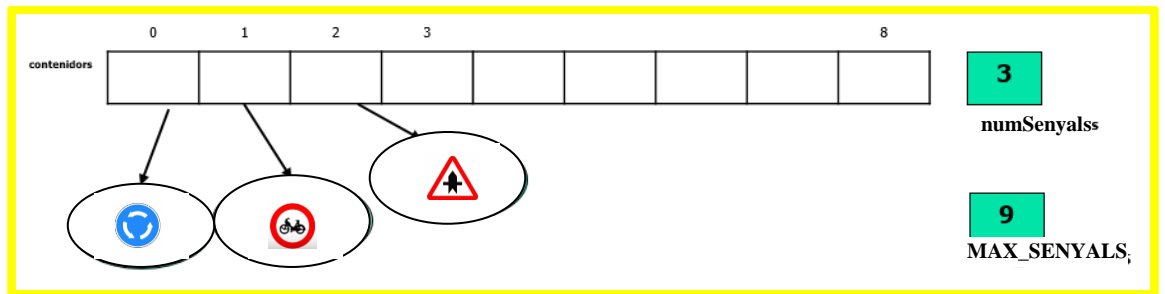
```
getSenyal(0) → retorna una referencia a
getSenyal(1) → retorna una referencia a
getSenyal(2) → retorna una referencia a
```



Per a qualsevol altra invocació retorna una referència nul·la.

- El mètode **boolean afegirSenyal(SenyalTransit c)** que rep com a paràmetre un objecte de la classe *SenyalTransit* per afegir-lo a la taula de senyals. Prèviament

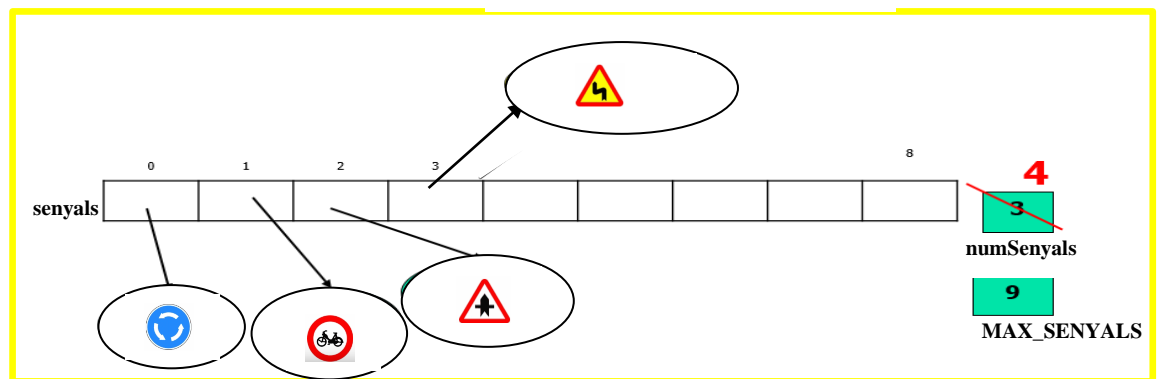
comprovarà que hi càpiga en el seu magatzem i que no hi hagi altre senyal a la ubicació amb el mateix codi (no s'admeten repetits). Així doncs, si es pot afegir retorna cert, altrament retorna fals. La senyal a afegir s'ha d'ubicar a **la primera posició buida** del magatzem.



Afegim un senyal



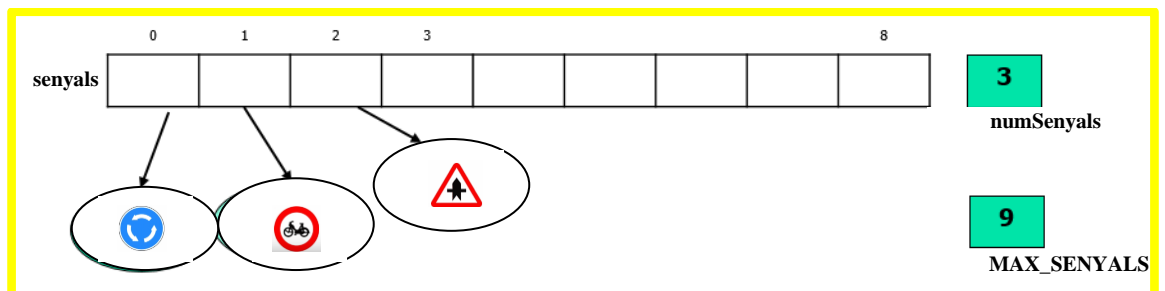
No està repetit



La posició a ocupar pel nou objecte és la d'índex indicat per l'atribut **numSenyals**.
Un cop afegit s'actualitza l'atribut augmentant una unitat.

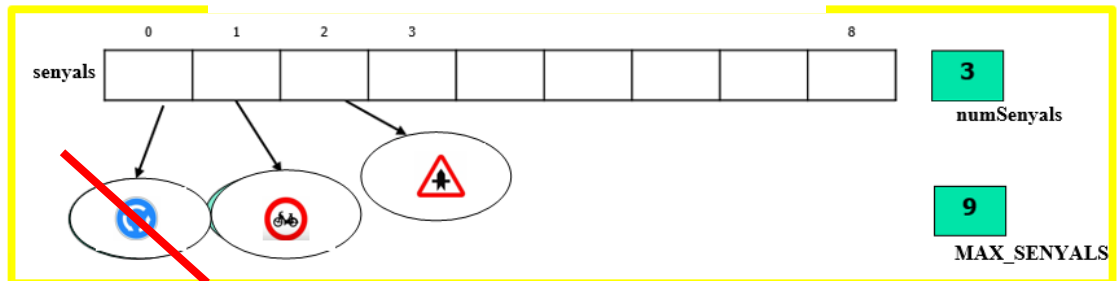
Imprescindible: El codi d'un senyal (descriu a la classe SenyalTransit més endavant) és un string i en conseqüència per comparar-los és imprescindible usar el mètode adient de la classe String: `compareTo`, `equals` ...

- El mètode **boolean treureSenyal(SenyalTransit c)** que rep un objecte de la classe SenyalTransit, el cerca en el magatzem senyals, i si el troba l'elimina del magatzem. Per saber si és el mateix *SenyalTransit* cal comparar el codi del senyal. Aquest mètode retorna cert si treu el senyal del magatzem, altrament retorna fals. **En cas d'eliminació s'ha d'ocupar la posició a buidar pel darrer senyal que hi ha al magatzem.**

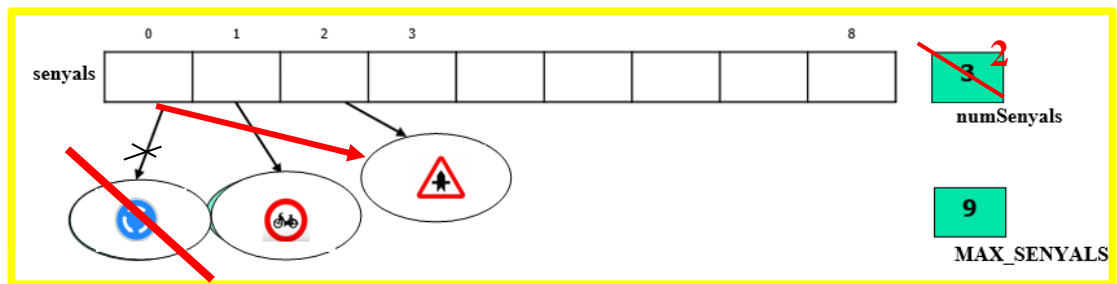




Eliminar un senyal. **Suposem es vol eliminar el que ocupa la posició d'índex 0.**



```
senyals[0]=senyals[numSenyals-1];  
numSenyals--;  
senyals[numSenyals]=null; //opcional
```



- 1.- Localitzar el senyal dins del magatzem
- 2.- En el lloc que ocupa el que es vol donar de baixa li posem la referència al darrer senyal del magatzem
- 3.- Actualitzar l'atribut **numSenyals**

- El mètode **String getSenyals()** que ha de retornar un String amb els codis de TOTS els senyals que té aquesta ubicació. Cada codi estarà separat del següent mitjançant el caràcter " - " .

Segona part: la classe SenyalTransit

Definir **la classe SenyalTransit** en el mateix paquet. Un senyal de trànsit ve caracteritzat per uns atributs i mètodes que cal implementar, tal com es descriu a continuació.

Pel que fa als atributs, la classe *SenyalTransit* tindrà **atributs constants de classe amb visibilitat pública**. Un per a cadascun dels tres tipus de senyals verticals, cada atribut estarà codificat mitjançant un valor numèric enter. Tipus de senyals verticals:

Senyals d'advertència de perill: indiquen proximitat de perill i tenen forma triangular.



Senyals de reglamentació: indiquen obligacions, limitacions o prohibicions i tenen forma rodona.



Senyals de indicació que donen informació d'utilitat al conductor i tenen forma quadrada o rectangular.



Quan un atribut es declara amb el modificador **static** l'atribut és de la classe i en conseqüència les seves instàncies el comparteixen. Per tant, per utilitzar un atribut estàtic d'una classe no cal disposar d'un objecte d'aquella classe es pot fer directament dirigint-nos a la pròpia classe. **Consulteu les transparències de teoria.**

En la implementació sempre que calgui treballar amb el tipus del senyal **serà imprescindible** usar aquests atributs de classe declarats.

A més del tres atributs de classe els objectes tindran atributs propis, els descrits a continuació.

Pel que fa als atributs, un objecte *SenyalTransit* tindrà els següents atributs propis amb visibilitat privada:

- **Atribut 1:** El codi del senyal, en aquest Ajuntament els tenen codificats mitjançant un valor alfanumèric seguint el següent format: "XXX-YYYY". XXX representen tres lletres corresponents als tres primers caràcters de la forma del senyal i YYYY són números que l'identifica. Així doncs un senyal pot tenir codificacions com ara: "ROD-3245", "TRI-3456" ... Aquest atribut és identificador, és a dir, no hi poden haver dos o més senyals amb el mateix codi.

Senyals d'avertència, el seu codi s'inicia per → TRI

Senyals de reglamentació, el seu codi s'inicia per → ROD

Senyals d'indicació (dos opcions segons siguin rectangulars o quadrades), el seu codi s'inicia per → QUA o REC

- **Atribut 2:** El tipus del senyal que s'ha de correspondre amb algun dels tres atributs constants de classe, ha de ser un valor enter.
- **Atribut 3:** La ubicació del senyal, que serà un objecte de la classe *Ubicacio*. Si el senyal està al magatzem de l'Ajuntament aquest atribut tindrà una referència nul·la.
- **Atribut 4:** L'any de col·locació, un enter de 4 xifres, que representa l'any que s'ubica al carrer el senyal. Si no està ubicat a la via pública, estarà al dipòsit, el valor d'aquest atribut serà un 0.
- **Atribut 5:** L'any de retirada, un enter de 4 xifres que representa l'any que es treu de la via pública el senyal i per tant s'ubica al dipòsit. Si encara està ubicat a la via pública al valor d'aquest atribut serà 0.

Adoneu-vos en referència als dos darrers atributs, que sempre un dels dos té un valor de zero en qualsevol objecte. Si el senyal està ubicat al carrer **Atribut 5** és zero, i si està al dipòsit **Atribut 4** és el que és zero.

Pel que fa als mètodes, un objecte de la classe *SenyalTransit* disposa:

- Un constructor amb **tants paràmetres** com atributs té l'objecte a construir. El mètode crea un objecte *SenyalTransit* amb els valors indicats en els paràmetres, que els podeu suposar sempre correctes.

```
public SenyalTransit(String codi, int tipus, Ubicacio ubicacio, int anyColocacio)
```

Aquest constructor **ubica** el senyal a la via pública indicada, ubicació donada en el tercer paràmetre. Si es dona el cas de què no és possible tal ubicació (mètode `afegirSenyal(SenyalTransit)` retorna un fals), només en aquest cas, el senyal serà ubicat al dipòsit i l'any de retirada vindrà donat en el darrer paràmetre. En cas contrari, el darrer paràmetre és l'any de col·locació.

- El mètode `String getTipusSenyal()` retorna una cadena informant del tipus de senyal. Possibles valors de retorn: **Advertència** o **Reglamentació** o **Indicació**.
- El mètode `boolean retirarViaPublica()` que simula la retirada de la via pública del senyal. Primerament invoca al mètode `treureSenyal(SenyalTransit)` de la classe *Ubicacio*, si aquest retorna fals el mètode no fa res i en cas contrari actualitza:
 - **Atribut 5** que emmagatzema l'any de retirada emmagatzemant-hi l'any actual obtenint-la del sistema.
Per obtenir la data d'avui (la del sistema de l'ordinador) i obtenir-ne l'any com un enter de 4 xifres, cal procedir de la següent manera (imprescindible importar el paquet que conté la classe *GregorianCalendar*):

```
GregorianCalendar avui = new GregorianCalendar(); //obtenir data d'avui  
int anyActual = avui.get(1); //aïllar l'any com un enter de 4 xifres
```

- **Atribut 3** posant la ubicació a nul
- **Atribut 4** que emmagatzema l'any de col·locació a la via pública, que ha d'emmagatzemar a partir de la retirada el valor de zero.
- El mètode `String getUbicacio()` si l'atribut que emmagatzema la ubicació del senyal no és nul, retorna la cadena que resulta de concatenar els atributs del nom i número de la via pública on està ubicat, altrament retorna la referència nul·la. Observeu que no retorna l'atribut que emmagatzema la ubicació, crea i retorna un String amb informació relativa a aquesta ubicació.
- El mètode `boolean canviarUbicacio (Ubicacio novaUbicacio)` realitza si és possible, el canvi d'ubicació modificant l'atribut ubicació. Si no es pot afegir el senyal en la nova ubicació (el mètode `afegirSenyal(SenyalTransit)` de la classe *Ubicacio* retorna fals), aquest mètode no fa res. Altrament treu el senyal de la ubicació actual, usant el mètode `treureSenyal(SenyalTransit)`, i canvia la ubicació del senyal. El mètode mitjançant el retorn indica si s'ha pogut o no efectuar el canvi d'ubicació.
- El mètode `String getAtribut1()` que retorna el codi del senyal.
- El mètode `Ubicacio getAtribut3()` que retorna la ubicació. Si està al dipòsit retorna una referència nul·la.

- El mètode `String getEstat()` que retorna una cadena amb l'estat del senyal, seguint la següent pauta:
 - Si té més de quatre anys retorna la cadena “vell”
 - Si té entre dos i quatre anys retorna “semi nou”.
 - Si té menys de dos anys retorna “nou”.

Els anys que té un senyal ve donat per la diferencia entre l'any de la data a dia d'avui i l'any de col·locació. Si el senyal no està a la via pública el mètode ha de retornar la cadena “retirat”.

- La classe *SenyalTransit* no té mètodes set.

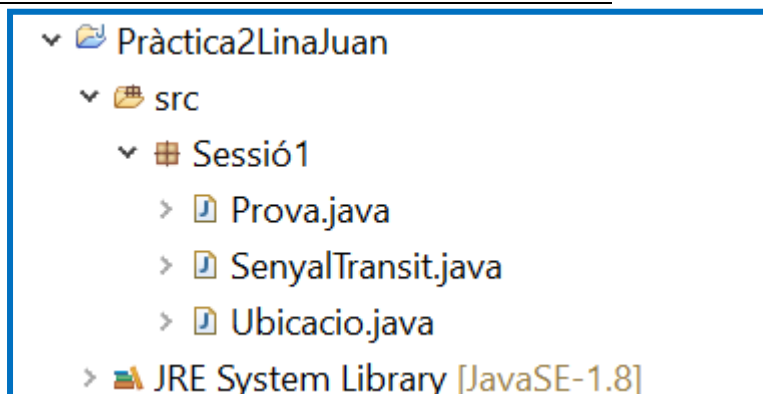
Tercera part: la classe Prova

En aquesta darrera part de la pràctica implementeu la **classe Prova** que ha de comprovar el correcte funcionament dels diferents mètodes sobre els objectes corresponents.

Decidiu vosaltres mateixos la funcionalitat, l'objectiu es que ho proveu tot per comprovar el correcte funcionament del codi desenvolupat en aquesta primera sessió.

Organització. Totes les pràctiques s'han de fer en grups de 2 estudiants

Seguireu la mateixa estructura que la usada a la pràctica 1. Cal crear un nou projecte amb nom **Pràctica2Cognom1Nom1&&Cognom2Nom2**. El projecte ha de tenir un paquet per cada sessió. En aquesta sessió hi ubicareu les tres classes demanades.



Què se us subministra?

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

Transparències classe String.

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 (no la classe Prova). 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 1 de la pràctica 2 té un pes del 50% de la qualificació total de la pràctica 2.

Valoració (sobre 10 punts):

- 1.- Escriptura de la classe **Ubicacio** (6 punts)
- 2.- Escriptura de la classe **SenyalTransit** (4 punts)

La correcció valorarà:

- 1.- L'aplicació dels **esquemes** adients (recoregut i cerca)
- 2.- Descomposició funcional i reaprofitament de codi en l'escriptura dels mètodes.
- 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.