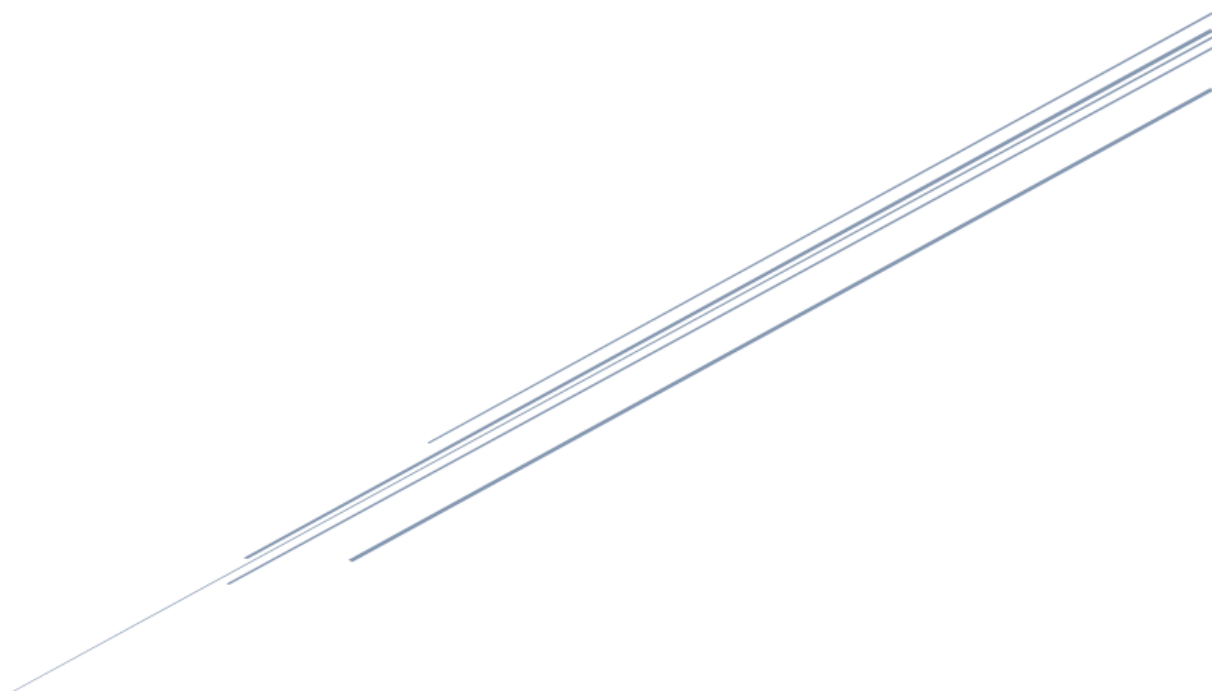


# Pràctica 2



Pol Rubio

Rems Nalivaiko

25 de feb. de 2022

103112 - PROGRAMACIÓ ORIENTADA OBJECTE

Fundació TecnoCampus Mataró-Maresme

## Índex:

<b>Sessió 1:</b>	<b>3</b>
<b>Sessió 2:</b>	<b>9</b>
<b>Sessió 3:</b>	<b>19</b>
<b>Organització de la feina:</b>	<b>30</b>
<b>Valoració:</b>	<b>30</b>

## Sessió 1:

Es tracta de fer un programa que amb un tauler de dimensions determinades i donat un estat inicial vagi aplicant les regles anteriors:

Codi:

### **SenyalTransit.java**

```
package Sessio1;
import java.util.GregorianCalendar;

public class SenyalTransit {

    public static final int Advertencia = 0;
    public static final int Reglamentacio = 1;
    public static final int Indicacio = 2;

    private String codi;
    private int tipus;
    private Ubicacio ubicacio;
    private int any_col=0;
    private int any_ret=0;

    public SenyalTransit(String codi, int tipus, Ubicacio ubicacio, int
anyColocacio) {
        this.codi = codi;
        this.tipus = tipus;

        if (ubicacio.afegirSenyal(this)) {
            this.ubicacio = ubicacio;
            any_col=anyColocacio;
        } else {
            ubicacio=null;
            any_ret=anyColocacio;
        }
    }
}
```

```
public boolean retirarViaPublica() {
    if(ubicacio.treureSenyal(this)) {
        GregorianCalendar avui = new GregorianCalendar();
        any_ret=avui.get(1);

        ubicacio=null;

        return true;
    }
    return false;
}

public boolean canviarUbicacio(Ubicacio novaUbicacio) {
    if(novaUbicacio.afegirSenyal(this)){
        ubicacio=novaUbicacio;
        return ubicacio.treureSenyal(this);
    }

    return false;
}

// GETs
public String getCodi() {return codi;}
public Ubicacio getUbicacio() { return ubicacio; } // nom en el doc:
getAtribut3()
public String getStrUbicacio() { // private o public // nom en el doc:
getUbicacio()
    if(ubicacio != null)
        return ubicacio.getNomVia() + ", " + ubicacio.getNumVia();
    return null;
}
public String getTipusSenyal() {
    switch(tipus) {
        case Advertencia:    return "Advertència";
        case Reglamentacio: return "Reglamentació";
        case Indicacio:      return "Indicació";
    }
}
```

```
    }  
    return null;  
}  
  
public String getEstat() { // Sessio 2?  
    if (any_col==0) return "retirat";  
    switch(any_col - (new GregorianCalendar().get(1)) ) {  
        case 0:  
        case 1: return "nou";  
        case 2:  
        case 3: return "semi nou";  
        case 4:  
        default: return "vell";  
    }  
}  
}
```

### Ubicacio.java

```
package Sessio1;  
  
public class Ubicacio {  
    private String nom_via;        // Atr. 1  
    private int num_via;           // Atr. 2  
    private boolean cruilla;       // Atr. 3  
  
    private SenyalTransit[] senyals;  
    private int numSenyals;  
    private final int MAX_SENYALS;  
  
    public Ubicacio(int Maxim, String carrer, int numero) {  
        MAX_SENYALS=Maxim;  
  
        nom_via=carrer;  
        num_via=numero;  
        cruilla=false;  
    }  
}
```

```
        senyals=new SenyalTransit[MAX_SENYALS];
        numSenyals=0;
    }

    public boolean afegirSenyal(SenyalTransit c) {
        if (numSenyals==MAX_SENYALS) return false;
        int cnt=0; boolean trobat=false;
        while (cnt<numSenyals) {
            trobat=c.getCodi().equals(senyals[cnt].getCodi()) ;
            if(trobat) return false;
            cnt++;
        }

        senyals[numSenyals]=c; numSenyals++;
        return true;
    }

    public boolean treureSenyal(SenyalTransit c) {
        for(int cnt=0;cnt<numSenyals; cnt++) {
            if(c.getCodi().equals(senyals[cnt].getCodi())) {
                senyals[cnt]=null;

                for (int i=cnt; (i>0 && i<numSenyals); i++) {
                    senyals[cnt]= (cnt+1==MAX_SENYALS) ? null :
senyals[cnt+1] ;
                }
            };
            cnt++;
        }
        return false;
    }

    //SETs
    public void setCruilla() { cruilla=!cruilla; }

    //GETs
    public String getNomVia() {return(nom_via);}
```

```
public int getNumVia() {return(num_via);}
public boolean getCruilla() {return(cruilla);}
public int getNumSenyals() {return(numSenyals);}
public int getMaxSenyals() {return(MAX_SENYALS);}
public SenyalTransit getSenyal(int quin) {
    if (quin>=0 && quin<MAX_SENYALS && quin<numSenyals) return
senyals[quin];
    return null;
}
public String getSenyals() {
    String msg="";
    for(SenyalTransit senyal : senyals) msg+=senyal.getCodi()+" - "; //
remove last " - "
    return msg;
}
}
```

Pol Rubio  
Rems Nalivaiko  
25 de feb. de 2022  
103112 - PROGRAMACIÓ ORIENTADA OBJECTE  
Fundació TecnoCampus Mataró-Maresme



## Sessió 2:

Codi:

### **SenyalTransit.java**

```
package Sessio2;
import java.util.GregorianCalendar;

public class SenyalTransit {

    public static final int Advertencia = 0;
    public static final int Reglamentacio = 1;
    public static final int Indicacio = 2;

    private String codi;
    private int tipus;
    private Ubicacio ubicacio;
    private int any_col=0;
    private int any_ret=0;

    public SenyalTransit(Ubicacio ubicacio) {
        this((new GregorianCalendar()).get(1), ubicacio);
    }

    public SenyalTransit(Ubicacio ubicacio, int tipus) {
        this(GenerarParametresSenyal.generarCodi(), tipus, ubicacio,
        GenerarParametresSenyal.generarAny((new GregorianCalendar()).get(1)));

        if(quinTipus(this.codi)!=tipus) {
            String[] c={"ROD","TRI", ( Math.random()>0.5 ? "REC" :
"QUA") };
            this.codi=c[tipus]+this.codi.substring(3);
        }
    }

    public SenyalTransit(int anyColocacio, Ubicacio ubicacio) {
        this(GenerarParametresSenyal.generarCodi(), 0, ubicacio,
anyColocacio);
        this.tipus=quinTipus(this.codi);
    }
}
```

```
    }

    public SenyalTransit(String codi, int tipus, Ubicacio ubicacio, int
anyColocacio) {
        this.codi = codi;
        this.tipus = tipus;

        if (ubicacio.afegirSenyal(this)) {
            this.ubicacio = ubicacio;
            any_col=anyColocacio;
        } else {
            ubicacio=null;
            any_ret=anyColocacio;
        }
    }

    private static int quinTipus(String codi) {
        switch(codi.substring(0,4)) {
            case "ROD": return Reglamentacio;
            case "TRI": return Advertencia;
            default: return Indicacio;
        }
    }

    public boolean retirarViaPublica() {
        if(ubicacio.treureSenyal(this)) {
            any_ret=(new GregorianCalendar()).get(1);
            ubicacio=null;

            return true;
        }
        return false;
    }

    public boolean retirarViaPublica(int anyRetirada) {
        if(ubicacio.treureSenyal(this)) {
            any_ret=anyRetirada;
        }
    }
}
```

```
        ubicacio=null;

        return true;
    }
    return false;
}

public boolean canviarUbicacio(Ubicacio novaUbicacio) {
    if(novaUbicacio.afegirSenyal(this)){
        ubicacio=novaUbicacio;
        return ubicacio.treureSenyal(this);
    }

    return false;
}

// GETs
public String getStrUbicacio() { // nom en el doc: getUbicacio()
    if(ubicacio != null)
        return ubicacio.getNomVia() + ", " + ubicacio.getNumVia();
    return null;
}

public Ubicacio getUbicacio() { // nom en el doc: getAtribut3()
    return ubicacio;
}

public String getTipusSenyal() {
    switch(tipus) {
        case Advertencia:    return "Advertència";
        case Reglamentacio: return "Reglamentació";
        case Indicacio:      return "Indicació";
    }
    return null;
}

public String getCodi() {return codi;}

public String getEstat() {
    GregorianCalendar avui = new GregorianCalendar();
    switch(any_col - avui.get(1)) {
```

```
        case 0:  
        case 1: return "nou";  
        case 2:  
        case 3: return "semi nou";  
        case 4:  
        default: return "vell";  
    }  
}  
}
```

### Ubicacio.java

```
package Sessio2;  
  
public class Ubicacio {  
    private String nom_via;        // Atr. 1  
    private int num_via;           // Atr. 2  
    private boolean cruilla;       // Atr. 3  
  
    private SenyalTransit[] senyals;  
    private int numSenyals;  
    private final int MAX_SENYALS;  
  
    public Ubicacio(int Maxim, String carrer, int numero) {  
        MAX_SENYALS=Maxim;  
  
        nom_via=carrer;  
        num_via=numero;  
        cruilla=false;  
  
        senyals=new SenyalTransit[MAX_SENYALS];  
        numSenyals=0;  
    }  
  
    public boolean afegirSenyal(SenyalTransit c) {  
        if (numSenyals==MAX_SENYALS) return false;  
        int cnt=0; boolean trobat=false;
```

```
        while (cnt<numSenyals) {
            trobat=c.getCodi().equals(senyals[cnt].getCodi()) ;
            if(trobat) return false;
            cnt++;
        }

        senyals[numSenyals]=c; numSenyals++;
        return true;
    }

    public boolean treureSenyal(SenyalTransit c) {
        for(int cnt=0;cnt<numSenyals; cnt++) {
            if(c.getCodi().equals(senyals[cnt].getCodi())) {
                senyals[cnt]=null;

                for (int i=cnt; (i>0 && i<numSenyals); i++) {
                    senyals[i]= (i==MAX_SENYALS) ? null :
senyals[i-1] ;
                }
            };
            cnt++;
        }
        return false;
    }

    //SETs
    public void setCruilla() { cruilla=!cruilla; }

    //GETs
    public String getNomVia() {return(nom_via);}
    public int getNumVia() {return(num_via);}
    public boolean getCruilla() {return(cruilla);}
    public int getNumSenyals() {return(numSenyals);}
    public int getMaxSenyals() {return(MAX_SENYALS);}
    public SenyalTransit getSenyal(int quin) {
        if (quin>=0 && quin<MAX_SENYALS && quin<numSenyals) return
senyals[quin];
        return null;
    }
}
```

```
    }  
    public String getSenyals() {  
        String msg="";  
        for(SenyalTransit senyal : senyals) msg+=senyal.getCodi()+" - "; //  
remove last " - "  
        return msg;  
    }  
}
```

### GenerarParametresSenyal.java

```
package Sessio2;  
  
public class GenerarParametresSenyal {  
  
    public static String generarCodi() {  
        String digit="";  
        for(int i=0; i<4; i++) digit+=generarDigit();  
  
        return( generarForma()+"-"+digit );  
    }  
  
    public static int generarAny(int maxim) {  
        // maxim inclusiu o exclusiu?  
        int minim=1980;  
        return( (int) (Math.random() * (maxim+1-minim))+minim ); // max. es  
inclusiu  
    }  
  
    public static char generarDigit() {  
        int minim=48; int maxim=57+1; // +1 per tal de 57 ser inclusiu.  
        int charNum=(int) (Math.random() * (maxim-minim))+minim ;  
        return( (char) charNum );  
    }  
  
    public static String generarForma() {  
        String[] conjunt={"ROD", "TRI", "REC", "QUA"};  
        return( conjunt[(int) (Math.random()*conjunt.length)] );  
    }  
}
```

```
}  
  
}
```

## Poblacio.java

```
package Sessio2;  
  
public class Poblacio {  
    private String nomPoblacio;  
    private int numHabitants;  
  
    private Ubicacio ubicacions[][]; // ? 0=cruilles , 1=no cuilles? ()  
    private int numParells;  
    private int numSenars;  
    private final int MAX_UBICACIONS;  
  
    public Poblacio(int Maxim, String poblacio, int numHabitants) {  
        nomPoblacio=poblacio;  
        this.numHabitants=numHabitants;  
        MAX_UBICACIONS=Maxim;  
  
        numParells=0;  
        numSenars=0;  
        ubicacions=new Ubicacio[2][MAX_UBICACIONS];  
    }  
  
    public boolean afegirUbicacio(Ubicacio c) {  
        int idx=c.getNumVia()%2==0 ? 0 : 1;  
        for (Ubicacio x: ubicacions[idx]) {  
            if(x.getNomVia().equals(c.getNomVia()) &&  
x.getNumVia()==c.getNumVia())  
                return false;  
        }  
  
        // *1  
        int llargada=(idx==1) ? numSenars : numParells;
```

```
        ubicacions[idx][llargada]=c;

        // *1
        if (idx==1) numSenars++;
        else numParells++;

        // *1 = trobar una solucio mes eficient.

        return true;
    }
    public boolean eliminarUbicacio(Ubicacio c) {
        int idx=c.getNumVia()%2==0 ? 0 : 1;
        int llargada=(idx==1) ? numSenars : numParells;

        int cnt=0; boolean trobat=false;
        while(cnt<llargada && !trobat) {
            trobat=ubicacions[idx][cnt].getNomVia().equals(c.getNomVia())
&& ubicacions[idx][cnt].getNumVia()==c.getNumVia();
            cnt++;
        }

        if(trobat) {
            ubicacions[idx][cnt-1]=null;

            // correcte?
            for (int i=cnt-1; i<llargada-1; i++)
                ubicacions[idx][i]=ubicacions[idx][i+1];

            if (idx==1) numSenars--;
            else numParells--;

            return true;
        }

        return false;
    }
}
```



```
public int eliminarBuides() {
    int cnt=0;
    int lengths[]= {numParells, numSenars};

    for(int x=0; x<lengths.length; x++) {
        for (int y=0; y<lengths[x]; y++) {
            if (ubicacions[x][y].getNumSenyals()==0) {
                if(eliminarUbicacio(ubicacions[x][y])) {
                    y--; cnt++;
                }
            }
        }
    }

    return cnt;
}

public int quantes() {
    int cnt=0;
    int lengths[]= {numParells, numSenars};
    for(int x=0; x<lengths.length; x++) {
        for (int y=0; y<lengths[x]; y++) {
            cnt+=ubicacions[x][y].getNumSenyals();
        }
    }

    return cnt;
}

// GETs
public String getNomPoblacio() {return nomPoblacio;}
public int getNumHabitants() {return numHabitants;}
public int getNumParells() {return numParells;}
public int getNumSenars() {return numSenars;}
public int getMaxUbicacions() {return MAX_UBICACIONS;}
}
```

## Sessió 3:

Codi:

### **Advetencia.java**

```
package Sessio3;

public class Advertencia extends SenyalTransit{

    private String significatSenyal;
    private int costat;

    public Advertencia(String codi, Ubicacio ubicacio, int
anyColocacio, String descripcio) {
        this(codi, ubicacio, anyColocacio, descripcio,
Costat);
    }
    public Advertencia(String codi, Ubicacio ubicacio, int
anyColocacio, String descripcio, int costat) {
        super(codi, Advertencia, ubicacio, anyColocacio);

        this.costat=costat;
        significatSenyal=descripcio;
        cntAdvertencia++;
    }

    public void visualitza() {
        super.visualitza();
        System.out.println("Descripcio: "+significatSenyal);
        System.out.println("Cada costat de: "+costat);
    }

    public float area() { return (float)
(Math.sqrt(3)/4*Math.pow(costat,2)); }
    public String getSignificat() { return significatSenyal; }
    public int getCostat() { return costat; }
}
```

### **GenerarParametresSenyal.java**

```
package Sessio3;

public class GenerarParametresSenyal {

    public static String generarCodi() {
        String digit="";
        for(int i=0; i<4; i++) digit+=generarDigit();
    }
}
```

```
        return( generarForma()+"-"+digit );
    }

    public static int generarAny(int maxim) {
        // maxim inclusiu o exclusiu?
        int minim=1980;
        return( (int) (Math.random() * (maxim-minim))+minim );
// max. es exclusiu
    }

    public static char generarDigit() {
        int minim=48; int maxim=57+1; // +1 per tal de 57 ser
inclusiu.
        int charNum=(int) (Math.random() *
(maxim-minim))+minim ;
        return( (char) charNum );
    }

    public static String generarForma() {
        String[] conjunt={"ROD","TRI","REC","QUA"};
        return( conjunt[(int) (Math.random()*conjunt.length)]
);
    }
}
```

### Indicacio.java

```
package Sessio3;

public class Indicacio extends SenyalTransit{

    private String significatSenyal;
    private int amplada;
    private int alcada;

    public Indicacio(String codi, Ubicacio ubicacio, int
anyColocacio, String descripcio) {
        this(codi, ubicacio, anyColocacio, descripcio,
Amplada, Alcada);
    }
    public Indicacio(String codi, Ubicacio ubicacio, int
anyColocacio, String descripcio, int amplada, int alcada) {
        super(codi, Indicacio, ubicacio, anyColocacio);

        this.amplada=amplada;
        this.alcada=alcada;
        significatSenyal=descripcio;
        cntIndicacio++;
    }
}
```

```
public void visualitza() {  
    super.visualitza();  
    System.out.println("Descripcio: "+significatSenyal);  
    System.out.println("Dimensions: "+amplada+"x"+alcada);  
}  
  
public float area() { return (amplada*alcada); }  
  
public String getSignificat() { return significatSenyal; }  
public int getAmplada() { return amplada; }  
public int getAlcada() { return alcada; }  
}
```

### **Poblacio.java**

```
package Sessio3;  
  
public class Poblacio {  
    private String nomPoblacio;  
    private int numHabitants;  
  
    private Ubicacio ubicacions[][]; // ? 0=cruilles , 1=no  
    cuilles? ()  
    private int numParells;  
    private int numSenars;  
    private final int MAX_UBICACIONS;  
  
    public Poblacio(int Maxim, String poblacio, int  
numHabitants) {  
        nomPoblacio=poblacio;  
        this.numHabitants=numHabitants;  
        MAX_UBICACIONS=Maxim;  
  
        numParells=0;  
        numSenars=0;  
        ubicacions=new Ubicacio[2][MAX_UBICACIONS];  
    }  
  
    public boolean afegirUbicacio(Ubicacio c) {  
        int idx=c.getNumVia()%2==0 ? 0 : 1;  
        for (Ubicacio x: ubicacions[idx]) {  
            if(x.getNomVia().equals(c.getNomVia()) &&  
x.getNumVia()==c.getNumVia())  
                return false;  
        }  
  
        // *1  
        int llargada=(idx==1) ? numSenars : numParells;
```

```
        ubicacions[idx][llargada]=c;

        // *1
        if (idx==1) numSenars++;
        else numParells++;

        // *1 = trobar una solucio mes eficient.

        return true;
    }
    public boolean eliminarUbicacio(Ubicacio c) {
        int idx=c.getNumVia()%2==0 ? 0 : 1;
        int llargada=(idx==1) ? numSenars : numParells;

        int cnt=0; boolean trobat=false;
        while(cnt<llargada && !trobat) {

trobat=ubicacions[idx][cnt].getNomVia().equals(c.getNomVia()) &&
ubicacions[idx][cnt].getNumVia()==c.getNumVia();
            cnt++;
        }

        if(trobat) {
            ubicacions[idx][cnt-1]=null;

            // correcte?
            for (int i=cnt-1; i<llargada-1; i++)
                ubicacions[idx][i]=ubicacions[idx][i+1];

            if (idx==1) numSenars--;
            else numParells--;

            return true;
        }

        return false;
    }
    public int eliminarBuides() {
        int cnt=0;
        int lengths[]= {numParells, numSenars};

        for(int x=0; x<lengths.length; x++) {
            for (int y=0; y<lengths[x]; y++) {
                if (ubicacions[x][y].getNumSenyals()==0) {
                    if(eliminarUbicacio(ubicacions[x][y]))
{
                        y--; cnt++;
                    }
                }
            }
        }
    }
```

```
        }

        return cnt;
    }

    public int quantes() {
        int cnt=0;
        int lengths[]= {numParells, numSenars};
        for(int x=0; x<lengths.length; x++) {
            for (int y=0; y<lengths[x]; y++) {
                cnt+=ubicacions[x][y].getNumSenyals();
            }
        }

        return cnt;
    }

    // GETs
    public String getNomPoblacio() {return nomPoblacio;}
    public int getNumHabitants() {return numHabitants;}
    public int getNumParells() {return numParells;}
    public int getNumSenars() {return numSenars;}
    public int getMaxUbicacions() {return MAX_UBICACIONS;}
}
```

### Prova.java

```
package Sessio3;

public class Prova {
    public static void main(String[] args) {
        int len_ubicacions=5;
        int len_senyals=30;
        int capacitat_ubicacio=5;

        SenyalTransit [] senyals = new
SenyalTransit[len_senyals];
        Ubicacio[] ubicacions=new Ubicacio[len_ubicacions];

        for(int i=0; i<len_ubicacions; i++) {
            ubicacions[i]=new Ubicacio(capacitat_ubicacio,
"Carrer "+i, 1);
            if ( (int)(Math.random()*4)==2 )
ubicacions[i].setCruilla(); // 1/3 de cada ubi sera cruilla
        }

        for(int i=0; i<senyals.length; i++) {
            int any=GenerarParametresSenyal.generarAny(2022);
            String
codi=GenerarParametresSenyal.generarCodi();
            Ubicacio ubi=ubicacions[i%len_ubicacions];
```

```
        String desc="descripcio de "+codi;
        switch(codi.substring(0,3)) {
            case "ROD": senyals[i]=new
Reglamentacio(codi, ubi, any, desc); break;
            case "TRI": senyals[i]=new
Advertencia(codi, ubi, any, desc); break;
            default:  senyals[i]=new Indicacio(codi,
ubi, any, desc); break;
        }
    }

    visualitzaSenyals(senyals);
    visualtizaAdvertencia(senyals);

}
private static void visualitzaSenyals(SenyalTransit []
senyals){
    for(SenyalTransit senyal : senyals) {
        senyal.visualitza();
        System.out.println("Area: "+senyal.area()+"
cm2");

        System.out.println();
    }
}

private static void visualtizaAdvertencia(SenyalTransit []
senyals) {
    for(SenyalTransit senyal : senyals) {
        if(senyal instanceof Advertencia) {
            senyal.visualitza();
            System.out.println("Area: "+senyal.area()+"
cm2");

            System.out.println();
        }
    }
}
}
```

### Reglametacio.java

```
package Sessio3;
public class Reglamentacio extends SenyalTransit{

    private String significatSenyal;
    private int diametre;

    public Reglamentacio(String codi, Ubicacio ubicacio, int
anyColocacio, String descripcio) {
        this(codi, ubicacio, anyColocacio, descripcio,
```

```
Diametre);  
    }  
    public Reglamentacio(String codi, Ubicacio ubicacio, int  
anyColocacio, String descripcio, int diametre) {  
        super(codi, Reglamentacio, ubicacio, anyColocacio);  
  
        this.diametre=diametre;  
        significatSenyal=descripcio;  
        cntReglamentacio++;  
    }  
  
    public void visualitza() {  
        super.visualitza();  
        System.out.println("Descripcio: "+significatSenyal);  
        System.out.println("Diametre: "+diametre);  
    }  
  
    public float area() { return (float)  
(Math.PI*Math.pow(diametre/2, 2)); }  
    public String getSignificat() { return significatSenyal; }  
    public int getDiametre() { return diametre; }  
}
```

### SenyalTransit.java

```
package Sessio3;  
import java.util.GregorianCalendar;  
  
public abstract class SenyalTransit {  
  
    public static final int Advertencia = 0;  
    public static final int Reglamentacio = 1;  
    public static final int Indicacio = 2;  
  
    public static final int Diametre=50;  
    public static final int Costat=70;  
    public static final int Alcada=132;  
    public static final int Amplada=93;  
  
    protected int cntAdvertencia=0;  
    protected int cntReglamentacio=0;  
    protected int cntIndicacio=0;  
  
    private String codi;  
    private int tipus;  
    private Ubicacio ubicacio;  
    private int any_col=0;  
    private int any_ret=0;  
  
    public SenyalTransit(Ubicacio ubicacio) {
```



```
        this((new GregorianCalendar()).get(1), ubicacio);
    }

    public SenyalTransit(int anyColocacio, Ubicacio ubicacio) {
        this(GenerarParametresSenyal.generarCodi(), 0,
ubicacio, anyColocacio);
        this.tipus=quinTipus(this.codi);
    }

    public SenyalTransit(Ubicacio ubicacio, int tipus) {
        this(GenerarParametresSenyal.generarCodi(), tipus,
ubicacio, GenerarParametresSenyal.generarAny((new
GregorianCalendar()).get(1)));

        if(quinTipus(this.codi)!=tipus) {
            String[] c={"ROD","TRI", ( Math.random()>0.5 ?
"REC" : "QUA")};
            this.codi=c[tipus]+this.codi.substring(3);
        }
    }

    public SenyalTransit(String codi, int tipus, Ubicacio
ubicacio, int anyColocacio) {
        this.codi = codi;
        this.tipus = tipus;

        if (ubicacio.afegirSenyal(this)) {
            this.ubicacio = ubicacio;
            any_col=anyColocacio;
        } else {
            ubicacio=null;
            any_ret=anyColocacio;
        }
    }

    private int quinTipus(String codi) {
        switch(codi.substring(0,3)) {
            case "ROD": return Reglamentacio;
            case "TRI": return Advertencia;
            default: return Indicacio;
        }
    }

    public boolean retirarViaPublica() {
        return(retirarViaPublica((new
GregorianCalendar()).get(1)));
    }
    public boolean retirarViaPublica(int anyRetirada) {
        if(ubicacio.treureSenyal(this)) {
            any_ret=anyRetirada;
            ubicacio=null;
        }
    }
}
```

```
        return true;
    }
    return false;
}
public boolean canviarUbicacio(Ubicacio novaUbicacio) {
    if(novaUbicacio.afegirSenyal(this)){
        ubicacio=novaUbicacio;
        return ubicacio.treureSenyal(this);
    }

    return false;
}
public void visualitza() {
    String msg=
        "Codi: "+codi +"\n" +
        "Tipus: "+getTipusSenyal()+"\n" +
        "Any colocacio: "+any_col+"\n" +
        "Any retirada: "+any_ret+"\n" +
        "Ubicacio: "+getStrUbicacio();

    ;

    System.out.println(msg);
}

public abstract float area();

// GETs
public String getStrUbicacio() { // nom en el doc:
getUbicacio()
    if(ubicacio != null)
        return ubicacio.getNomVia()+"",
        "+ubicacio.getNumVia();
    return null;
}
public Ubicacio getUbicacio() { // nom en el doc:
getAtribut3()
    return ubicacio;
}
public String getTipusSenyal() {
    switch(tipus) {
        case Advertencia:    return "Advertencia";
        case Reglamentacio: return "Reglamentació";
        case Indicacio:      return "Indicació";
    }
    return null;
}
public String getCodi() {return codi;}
public String getEstat() {
    GregorianCalendar avui = new GregorianCalendar();
    switch(any_col - avui.get(1)) {
```

```
        case 0:  
        case 1: return "nou";  
        case 2:  
        case 3: return "semi nou";  
        case 4:  
        default: return "vell";  
    }  
}  
public int getCntAdvertencia(){return cntAdvertencia;}  
public int getCntReglamentacio(){return cntReglamentacio;}  
public int getCntIndicacio(){return cntIndicacio;}  
}
```

### Ubicacio.java

```
package Sessio3;  
  
public class Ubicacio {  
    private String nom_via;    // Atr. 1  
    private int num_via;      // Atr. 2  
    private boolean cruilla;  // Atr. 3  
  
    private SenyalTransit[] senyals;  
    private int numSenyals;  
    private final int MAX_SENYALS;  
  
    public Ubicacio(int Maxim, String carrer, int numero) {  
        MAX_SENYALS=Maxim;  
  
        nom_via=carrer;  
        num_via=numero;  
        cruilla=false;  
  
        senyals=new SenyalTransit[MAX_SENYALS];  
        numSenyals=0;  
    }  
  
    public boolean afegirSenyal(SenyalTransit c) {  
        if (numSenyals==MAX_SENYALS) return false;  
        int cnt=0; boolean trobat=false;  
        while (cnt<numSenyals) {  
            trobat=c.getCodi().equals(senyals[cnt].getCodi());  
            if(trobat) return false;  
            cnt++;  
        }  
  
        senyals[numSenyals]=c; numSenyals++;  
        return true;  
    }  
};
```

```
    }  
    public boolean treureSenyal(SenyalTransit c) {  
        for(int cnt=0;cnt<numSenyals; cnt++) {  
            if(c.getCodi().equals(senyals[cnt].getCodi())) {  
                senyals[cnt]=null;  
  
                for (int i=cnt; (i>0 && i<numSenyals);  
i++) {  
                    senyals[cnt]= (cnt+1==MAX_SENYALS) ?  
null : senyals[cnt+1] ;  
                }  
            };  
            cnt++;  
        }  
        return false;  
    }  
  
    //SETs  
    public void setCruilla() { cruilla=!cruilla; }  
  
    //GETs  
    public String getNomVia() {return(nom_via);}   
    public int getNumVia() {return(num_via);}   
    public boolean getCruilla() {return(cruilla);}   
    public int getNumSenyals() {return(numSenyals);}   
    public int getMaxSenyals() {return(MAX_SENYALS);}   
    public SenyalTransit getSenyal(int quin) {  
        if (quin>=0 && quin<MAX_SENYALS && quin<numSenyals)  
return senyals[quin];  
        return null;  
    }  
    public String getSenyals() {  
        String codis[]=new String[senyals.length];  
        for(int i=0; i<senyals.length; i++) {  
            if (senyals[i]!=null)  
codis[i]=senyals[i].getCodi();  
        }  
  
        return String.join(" - ", codis);  
    }  
}
```

## Organització de la feina:

En aquesta pràctica n'hem dividit les diferents classes en cada sessió per tal d'anar fent-ho paral·lelament amb l'objectiu d'acabar més de pressa, una vegada acabàvem les nostres classes, les posàvem en comú per tal d'entendre que havia fet l'altre i detectar possibles errors o afegir possibles optimitzacions.

## Valoració:

Pel que fa a la valoració del funcionament dels exercicis desenvolupats en aquesta primera pràctica, val a dir que no hem tingut cap problema greu per a resoldre'ls.