

Programació Orientada a Objecte

Pràctica 2

Miquel Rodríguez Juvany

Eloi Egea

Grup 101

2021-22

Índex

Participació.....	2
Practica2Sessio1 SenyalTransit.java.....	3
Practica2Sessio1 Ubicacio.java.....	5
Practica2Sessio2 GenerarParametresSenyal.java.....	7
Practica2Sessio2 Poblacio.java.....	8
Practica2Sessio2 SenyalTransit.java.....	10
Practica2Sessio3 Advertencia.java.....	13
Practica2Sessio3 Indicacio.java.....	14
Practica2Sessio3 Reglamentacio.java.....	15
Practica2Sessio3 SenyalTransit.java.....	16

Participació

Per a realitzar aquesta pràctica des del primer dia ens vam posar a seguir les pautes del PDF Pràctica 2 Sessió 1. Durant la primera sessió ens vam trobar alguns problemes sobretot per l'ús del `this`, que no sabíem que es podia fer servir com a constructor, pensàvem que havíem de crear objectes de tipus `Ubicacio.java` i `SenyalTransit.java` per poder implementar correctament els mètodes. També vam tenir problemes a l'hora de guardar els senyals al vector perquè ens vam equivocar escrivint un `for`, agafant com a límit el màxim de senyals en comptes del número de senyals del vector, fent que el programa no realitzés correctament el que pretenia. La Lina ens va ajudar amb tots aquests problemes i ho vam entendre perfectament. A més, a la classe `Ubicacio.java` tenim un problema que no hem pogut solucionar en el que l'Eclipse ens mostra un warning al mètode `getSenyals()` que diu Dead code, a la línia 105 quan fem el `i++` dins del `for`. No hem sabut arreglar-ho.

A la segona sessió ens vam topar amb bastants més problemes. Per a fer la classe `GenerarParametresSenyal.java` vam discutir al respecte de com optimitzar al màxim el codi perquè un pretenia fer-ho d'una manera i l'altre d'una altra diferent i creiem que hem escollit les millors idees de cadascun. Amb la classe `Poblacio.java` vam trobar problemes amb les files de parells i senars, no sabíem com accedir a uns o altres depenent de a quina fila pertanyés. A la classe `SenyalTransit.java` vam trobar grans problemes, a l'hora de fer les sobrecàrregues del constructor sobretot. Intentàvem cridar el constructor però a l'hora de cridar-lo ens faltaven paràmetres i els definíem abans de cridar-lo. Això ens donava un error, i és que a l'hora de sobrecarregar el constructor, la crida al constructor previ a de ser a la primera línia. Per a solucionar-ho i de manera que acabem estalviant codi en altres mètodes també, hem optat per fer uns mètodes ajudants com ho són `generarCodi()`, `generarTipus()` i `anyActual()`. Aquests mètodes ens ajuden a passar paràmetres pel constructor i els podem utilitzar en altres mètodes (sobretot el de `anyActual` que es fa servir bastants vegades i fins aquell moment únicament creavem una variable al principi del programa i això ens donava errors més tard per problemes de static i non-static references). Amb el mètode `Ubicacio.java` no vam trobar problemes. A la classe `Ubicacio.java` seguim tenint el problema del Dead code.

A la tercera sessió hem trobat molts problemes. En els constructors no sabem a què es refereix el paràmetre descripció. No ens agafa bé l'atribut `super` per construir els atributs de la superclasse. Són errors grans que no hem pogut resoldre i ens impedeixen comprovar-ho al main.

Sessió 1

SenyalTransit.java

```
package Practica2Sessio1;

import java.util.GregorianCalendar;

public class SenyalTransit {

    public final int advertencia = 0;
    public final int reglamentacio = 1;
    public final int indicacio = 2;

    private String codiSenyal; // Atribut 1
    private int tipus; // Atribut 2
    private Ubicacio ubi; // Atribut 3
    private int anyColocacio; // Atribut 4
    private int anyRetirada; // Atribut 5

    GregorianCalendar avui = new GregorianCalendar();
    int anyActual = avui.get(1);

    // Constructor

    public SenyalTransit(String codi, int tipus, Ubicacio ubicacio, int anyColocacio) {
        this.codiSenyal = codi;
        this.tipus = tipus;
        if (!ubicacio.afegirSenyal(this)) {
            this.ubi = null;
            this.anyColocacio = 0;
            this.anyRetirada = anyActual;
        } else {
            this.ubi = ubicacio;
            this.anyColocacio = anyColocacio;
            this.anyRetirada = 0;
        }
    }

    public String getTipusSenyal() {
        String retorn;
        if (this.tipus == advertencia) {
            retorn = "Advertencia";
        } else if (this.tipus == reglamentacio) {
            retorn = "Reglamentacio";
        } else {
            retorn = "Indicacio";
        }

        return retorn;
    }

    public boolean retirarViaPublica() {
        boolean retorn = false;
        if (ubi.treureSenyal(this)) {
            int anyActual = avui.get(1);
            this.anyRetirada = anyActual;
            this.ubi = null;
            this.anyColocacio = 0;
            retorn = true;
        }
        return retorn;
    }

    public String getUbicacio() {
        // String retorn = "";
        String carrer = "";
        String numero = "";
        // Ubicacio u;
        if (this.ubi != null) {
            carrer = ubi.getVia();
            numero = Integer.toString(ubi.getNum());
            carrer = carrer + " " + numero;
        }
    }
}
```

```

        }
        return carrer;
    }

    boolean canviarUbicacio(Ubicacio novaUbicacio) {
        if (ubi.afegirSenyal(this) == false) {
            return false;
        } else {
            ubi.treureSenyal(this);
            ubi.afegirSenyal(this);
            return true;
        }
    }

    String getCodiSenyal() {
        return this.codiSenyal;
    }

    Ubicacio getUbi() {
        return this.ubi;
    }

    String getEstat() {
        int any;
        String estat;
        any = anyActual - anyColocacio;
        if (any > 4) {
            estat = "vell";
        } else if (any < 4 && any >= 2) {
            estat = "semi nou";
        } else {
            estat = "nou";
        }
        return estat;
    }
}

```

Ubicacio.java

```
package Practica2Sessio1;

public class Ubicacio {

    // Declaració atributs

    private String via;
    private int num;
    private boolean cruilla;

    private SenyalTransit[] senyals;
    private int numSenyals;
    private final int max_senyals;

    // Mètode constructor

    public Ubicacio(int maxim, String carrer, int numero) {
        this.via = carrer;
        this.num = numero;
        this.cruilla = false;

        this.senyals = new SenyalTransit[maxim];
        this.numSenyals = 0;
        this.max_senyals = maxim;
    }

    public String getVia() {
        return this.via;
    }

    public int getNum() {
        return this.num;
    }

    public boolean getCruilla() {
        return this.cruilla;
    }

    public int getNumSenyals() {
        return this.numSenyals;
    }

    public int getMaxSenyals() {
        return this.max_senyals;
    }

    public void setCruilla() {
        if (this.cruilla == false)
            this.cruilla = true;
        else
            this.cruilla = false;
    }

    public SenyalTransit getSenyal(int quin) {
        if (quin > 0 && quin <= this.max_senyals)
            return this.senyals[quin - 1];
        else
            return null;
    }

    public boolean afegirSenyal(SenyalTransit c) {
        boolean retorn = false;
        for (int i = 0; i < this.numSenyals; i++) {
            if (this.senyals[i].getCodiSenyal().equals(c.getCodiSenyal())) {
                return false;
            }
        }
        if (this.numSenyals < this.max_senyals) {
            this.senyals[this.numSenyals] = c;
            this.numSenyals++;
            retorn = true;
        }
    }
}
```

```

    }
    return retorn;
}

public boolean treureSenyal(SenyalTransit c) {
    boolean retorn = false;
    for (int i = 0; i < this.numSenyals; i++) {
        if (this.senyals[i].equals(c)) {
            this.senyals[i] = null;
            this.senyals[i] = this.senyals[this.max_senyals - 1];
            this.numSenyals--;
            retorn = true;
        } else
            retorn = false;
    }
    return retorn;
}

public String getSenyals() {
    String retorn = "";
    for (int i = 0; i < this.max_senyals; i++) {
        retorn = String.valueOf(this.senyals[i]);
        System.out.print('-');
        return retorn;
    }
    return retorn;
}
}

```

Sessió 2

GenerarParametresSenyal.java

```
package Practica2Sessio2;
import java.util.Random;

public class GenerarParametresSenyal{

    Random random;

    public static String generarCodi() {
        String codi = "";
        String forma = GenerarParametresSenyal.generarForma();
        int digit = GenerarParametresSenyal.generarDigit();
        // codi = forma + digit;
        codi = forma + "-" + digit;
        return codi;
    }

    public static int generarAny(int maxim) {
        Random random = new Random();
        int minim = 1980;
        int randomAny = random.nextInt((maxim - minim)+1)+minim;
        return randomAny;
    }

    public static char generarDigit() {
        int maxim = 57;
        int minim = 48;
        Random random = new Random();
        char randomDigit = (char)(random.nextInt((maxim-minim)+1)+minim);
        return randomDigit;
        // Codi que genera un char random entre a-z
        // Random r = new Random();
        // char c = (char)(r.nextInt(26) + 'a');
    }

    public static String generarForma() {
        Random rnd = new Random();
        int random = rnd.nextInt(3);
        String forma = "";
        switch (random) {
            case 0: forma = "ROD";
            case 1: forma = "TRI";
            case 2: forma = "REC";
            case 3: forma = "QUA";
        }
        return forma;
    }
}
```


Poblacio.java

```
package Practica2Sessio2;

public class Poblacio {

    private String poblacio;           // Atribut 1
    private int habitants;             // Atribut 2
    private Ubicacio[][] ubicacions;  // Atribut 3
    private int numParells;            // Atribut 4
    private int numSenars;             // Atribut 5
    private final int MAX_UBICACIONS; // Atribut 6

    // Constructor

    public Poblacio (int Maxim, String poblacio, int numHabitants) {
        this.ubicacions = new Ubicacio[2][Maxim];
        this.numParells = 0;
        this.numSenars = 0;
        this.poblacio = poblacio;
        this.habitants = numHabitants;
        this.MAX_UBICACIONS = Maxim;
    }

    // Metodes acces atributs

    public String getPoblacio () {return this.poblacio;}
    public int getHabitants () {return this.habitants;}
    public int getNumParells () {return this.numParells;}
    public int getNumSenars () {return this.numSenars;}
    public int getMaxUbicacions () {return this.MAX_UBICACIONS;}

    boolean afegirUbicacio (Ubicacio c) {
        boolean retorn = false;
        int correspondencia;
        if (c.getCruilla()) {
            correspondencia = 0;
        } else correspondencia = 1;

        for(int i=0; i<this.MAX_UBICACIONS; i++) {
            if (this.ubicacions[correspondencia][i] != null) {
                retorn = false;
            } else if (c.getVia()==null){
                retorn = false;
            } else if (c == this.ubicacions[correspondencia][i]) {
                retorn = false;
            } else {
                if (correspondencia == 0) {
                    this.ubicacions[correspondencia][this.numParells] = c;
                    this.numParells++;
                    retorn = true;
                } else {
                    this.ubicacions[correspondencia][this.numSenars] = c;
                    this.numSenars++;
                    retorn = true;
                }
            }
        }
        return retorn;
    }

    boolean eliminarUbicacio (Ubicacio c) {
        boolean retorn = false;
        int correspondencia;
        if (c.getCruilla()) {
            correspondencia = 0;
        } else correspondencia = 1;

        if (c.getVia() != null) {
            retorn = false;
        } else {
            if (correspondencia == 0) {
                this.ubicacions[correspondencia][this.numParells] = null;
            }
        }
    }
}
```

```

        this.numParells--;
        retorn = true;
    } else {
        this.ubicacions[correspondencia][this.numSenars] = null;
        this.numSenars--;
        retorn = true;
    }
}
return retorn;
}

int eliminarBuides() {
    int buides=0;
    for (int i=0; i<this.MAX_UBICACIONES; i++) {
        for (int j=0; j<this.MAX_UBICACIONES; j++) {
            if (this.ubicacions[i][j] == null) {
                buides++;
            }
        }
    }
    return buides;
}

int quantes() {
    int quantes=0;
    quantes = this.numParells + this.numSenars;
    return quantes;
}
}

```

SenyalTransit.java

```
package Practica2Sessio2;

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 codiSenyal; // Atribut 1
    private int tipus; // Atribut 2
    private Ubicacio ubi; // Atribut 3
    private int anyColocacio; // Atribut 4
    private int anyRetirada; // Atribut 5

    // SenyalTransit t;
    GregorianCalendar avui = new GregorianCalendar();
    int anyActual = avui.get(1);

    // Constructor

    public SenyalTransit(String codi, int tipus, Ubicacio ubicacio, int anyColocacio) {
        this.codiSenyal = codi;
        this.tipus = tipus;
        if (!ubicacio.afegirSenyal(this)) {
            this.ubi = null;
            this.anyColocacio = 0;
            this.anyRetirada = anyActual;
        } else {
            this.ubi = ubicacio;
            this.anyColocacio = anyColocacio;
            this.anyRetirada = 0;
        }
    }

    // Sobrecrrega 1 constructors Sessió 2

    public SenyalTransit(Ubicacio ubicacio) {
        this(generarCodi(generarTipus()), generarTipus(), ubicacio, getAny());
    }

    // Sobrecrrega 2 constructors Sessió 2

    public SenyalTransit(Ubicacio ubicacio, int tipus) {
        this(generarCodi(tipus), tipus, ubicacio,
GenerarParametresSenyal.generarAny(getAny()));
    }

    // Sobrecrrega 3 constructors Sessió 2

    public SenyalTransit(int anyColocacio, Ubicacio ubicacio) {
        this(generarCodi(generarTipus()), generarTipus(), ubicacio, anyColocacio);
    }

    public String getTipusSenyal() {
        String retorn;
        if (this.tipus == advertencia) {
            retorn = "Advertencia";
        } else if (this.tipus == reglamentacio) {
            retorn = "Reglamentacio";
        } else
            retorn = "Indicacio";

        return retorn;
    }

    public boolean retirarViaPublica() {
        boolean retorn = false;
        if (ubi.treureSenyal(this)) {
            int anyActual = avui.get(1);
            this.anyRetirada = anyActual;
        }
    }
}
```

```

        this.ubi = null;
        this.anyColocacio = 0;
        retorn = true;
    }
    return retorn;
}

// Problema trobat: S'ha de "crear" un objecte Ubicacio
// per poder treballar amb les funcions de la classe
// Ubicacio

public String getUbicacio() {
    // String retorn= "";
    String carrer = "";
    String numero = "";
    // Ubicacio u;
    if (this.ubi != null) {
        carrer = ubi.getVia();
        numero = Integer.toString(ubi.getNum());
        carrer = carrer + " " + numero;
    }
    return carrer;
}

boolean canviarUbicacio(Ubicacio novaUbicacio) {
    if (ubi.afegirSenyal(this) == false) {
        return false;
    } else {
        ubi.treureSenyal(this);
        ubi.afegirSenyal(this);
        return true;
    }
}

String getCodiSenyal() {
    return this.codiSenyal;
}

Ubicacio getUbi() {
    return this.ubi;
}

String getEstat() {
    int any;
    String estat;
    any = anyActual - anyColocacio;
    if (any > 4) {
        estat = "vell";
    } else if (any < 4 && any >= 2) {
        estat = "semi nou";
    } else {
        estat = "nou";
    }
    return estat;
}

// Sobrecarrega retirarViaPublica

public boolean retirarViaPublica(int anyRetirada) {
    boolean retorn = false;
    if (ubi.treureSenyal(this)) {
        this.anyRetirada = anyRetirada;
        this.ubi = null;
        this.anyColocacio = 0;
        retorn = true;
    }
    return retorn;
}

private static String generarCodi(int tipus) {
    String codi = "";
    switch (tipus) {
        case 1:

```

```

        codi = "ROD";
        break;
    case 2:
        codi = "TRI";
        break;
    case 3:
        codi = "QUA";
        break;
    }
    for (int i = 0; i < 4; i++) {
        codi.concat(GenerarParametresSenyal.generarDigit() + "");
    }
    return codi;
}

private static int generarTipus() {
    String t = "";

    for (int i = 0; i < 4; i++) {
        t.concat(GenerarParametresSenyal.generarCodi().substring(i));
    }
    switch (t) {
        case "ROD":
            return advertencia;
        case "TRI":
            return reglamentacio;
        default:
            return indicacio;
    }
}

private static int getAny() {
    GregorianCalendar avui = new GregorianCalendar();
    int anyActual = avui.get(1);
    return anyActual;
}
}

```

Sessió 3

Advertencia.java

```
package Practica2Sessio3;

import java.math.*;

public class Advertencia extends SenyalTransit {

    private String significatSenyal;

    private double costat;

    public String getSignificatSenyal() {
        return this.significatSenyal;
    }

    public double getCostat() {
        return this.costat;
    }

    public Advertencia(String codiSenyal, Ubicacio ubicacio, int anyColocacio, String
descripcio) {

        super(codiSenyal, ubicacio, anyColocacio, descripcio);
        this.significatSenyal = codiSenyal;

    }

    public Advertencia(double costat) {
        this(codiSenyal, getUbicacio(), anyColocacio, getDescripcio());
        this.costat=costat;
    }

    public float area() {

        this.costat = Math.PI * (CIRCULAR / 2) * (CIRCULAR / 2);
        return (float) this.costat;
    }

    public void visualitza() {
        System.out.println("L'Area ♦s de "+area()+"cm^2");
        System.out.println("La descripci♦ ♦s "+getDescripcio());
        super.visualitza();
    }

    public String getUbicacio() {
        String t = "";
        t = Ubicacio.getVia() + "-" + Ubicacio.getNum();
        return t;
    }

    public String getDescripcio() { //no sabem quina descripcio es refereix
        String t = "";
        return t;
    }

}
```

Indicacio.java

```
package Practica2Sessio3;

public class Indicacio extends SenyalTransit {

    private String significatSenyal;

    private int alturaRectangle;
    private int baseRectangle;

    public String getSignificatSenyal() {
        return this.significatSenyal;
    }

    public int getAlturaRectangle() {
        return this.alturaRectangle;
    }

    public int getBaseRectangle() {
        return this.baseRectangle;
    }

    public Indicacio(String codiSenyal, Ubicacio ubicacio, int anyColocacio, String
descripcio) {

        super(codiSenyal, ubicacio, anyColocacio, descripcio);
        this.significatSenyal = codiSenyal;
    }

    public Indicacio(int base, int altura) {
        this(codiSenyal, getUbicacio(), anyColocacio, getDescripcio());
        this.baseRectangle = base;
        this.alturaRectangle=altura;
    }

    public float area() {
        float x = 0;
        x = this.alturaRectangle * this.baseRectangle;
        return (float) x;
    }

    public void visualitza() {
        System.out.println("L'Area de "+area()+"cm^2");
        System.out.println("La descripci de "+getDescripcio());
        super.visualitza();
    }

    public String getUbicacio() {
        String t = "";
        t = Ubicacio.getVia() + "-" + Ubicacio.getNum();
        return t;
    }

    public String getDescripcio() { // no sabem quina descripcio es refereix
        String t = "";
        return t;
    }

}
```

Reglamentacio.java

```
package Practica2Sessio3;

public class Reglamentacio extends SenyalTransit {

    private String significatSenyal;

    private int diametre;

    public String getSignificatSenyal() {
        return this.significatSenyal;
    }

    public int getDiametre() {
        return this.diametre;
    }

    public Reglamentacio(String codiSenyal, Ubicacio ubicacio, int anyColocacio, String
descripcio) {

        super(codiSenyal, ubicacio, anyColocacio, descripcio);
        this.significatSenyal = codiSenyal;
        this.diametre = CIRCULAR;

    }

    public Reglamentacio(int diametre) {
        this(codiSenyal, getUbicacio(), anyColocacio, getDescripcio());
        this.diametre = diametre;
    }

    public float area() {
        float x = 0;
        x = (float) (Math.PI * (this.diametre / 2) * (this.diametre / 2));
        return (float) x;
    }

    public void visualitza() {
        System.out.println("L'Area de "+area()+"cm^2");
        System.out.println("La descripció "+getDescripcio());
        super.visualitza();
    }

    public String getUbicacio() {
        String t = "";
        t = Ubicacio.getVia() + "-" + Ubicacio.getNum();
        return t;
    }

    public String getDescripcio() { // no sabem quina descripcio es refereix
        String t = "";
        return t;
    }

}
```


SenyalTransit.java

```
package Practica2Sessio3;

import Practica2Sessio3.SenyalTransit;

public class Ubicacio {

    // Declaració atributs

    private String via;
    private int num;
    private boolean cruilla;

    private SenyalTransit[] senyals;
    private int numSenyals;
    private final int max_senyals;

    // Mètode constructor

    public Ubicacio(int maxim, String carrer, int numero) {
        this.via = carrer;
        this.num = numero;
        this.cruilla = false;

        this.senyals = new SenyalTransit[maxim];
        this.numSenyals = 0;
        this.max_senyals = maxim;
    }

    public String getVia() {
        return this.via;
    }

    public int getNum() {
        return this.num;
    }

    public boolean getCruilla() {
        return this.cruilla;
    }

    public int getNumSenyals() {
        return this.numSenyals;
    }

    public int getMaxSenyals() {
        return this.max_senyals;
    }

    public void setCruilla() {
        if (this.cruilla == false)
            this.cruilla = true;
        else
            this.cruilla = false;
    }

    public SenyalTransit getSenyal(int quin) {
        if (quin > 0 && quin <= this.max_senyals)
            return this.senyals[quin - 1];
        else
            return null;
    }

    public boolean afegirSenyal(SenyalTransit c) {
        boolean retorn = false;
        for (int i = 0; i < this.numSenyals; i++) {
            if (this.senyals[i].getCodiSenyal().equals(c.getCodiSenyal())) {
                return false;
            }
        }
        if (this.numSenyals < this.max_senyals) {
            this.senyals[this.numSenyals] = c;
        }
    }
}
```

```

        this.numSenyals++;
        return = true;
    }
    return return;
}

public boolean treureSenyal(SenyalTransit c) {
    boolean return = false;
    for (int i = 0; i < this.numSenyals; i++) {
        if (this.senyals[i].equals(c)) {
            this.senyals[i] = null;
            this.senyals[i] = this.senyals[this.max_senyals - 1];
            this.numSenyals--;
            return = true;
        } else
            return = false;
    }
    return return;
}

public String getSenyals() {
    String return = "";
    for (int i = 0; i < this.max_senyals; i++) {
        return = String.valueOf(this.senyals[i]);
        System.out.print('-');
        return return;
    }
    return return;
}
}
}

```