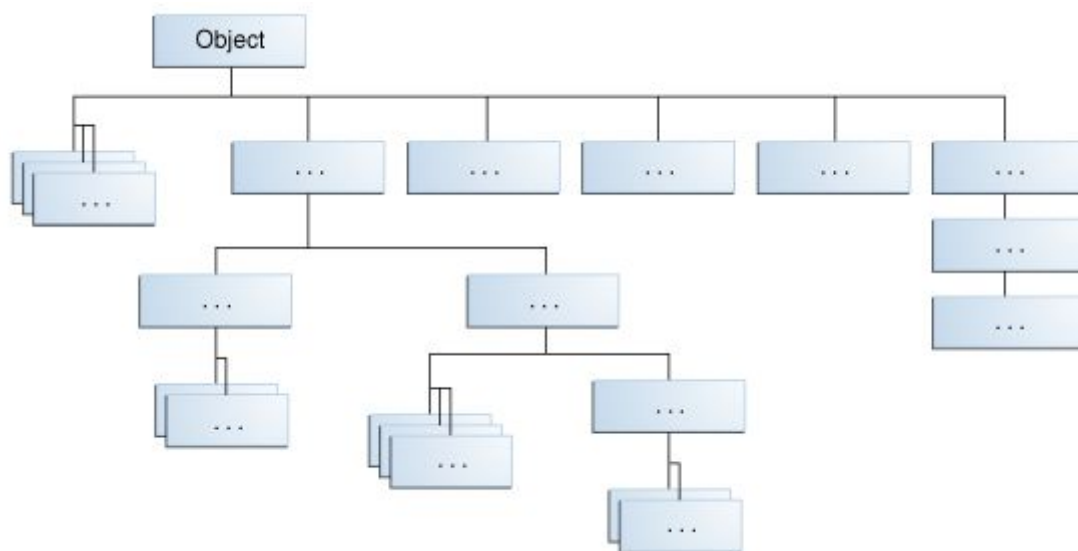


UF2 Disseny modular. Cas5



CFGS: 1r ASIX - DAW

Mòduls professionals: M03 Programació bàsica. Grup B

Grup: 4

Membres: Anwar El Khattabi - Marc España - Alberto Dos Santos

Professor/Professora assignat: Maria Merino Sanjuán



Generalitat de Catalunya
Departament d'Ensenyament
Institut Montsià

Mòdul 03 Cas 5





Professorat: Maria Merino Sanjuán

Alumnat: Anwar El Khattabi - Marc España - Alberto Dos Santos

Curs: ASIX_DAW_1B

Índex

Índex	2
Introducció	3
Programació Orientada a Objectes “POO”	4
Constructors	6
This	7
Getters I Setters	9
Paquet	10
Llistat de funcions i variables	11
Client	11
Seient	12
Obra	13
Teatre	14
Conclusions	33
JavaDocs	34
Webgrafia	35

 Generalitat de Catalunya Departament d'Ensenyament Institut Montsià	Mòdul 03 Cas 5	
Professorat: Maria Merino Sanjuán Alumnat: Anwar El Khattabi - Marc España - Alberto Dos Santos		Curs: ASIX_DAW_1B

Introducció



En aquest treball començarem primer amb una part més teòrica però amb alguns exemples que nosaltres realitzarem. Anirem veient i treballant l'entorn a la programació orientada a objectes, treballarem sobretot amb classes que crearem nosaltres mateixos, on veurem aspectes que treballarem com poden ser:

Constructors, Variables d'Instància, Paràmetres de Constructors, Objectes...
etc.

També veurem per a què serveix i com s'utilitza la paraula clau This, que té relació a com podem cridar mètodes dintre d'altres mètodes, per exemple.

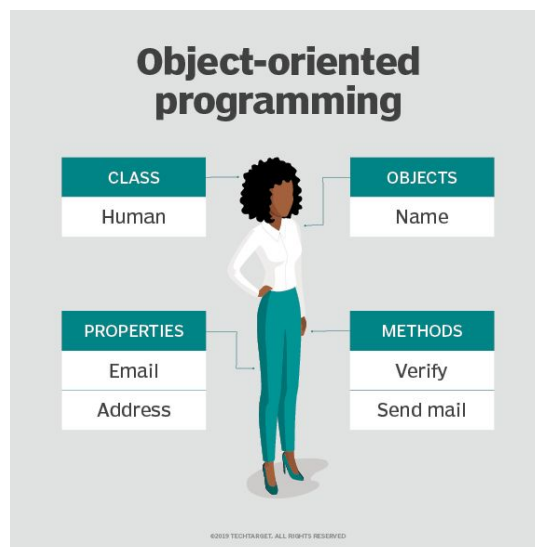
Per primer cop crearem el nostre paquet per a poder posteriorment utilitzar-lo.

Finalment acabarem amb quines són les nostres conclusions després d'haver treballat amb la programació orientada a objectes.

 Generalitat de Catalunya Departament d'Ensenyament Institut Montsià	Mòdul 03 Cas 5	
Professorat: Maria Merino Sanjuán Alumnat: Anwar El Khattabi - Marc España - Alberto Dos Santos		Curs: ASIX_DAW_1B

Programació Orientada a Objectes “POO”



La programació orientada a objectes és un enfocament de programació que combina dades i mètodes en objectes. Aquest enfocament augmenta la capacitat per administrar la complexitat del programari, la qual cosa resulta especialment important quan es desenvolupen i mantenen aplicacions i estructures de dades de grans dimensions.



La POO és una forma especial de programar, aquest paradigma de programació és proper a com expressem les coses a la vida real en el nostre dia a dia. Amb la POO hem d'aprendre a pensar les coses d'una manera diferent per poder escriure els nostres programes en termes d'objectes, propietats i mètodes.

Alguns dels **avantatges** de la programació orientada a objectes són els següents:

1. Els components es poden reutilitzar.
2. Facilitat de manteniment i modificació dels objectes existents.
3. Una estructura modular clara es pot obtenir, la qual no revela el mecanisme darrere de el disseny.

 Generalitat de Catalunya Departament d'Ensenyament Institut Montsià	Mòdul 03 Cas 5	
Professorat: Maria Merino Sanjuán Alumnat: Anwar El Khattabi - Marc España - Alberto Dos Santos		Curs: ASIX_DAW_1B

Desavantatges

1. Limitacions del programador: És possible el programador desconegui algunes característiques del paradigma i de fet segueix utilitzant el paradigma estructurat.
2. No hi ha una forma única de resoldre els problemes. Això pot portar a que diferents interpretacions de la solució plantejada emergeixin.
3. Es requereix una documentació àmplia per determinar la solució plantejada.



Constructors

Un **constructor** és un mètode part d'un objecte que serveix per **inicialitzar** diferents **variables** coherents i necessàries per a un **objecte**.



Els constructors tenen una sèrie de requisits:

1. S'anomena igual que la classe que el conté.
2. No retorna ningú valor, ni void.
3. Es poden crear diferents constructors a la mateixa classe, però només s'executara un d'ells. És solen crear diferents constructors amb diferents arguments d'entrada, per a que depenent de la quantitat d'arguments al iniciar un objecte executi un constructor o un altre.
4. Normalment és sol demanar arguments al constructor per a inicialitzar diferents variables de l'objecte.

```
public class Pilota{  
    int diametre;  
    String color;  
  
    public Pilota(int diametre, String color){  
        this.diametre = diametre;  
        this.color = color;  
    }  
    public Pilota(int diametre){  
        this.diametre = diametre;  
        this.color = "Blanc";//Color per defecte  
    }  
}
```

Exemple d'un constructor:

En el exemple superior podem veure que tenim la classe pilota i conté dos constructors. Cada constructor conté diferents paràmetres i depenent de la quantitat d'arguments inserits al iniciar l'objecte s'executara un constructor o l'altre.

 <p>Generalitat de Catalunya Departament d'Ensenyament Institut Montsià</p>	<p>Mòdul 03 Cas 5</p>	
<p>Professorat: Maria Merino Sanjuán Alumnat: Anwar El Khattabi - Marc España - Alberto Dos Santos</p>		<p>Curs: ASIX_DAW_1B</p>

This

La paraula clau **This** és una variable referent dintre del llenguatge Java, que fa referència a un objecte en concret.

Aquesta paraula clau, té diferents usos, com poden ser:

- Es pot utilitzar per a fer una referència a una instància o variable de la classe actual.
- Es pot fer passar com a un argument al cridar un mètode.
- Es pot fer passar com a un argument al cridar un constructor.
- Es pot utilitzar per a retornar la actual instància de la classe.

*S'ha de destacar que la paraula clau **This**, funciona igual dintre d'un constructor.

A continuació veurem un exemple general:

```
public class Persona {

    String nom;
    String cognom;
    String direccio;

    public Persona() {

        this.nom = "No assignat";
        this.cognom = "No assignat";
        this.direccio = "No assignat";
    }
}
```

Primer hem creat les 3 variables que utilitzarem i els hi assignem un valor "per defecte" donat pel **constructor buit**.





```
public Persona (String nom) {  
    this();  
    this.nom = nom;  
}  
  
public Persona(String nom, String cognom, String direccio) {  
    super();  
    this.nom = nom;  
    this.cognom = cognom;  
    this.direccio = direccio;  
}  
  
public String toString() {  
    return "Persona = " + this.nom + " " + cognom + " - Dir: " + direccio;  
}  
  
public static void main(String[] args) {  
    Persona p1 = new Persona();  
    Persona p2 = new Persona("Anwar");  
    Persona p3 = new Persona("Alberto", "Dos Santos", "Gran Via 14");  
    System.out.println(p1.toString());  
    System.out.println(p2.toString());  
    System.out.println(p3.toString());  
}
```

+Si ens fixem, al mètode **toString** utilitzem la clau **This** per accedir al camp **nom**. En realitat això no és realment necessari com es pot veure a la mateixa línia quan accedim al camp **nom**. Per defecte el compilador entén que ens referim al paràmetre i no al camp.

+Un altre cas d'exemple és el relacionat amb el **constructor** **Persona(String nom)**. Aquí es crida al constructor informant només el nom. Volem que els demés camps tinguin el valor per defecte donat pel constructor buit "**No assignat**".

+Al **constructor** **Persona(String nom, String cognom...)**, existeix un paràmetre anomenat **nom**. Si escrivim només **nom**, el compilador entén que ens referim al paràmetre i no al camp, per això utilitzem this.nom = nom.

Al mètode **main** podem veure com hem creat 3 objectes de persona, utilitzant els diferents constructors i després imprimim cada un dels objectes, utilitzant el mètode **toString()**.

 Generalitat de Catalunya Departament d'Ensenyament Institut Montsià	Mòdul 03 Cas 5	
Professorat: Maria Merino Sanjuán Alumnat: Anwar El Khattabi - Marc España - Alberto Dos Santos		Curs: ASIX_DAW_1B

Getters i Setters



Abans d'entendre que són els **Getters** i **Setters**, és necessari saber que és l'encapsulament, hi ha un tipus (private) que s'introdueix davant un atribut (en aquest cas) per a que no es pugui accedir des de una altra classe i sigui necessari utilitzar un mètode accesor als atributs amb els els Getters i Setters. Per exemple: private int edat;

Setters: Fan referència a la paraula Set, (establir, determinar, ...) de forma que, la seva funció és establir un nou valor a un atribut d'una classe. Crearem un nou mètode anomenat (Set+atribut) per a l'atribut en concret i amb un argument que indiqui la nova assignació. Per exemple:

```
public void SetEdat (int edat) {
    this.edat = edat;
}
```

Getters: Fan referència a la paraula Get, (obtenir, mostrar, ...) de forma que, crearem una funció amb el nom (Get+atribut) per al mateix atribut del Setter i a partir del getters cridar-lo des d'una altra classe. Per exemple:

```
public getEdat() {
    return edat;
}
```

 Generalitat de Catalunya Departament d'Ensenyament Institut Montsià	Mòdul 03 Cas 5	
Professorat: Maria Merino Sanjuán Alumnat: Anwar El Khattabi - Marc España - Alberto Dos Santos		Curs: ASIX_DAW_1B

Paquet

En la programació, normalment és útil agrupar peces relacionades d'un programa. En Java, això es pot fer utilitzant **paquets**.

Podríem dir que un paquet té dos propòsits:



- En primer lloc, proporciona un mecanisme el qual les parts relacionades d'un programa es poden organitzar com una unitat. S'accedeix a les classes definides a un paquet a través del seu propi nom.
- En segon lloc, un paquet participa en el mecanisme de control d'accés de Java. Les classes definides dintre d'un paquet es poden fer privades (**private**) per a aquell mateix paquet, per tant, no es podrà accedir per codi fora del paquet. Per tant, el paquet proporciona un mitjà pel qual les classes poden ser encapsulades.

La forma per a declarar un paquet seria la següent:

```
1 package elmeupaquet;
```

package + nom que li volem posar

*És important saber que els noms dels paquets són "case sensitive", distingeixen entre majúscules i minúscules, per tant, el directori en el qual s'emmagatzemen els arxius **.class** ha de tindre exactament el mateix nom que el paquet.

 Generalitat de Catalunya Departament d'Ensenyament Institut Montsià	Mòdul 03 Cas 5	
Professorat: Maria Merino Sanjuán Alumnat: Anwar El Khattabi - Marc España - Alberto Dos Santos		Curs: ASIX_DAW_1B

Llistat de funcions i variables

Client

Camps:

1. String nom
2. int edat
3. int diners

Constructors:



1. Client(String nom, int edat, int diners)
2. Client(int edat, int diners)
3. Client(int diners)

Funcions:

1. pagarEntrada
2. teDiners
3. majorEdat
4. toString

Getters i Setters:

1. getNom
2. getEdat
3. getDiners
4. setNom
5. setEdat
6. setDiners

 Generalitat de Catalunya Departament d'Ensenyament Institut Montsià	Mòdul 03 Cas 5	
Professorat: Maria Merino Sanjuán Alumnat: Anwar El Khattabi - Marc España - Alberto Dos Santos		Curs: ASIX_DAW_1B

Seient

Camps:

1. Int columna
2. Int fila
3. Persona persona

Constructors:



1. Seient(int columnaSeient, int filaSeient, Persona personaSeient)
2. Seient(int columnaSeient, int filaSeient)

Funcions:

1. ToString

Getters i Setters:

1. getColumna
2. setColumna
3. getFila
4. setFila
5. getPersona
6. setPersona

 Generalitat de Catalunya Departament d'Ensenyament Institut Montsià	Mòdul 03 Cas 5	
Professorat: Maria Merino Sanjuán Alumnat: Anwar El Khattabi - Marc España - Alberto Dos Santos		Curs: ASIX_DAW_1B

Obra

Camps:

1. String titol
2. Int durada
3. String autor
4. Int Edat

Constructors:



1. Obra (String titol, int durada, String autor, int edat)

Funcions:

1. MajorEdat
2. toString

Getters i Setters

1. setTitol
2. setDurada
3. setAutor
4. setEdat

 Generalitat de Catalunya Departament d'Ensenyament Institut Montsià	Mòdul 03 Cas 5	
Professorat: Maria Merino Sanjuán Alumnat: Anwar El Khattabi - Marc España - Alberto Dos Santos		Curs: ASIX_DAW_1B

Teatre

Camps:

1. Seient sessio [] []
2. double preu
3. Obra obra

Constructor:

1. Teatre (int files, int columnes, double preu, Obra obra)

Funcions:

1. seientBuit
2. potEntrar
3. seure

Getters i Setters:

1. setSessio
2. getSessio
3. setpreu
4. getPreu
5. setObra
6. getObra





Seient.java

Comencem creant la classe Seient que serà el bloc que engloba tot el document i a continuació els atributs o variables de instància (columna, fila, client). En els atributs columna i fila és evident perquè utilitzem un enter, però en el cas de client, utilitzem la classe Client perquè al ser un objecte podrem guardar més de una dada.

```
/* LA CLASSE Y ELS ATRIBUTS -----  
  
public class Seient {  
    private int columna;  
    private int fila;  
    private Client client;
```

A continuació crearem el número de constructors que sigui necessari, en aquest cas, simplement hauréu de fer dos. Un que pugui utilitzar tots els atributs com arguments i l'altre per a indicar només la posició del seient.

```
/* CONSTRUCTORS -----  
  
    public Seient(int columnaSeient, int filaSeient, Client clientSeient) {  
        columna = columnaSeient;  
        fila = filaSeient;  
        client = clientSeient;  
    }  
  
    public Seient(int columnaSeient, int filaSeient) {  
        columna = columnaSeient;  
        fila = filaSeient;  
    }  
}
```

 Generalitat de Catalunya Departament d'Ensenyament Institut Montsià	Mòdul 03 Cas 5	
Professorat: Maria Merino Sanjuán Alumnat: Anwar El Khattabi - Marc España - Alberto Dos Santos		Curs: ASIX_DAW_1B

Seguidament haurem d'establir en un Setter a cada atribut, indicant un argument que serà necessari per a indicar-li a quin atribut fa referència, i crear un Getter que utilitzarem en un altra classe per a fer la crida del atribut privat.

```

/* GETTERS Y SETTERS DE CADA ATRIBUT -----
public int getColumna() {
    return columna;
}

public void setColumna(int columnaS) {
    columna = columnaS;
}



public int getFila() {
    return fila;
}

public void setFila(int filaS) {
    fila = filaS;
}

public Client getPersona() {
    return client;
}

public void setPersona(Client personaS) {
    client = personaS;
}

```


 Generalitat de Catalunya Departament d'Ensenyament Institut Montsià	Mòdul 03 Cas 5	
Professorat: Maria Merino Sanjuán Alumnat: Anwar El Khattabi - Marc España - Alberto Dos Santos		Curs: ASIX_DAW_1B

Crearem un mètode toString que servirà per a imprimir les dades que s'hauran escrit de l'usuari (el nom i el lloc on es posicionarà).

```

/* METODE TOSTRING -----
    public String toString() {
        return ("Seient: Columna:" + columna + " Fila:" + fila + " Client:" + client.nom);
    }
}

```



Generalitat de Catalunya
Departament d'Ensenyament
Institut Montsià

Mòdul 03 Cas 5



Professorat: Maria Merino Sanjuán



Alumnat: Anwar El Khattabi - Marc España - Alberto Dos Santos

Curs: ASIX_DAW_1B

Client.java

Començarem creant la classe Client que engloba tot el document, amb els atributs o variables d'instància. Per a donar-li sentit als getters y setters seria necessari posar-los en private.

```
/* LA CLASSE I ELS ATRIBUTS -----  
public class Client {  
    private String nom;  
    private int edat;  
    private double diners;
```

 Generalitat de Catalunya Departament d'Ensenyament Institut Montsià	Mòdul 03 Cas 5	
Professorat: Maria Merino Sanjuán Alumnat: Anwar El Khattabi - Marc España - Alberto Dos Santos		Curs: ASIX_DAW_1B

A continuació crearem els constructors necessaris per a la classe client, que en aquest cas, només en necessitem tres, un per a utilitzar tots els atributs i així passar-lo per paràmetre i els altres dos en cas que necessitem especificar un o dos arguments.

```

/* CONSTRUCTORS -----
public Client(String nomClient, int edatClient, int dinersClient) {
    nom = nomClient;
    edat = edatClient;
    diners = dinersClient;
}

public Client(int edatClient, int dinersClient) {
    nom = "anonim";
    edat = edatClient;
    diners = dinersClient;
}



public Client(int dinersClient) {
    nom = "anonim";
    edat = 99;
    diners = dinersClient;
}

```



Com hem fet en totes les nostres classes, crearem un Setter y un Getter per a cada atribut.

```
/* GETTERS Y SETTERS DE CADA ATRIBUT -----  
  
    public String getNom() {  
        return nom;  
    }  
  
    public void setNom(String nomC) {  
        nom = nomC;  
    }  
  
    public int getEdat() {  
        return edat;  
    }  
  
    public void setEdat(int edatC) {  
        edat = edatC;  
    }  
  
    public double getDiners() {  
        return diners;  
    }  
  
    public void setDiners(int dinersC) {  
        diners = dinersC;  
    }  
}
```



 Generalitat de Catalunya Departament d'Ensenyament Institut Montsià	Mòdul 03 Cas 5	
Professorat: Maria Merino Sanjuán Alumnat: Anwar El Khattabi - Marc España - Alberto Dos Santos		Curs: ASIX_DAW_1B

En aquesta classe serà necessari especificar unes funcions concretes, la primera servirà per a restar el valor del preu de la entrada amb els diners (atribut) que disposa el client. Amb un print mostrarem per pantalla el resultat.

```
/* FUNCIONS -----
//Indicarà el valor per a restar-li al usuari
public boolean pagarEntrada(double preuEntrada) {
    diners = diners - preuEntrada;
    System.out.println("Entrada pagada \nLi queden " + diners + " euros");
    return (false);
}
```

La següent funció servirà per a indicar si l'usuari introduït (client) és major d'edat o no, mitjançant una condició. La funció ha de tornar un booleà, (major) aquest comença iniciat amb (false), si es un número menor que 18 el booleà es mantindrà amb false i tornarà el resultat de la variable. En cas de que tingue 18 o més, el booleà canvia d'estat i llavors tornaria la variable amb true.

```
public boolean majorEdat() {
    boolean major = false;
    if (edat < 18) {
        major = false;
        return (major);
    } else {
        major = true;
        return (major);
    }
}
```

 <p>Generalitat de Catalunya Departament d'Ensenyament Institut Montsià</p>	<p>Mòdul 03 Cas 5</p>	
<p>Professorat: Maria Merino Sanjuán Alumnat: Anwar El Khattabi - Marc España - Alberto Dos Santos</p>		<p>Curs: ASIX_DAW_1B</p>

Aquí farem una funció molt pareguda a l'anterior però que servirà per a saber si l'usuari ha introduït més diners del que val la entrada.

Comencem declarant un booleà (diners) i la iniciem amb false. A la condició indiquem que si l'atribut diners té un valor inferior al preu de la entrada (Preu) la variable es mantindrà i la tornarà amb false, en cas contrari canviarà el valor de la variable i la tornarà amb true.

```
public boolean teDiners(Teatre tr) {
    boolean dineros = false;
    if (diners < tr.getPreu()) {
        dineros = false;
        return (dineros);
    } else {
        dineros = true;
        return (dineros);
    }
}
```



Obra.java

Primer comencem declarant la classe i a continuació els seus atributs o també coneguts com variables d'instància, que són els següents:

Títol, Durada, Autor i MajorEdat.

Ho fem de la següent manera:

```
class Obra {  
  
    /*Atributs*/  
    private String titol; //Títol de l'obra  
  
    private int durada; //Durada de l'obra en minuts  
  
    private String autor; //Autor de l'obra  
  
    private int edat; //Major d'edat
```

Hem creat els atributs de forma privada per a poder obtenir un major control sobre la classe.

*Cal recordar que si els atributs són **private**, només podrem accedir a ells des de la mateixa classe.

A continuació hem creat un parell de constructors, un que serà el que doni els valors per defecte i un altre amb quatre paràmetres que són dels atributs que hem declarat anteriorment:

```
/*Constructors*/  
/*Constructor per defecte*/  
public Obra() {  
    this("", 1, "", 18);  
}
```



```
/*Constructor amb 4 paràmetres*/  
public Obra(String titol, int durada, String autor, int edat) {  
    this.titol = titol;  
    this.durada = durada;  
    this.autor = autor;  
    this.edat = edat;  
}
```

Ens podem fixar que hem utilitzat la paraula clau **this** per a fer referència a unes instàncies en concret de la classe, que en aquest cas són les indicades.

*Això ens serveix per a poder especificar els atributs que volem utilitzar.

Després d'haver acabat de crear el constructor necessari, hem començat a treballar amb els **mètodes**:

```
/*Mètodes públics*/  
/*Modifica el nom del titol*/  
public void setTitol(String titol) {  
    this.titol = titol;  
}  
  
/*Modifica la durada de l'obra*/  
public void setDurada(int durada) {  
    this.durada = durada;  
}  
  
/*Modifica el nom de l'autor*/  
public void setAutor(String autor) {  
    this.autor = autor;  
}  
  
/*Modifica l'edat*/  
public void setEdat(int edatClient) {  
    this.edat = edat;  
}
```




A continuació hem creat una funció per a saber si l'obra que es representarà, serà per a majors d'edat o no. Ho hem fet de la següent manera:

```
/*Indica si l'obra és per a majors d'edat*/  
public boolean MajorEdat() {  
    boolean major = false;  
    if (edat >= 18) {  
        major = true;  
    }  
    return major;  
}
```

Finalment hem **retornat** la informació dels objectes que hem creat, gràcies al **toString**. Ho hem fet de la següent manera:

```
/*Retorna la informació de l'objecte*/  
public String toString() {  
    return "Informació de la Obra:\n"  
        + "Títol: " + titol + "\n"  
        + "Durada: " + durada + " hores\n"  
        + "Autor: " + autor + "\n"  
        + "Edat de l'espectador: \n"  
        + "Edat: " + edat + " anys\n";  
}  
}
```



Teatre.java

Primer començem declarant la **classe** i seguidament les variables d'instància que la compondran. Ho fem de la següent manera:



```
class Teatre {  
  
    /*Variables d'instància*/  
    Obra obra; //Nom de l'obra que es representa  
  
    double preu; //Preu que costa l'obra representada  
  
    private double diners; //Diners que tindrà el client  
  
    private Seient[][] sessio; //Array 2d on emmagatzemarem les files i les columnes  
}
```

*Ens podem donar compte de que tenim una variable d'instància que és un array bidimensional, això significa que és **2d** (dos dimensions) i que per tant, estem parlant d'un array que tindrà **files** i **columnes**. També cal destacar que l'array és de tipus **Seient**, que és el nom que li hem posat a una de les classes que hem creat anteriorment.

Passant a la part del **constructor**, tenim el següent:

```
/*Constructors*/  
/*Constructor amb 5 paràmetres*/  
public Teatre(Obra obraT, double preuT, double dinersC, int files, int columnes) {  
    obra = obraT;  
    preu = preuT;  
    diners = dinersC;  
    sessio = new Seient[files][columnes];  
}
```

*Ens podem fixar que per a l'array bidimensional, utilitzem *dos paràmetres*, un per a cada dimensió.

 Generalitat de Catalunya Departament d'Ensenyament Institut Montsià	Mòdul 03 Cas 5	
Professorat: Maria Merino Sanjuán Alumnat: Anwar El Khattabi - Marc España - Alberto Dos Santos	Curs: ASIX_DAW_1B	

A continuació hem passat als Getters i Setters d'aquesta classe, que són els següents:

```

/*Mètodes públics*/

/*Accedeix al nom de l'obra*/
public Obra getObra() {
    return obra;
}

/*Modifica el nom de l'obra*/
public void setObra(Obra obraTeatre) {
    obra = obraTeatre;
}

/*Accedeix al preu de l'obra*/
public double getPreu() {
    return preu;
}

/*Modifica el preu de l'obra*/
public void setPreu(double preuTeatre) {
    preu = preuTeatre;
}

```

Podem veure els **getters** i **setters** de les dos primeres variables d'instància: **Obra** i **Preu**.



Getters i setters de la variable **diners**, que farem servir per a una de les nostres funcions:

```
/*Accedeix als diners del client*/  
public double getDiners() {  
    return diners;  
}  
  
/*Modifica els diners del client*/  
public void setDiners(double DinersClient) {  
    diners = DinersClient;  
}
```

Posteriorment hem creat el **setter** i **getter** del nostre array bidimensional. Ho hem fet de la següent manera:

```
/*Accedeix a les files i columnes*/  
public Seient getElement_sessio(int f, int c) {  
    return sessio[f][c];  
}  
  
/*Modifica les files i columnes*/  
public void SetSessio(Seient[][] sessio) {  
    this.sessio = sessio;  
}
```



Finalment hem concluit amb les funcions d'aquesta classe, que són les següents:

```
/*Funcions*/  
/*Assigna un seient*/  
public void setElement_sessio(Seient cadira){  
    this.sessio[cadira.files][cadira.columnes] = cadira;  
}
```

```
/*Té diners el client?*/  
public boolean preuDiners() {  
    boolean teDiners = true;  
    if (diners < preu) {  
        teDiners = false;  
    }  
    return teDiners;  
}
```




Main.java

En aquest arxiu/classe utilitzarem les classes prèviament creades per a fer un cas real d'un Teatre amb dades que introduirà l'usuari.

Per fer aquests inputs utilitzarem l'eina Scanner de Java.Util.

L'arxiu consta únicament d'un mètode "main".

```
Scanner sc = new Scanner(System.in);
```

Avans de començar crearem un objecte de tipus Scanner per a utilitzar els inputs.

Seguidament li demanem al usuari les dades dels objectes Obra i Teatre.

```
System.out.println("Introduïex el preu de la entrada (decimal en coma)");

double preuEntrada = sc.nextDouble();
sc.nextLine();
System.out.println("Numero de files al teatre (int)");

int filesTeatre = sc.nextInt();
sc.nextLine();
System.out.println("Numero de seients per fila al teatre (int)");

int cadiresFilaTeatre = sc.nextInt();
sc.nextLine();
System.out.println("Nom de la obra:");

String nomObra = sc.nextLine();



System.out.println("Durada de l'obra en minuts (int):");

int duradaObra = sc.nextInt();
sc.nextLine();
System.out.println("Autor de l'obra:");

String autorObra = sc.nextLine();

System.out.println("Restriccio adults nomes?(true/false)");

boolean adultsObra = sc.nextBoolean();
sc.nextLine();
```

 Generalitat de Catalunya Departament d'Ensenyament Institut Montsià	Mòdul 03 Cas 5	
Professorat: Maria Merino Sanjuán Alumnat: Anwar El Khattabi - Marc España - Alberto Dos Santos	Curs: ASIX_DAW_1B	

Una vegada tenim els valors dels objectes els creem.

```
Obra obraprova = new Obra(nomObra, duradaObra, autorObra, adultsObra);



Teatre provateatre = new Teatre(obraprova,preuEntrada,filesTeatre,cadiresFilaTeatre);
```

Després tenim que crear els clients, per fer això hem creat un array d'una llargada que marcarà l'usuari, que sirà el total de clients dins del teatre.

```
System.out.println("Quants clients vols crear? (int)");
Client[] clients = new Client[sc.nextInt()];
sc.nextLine();
for (int e = 0; e < clients.length; e++) {
    System.out.println("Introdueix el nom del client (String)");
    String nomClient = sc.nextLine();
    System.out.println("Introdueix la edat del client (int)");
    int edatClient = sc.nextInt();
    sc.nextLine();
    System.out.println("Introdueix els diners del client (int)");
    int dinersClient = sc.nextInt();
    sc.nextLine();
    clients[e] = new Client(nomClient, edatClient, dinersClient);
    if (obraprova.getAdults()) {
        if (clients[e].majorEdat() == false) {
            continue;
        }
    }
    if (!clients[e].pagarEntrada(preuEntrada)) {
        System.out.println(clients[e] + "\" Not enough money for client: "+ clients[e].nom+"\"");
        continue;
    }

    int filaClient,seientClient;
    do {
        System.out.println("Introdueix la fila que el vols situar(int)");
        filaClient = sc.nextInt();
        sc.nextLine();
        System.out.println("Introdueix el seient de la fila "+filaClient+" (int)");
        seientClient = sc.nextInt();
        sc.nextLine();
    } while (provateatre.seientOcupat(filaClient, seientClient) != "[0]");
    provateatre.setSeient(new Seient(filaClient, seientClient, clients[e]));
}
System.out.println(provateatre);
```

1. Demanem les dades de l'usuari i creem el Client.
2. Comprovem si la Obra és per a majors d'edat i si ho hes comprovem el client en cas que no pugui entrar podem veure "continue;" això fa que el bucle for canviï d'iteració i vagi a la següent sense executar més codi de la actual.
3. Comprovem si el client té prous diners per a pagar la entrada i utilitzem la mateixa declaració (continue).

 Generalitat de Catalunya Departament d'Ensenyament Institut Montsià	Mòdul 03 Cas 5	
Professorat: Maria Merino Sanjuán Alumnat: Anwar El Khattabi - Marc España - Alberto Dos Santos		Curs: ASIX_DAW_1B

4. Demanem al usuari una fila i un seient per a crear un seient. Aquest codi l'hem posat dins d'un while per a que pugui tornar a introduir un altre seient en cas de que hagi introduït un d'ocupat.
 Seguidament creem el seient i el posem a l'array del teatre amb `Teatre.setSeient()`;



Generalitat de Catalunya
Departament d'Ensenyament
Institut Montsià

Mòdul 03 Cas 5

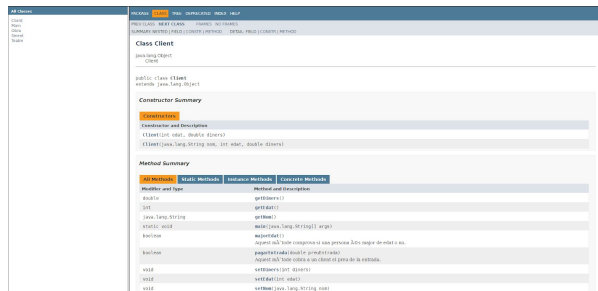


Professorat: Maria Merino Sanjuán
Alumnat: Anwar El Khattabi - Marc España - Alberto Dos Santos

Curs: ASIX_DAW_1B

JavaDocs

Javadoc és un programa d'Oracle que serveix per a generar una documentació a partir d'un codi Java. En el nostre programa hem afegit els comentaris necessaris damunt de cada mètode per a generar aquesta documentació.

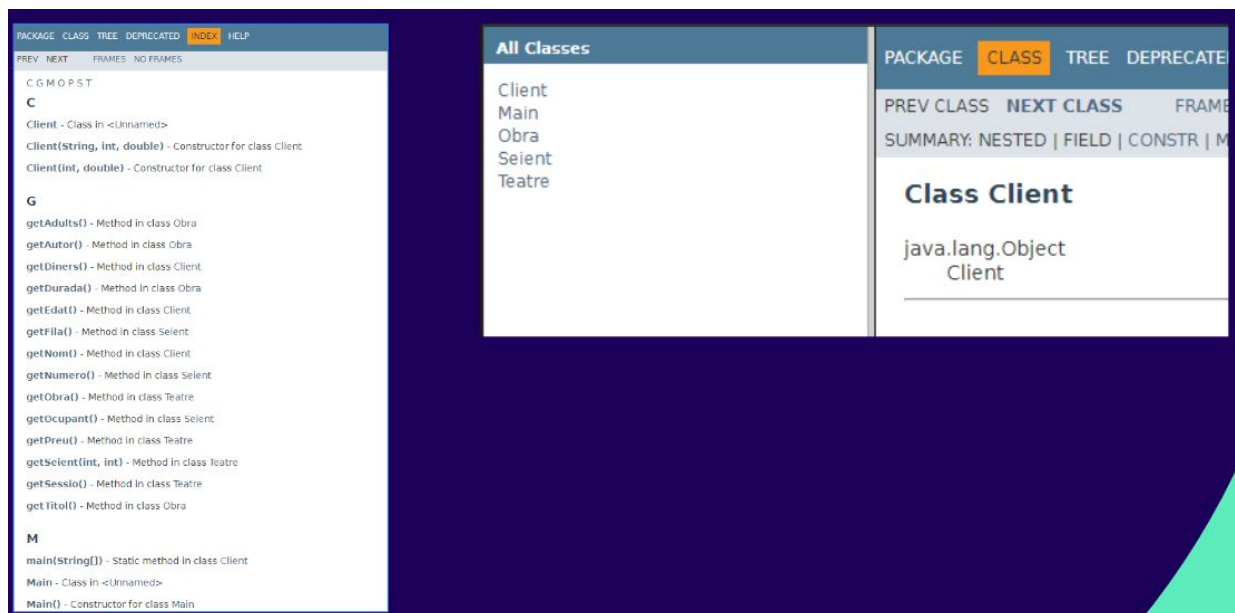




Podem visualitzar totes les funcions i la descripció dels mètodes.

En aquest cas hem après a executar la eina javadoc a múltiples documents a la vegada.

```
>javadoc *.java
```

També tindrem un javadoc més complet ja que podem veure tots els mètodes de totes les classes a un mateix lloc.





 Generalitat de Catalunya Departament d'Ensenyament Institut Montsià	Mòdul 03 Cas 5	
Professorat: Maria Merino Sanjuán Alumnat: Anwar El Khattabi - Marc España - Alberto Dos Santos		Curs: ASIX_DAW_1B

Conclusions

En aquest treball hem millorat bastant la programació en general, ja que al treballar amb POO “Programació Orientada a Objectes” ens veiem obligats a utilitzar diferents documents per a realitzar un programa més polit i entendible ja sigui per a fer canvis o bé com per a que un tercer tingui una major facilitat.

Els punts més importants que hem destacat d'aquest treball són:

- Veure com podem utilitzar els atributs d'una classe a una altra.
- Conèixer la funcionalitat dels constructors.
- Entendre la funció de “seguretat” dels getters i setters per als atributs privats.
- Imprimir les dades mitjançant el mètode toString.
- Utilitzar funcions cridant-les des de altres classes.

 Generalitat de Catalunya Departament d'Ensenyament Institut Montsià	Mòdul 03 Cas 5	
Professorat: Maria Merino Sanjuán Alumnat: Anwar El Khattabi - Marc España - Alberto Dos Santos		Curs: ASIX_DAW_1B

Webgrafia

Paquetes en Java: Definición, Uso y Ejemplos

<https://javadesdecero.es/intermedio/paquetes-en-java/>

Curso Java desde cero #21 | Arreglos bidimensionales (Matrices):

<https://www.youtube.com/watch?v=bPVcx44wgmQ>

Using the this Keyword:

<https://docs.oracle.com/javase/tutorial/java/javaOO/thiskey.html>

Ejercicio Básico POO Java. Clase Cuenta:

<http://puntocomnoesunlenguaje.blogspot.com/2015/04/ejercicio-basico-poo-java.html>