Getters

i

Setters

1.- Encapsulament

Un pilar fonamental de la programació orientada a objectes (POO) és **l'encapsulament**:

"Es denomina **encapsulament** a l'ocultació de **l'estat d'un objecte**, és a dir, de les **dades membre** d'un objecte, **(atributs i mètodes)** de manera que només es puga canviar si nosaltres així ho permetem, i sempre a través de les operacions que definim per a este propòsit.

Cada objecte està aïllat de l'exterior. L'aïllament **protegeix les dades associades a un objecte contra la seua modificació per qui no tinga dret a accedir a ells**, eliminant efectes secundaris i interaccions.

D'aquesta manera **un usuari de la classe pot obviar la implementació dels mètodes i propietats per a concentrar-se només en com usar-los.**

D'altra banda s'evita que l'usuari puga canviar el seu estat de maneres imprevistes i incontrolades.

Per això, una pràctica molt habitual en POO consisteix en ocultar tots els atributs (es a dir fer-los **private**) perquè no es puguen modificar directament des de fora de la classe.

En el seu lloc afegirem **mètodes getters** i **setters** visibles (**public**) que permeten llegir i modificar aquests atributs des de fora de la classe.

Els mètodes get i set són simples mètodes que utilitzem en les classes per mostrar (get) o modificar (set) el valor d'un atribut.

Per conveni,

- El nom del mètode sempre serà **get** o **set** acompanyat del nom de l'atribut. (**getNom** o **setNom**)
- El seu modificador serà sempre **públic**, ja que volem mostrar o modificar l'atribut des de fora de la classe.

2.- Creació de getters i setters

Hi ha una estructura per a crear els **getters** i **setters.**

En els mètodes **getters** sempre ens r**etornarà el valor** de l'atribut sense necessitat de passar **cap paràmetre.**

```
public tipus_de_atribut getAtribut (){

     return atribut;
}
```

En els mètodes **setters** sempre ens **demanarà algun valor** com a paràmetre per a guardar-lo a l'atribut de la classe, i aquest **mai haurà de retornar algun valor**.

```
public void setAtribut (tipus_de_atribut variable){

    this.atribut = variable;
}
```

Recorda: El nom del mètode depén de cada programador, però per a prendre un bon estil de programació és recomanable anteposar la paraula **get** o **set** i **el nom de l'atribut** utilitzant la notació **camelCase.**

Exemple:

```
public class Persona {

	// ATRIBUTS

	private String nom;
	private String cognoms;
	private int edat;
	private double altura;
	private double pes;
	private char genere;


	// GETTER'S i SETTER'S

	public String getNom() {
		return nom;
	}

	public void setNom(String nom) {
		this.nom = nom;
	}

	public String getCognoms() {
		return cognoms;
	}

	public void setCognoms(String cog-
noms) {
		this.cognoms = cognoms;
	}

	public int getEdat() {
		return edat;
	}

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

	public double getAltura() {
		return altura;
	}

	public void setAltura(double altu-
ra) {
		this.altura = altura;
	}

	public double getPes() {
		return pes;
	}

	public void setPes(double pes) {
		this.pes = pes;
	}

	public char getGenere() {
		return genere;
	}

	public void setGenere(char genere)
{
		this.genere = genere;
	}

}// De la classe.
```

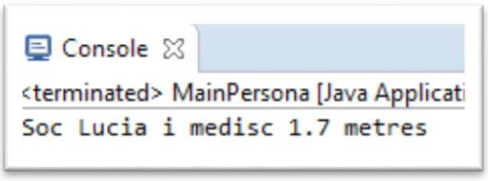
Al alterar els atributs de la classe amb el modificador d'accés **private** no podem accedir a ells des de el programa principal com fèiem als exemples anteriors, per exemple:

```
	public static void main(String[] args) {
		// Creem l'objecte p1, es a dir una instancia de la classe PERSONA.

		Persona p1 = new Persona();

		// Assignem valors als atributs de l'objecte
		//(definits en la classe PERSONA)
		p1.nom = "Lucia";
		p1.cognoms = "García";
		p1.edat = 32;
		p1.altura = 1.70;
		p1.pes = 58.5;
		p1.genere = 'F';

		System.out.println("El nom de la persona és: " + p1.nom);

	}
```

Totes estes assignacions (i el `println()`) donarien **error**, ja que hem declarat els atributs com **private**.

Per accedir als membres o atributs dels objectes hem de fer:

```
  public static void main(String[] args) {

	Persona p1 = new Persona();

	// Modificació dels atributs de l'objecte

	p1.setNom("Lucia");
	p1.setCognoms("García");
	p1.setEdat(32);
	p1.setAltura(1.70);
	p1.setPes(58.5);
	p1.setGenere('F');

	// Accés als atributs de l'objecte

	String cad1, cad2;
	int edatDeLaPersona;

	cad1=p1.getNom();
	cad2=p1.getCognoms();
	edatDeLaPersona=p1.getEdat();

	System.out.println("Soc " + p1.getNom() + " i medisc " + p1.getAltura()
+ " metres");
  }
```



Amb els **getters** i **setters** podem controlar l'accés i la modificació dels atributs dels objectes. A declarar els atributs amb el modificador **private** impedim l'accés als mateixos. Sols permetrem l'accés i/o la modificació d'aquells que declarem prèviament el seu mètode **get** o **set**.

Per exemple: En la següent definició de la classe Persona, sols podrem llegir i modificar el atribut nom, i sols podrem llegir l'atribut cognom (que **NO** modificar).

La resta d'atributs no serien accessibles.

```
public class Persona {

	// ATRIBUTS

	private String nom;
	private String cognoms;
	private int edat;
	private double altura;
	private double pes;
	private char genere;


	// GETTER'S i SETTER'S

	public String getNom() {
		return nom;
	}

	public void setNom(String nom) {
		this.nom = nom;
	}

	public String getCognoms() {
		return cognoms;
	}


}// De la classe.
```