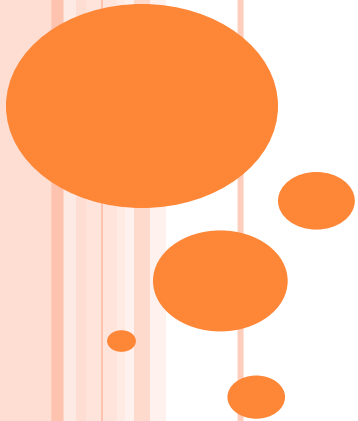


OTROS ELEMENTOS DE UNA CLASE



MÉTODOS ESTÁTICOS

- Su ejecución no depende de una instancia concreta.
- Se definen con la palabra `static` y solo operan con los parámetros recibidos y otros miembros estáticos

```
class Mesa{  
    int alto;  
    int ancho;  
    static String fabricante;  
    static void imprimir(String dato){  
        System.out.println(fabricante); //ok  
        System.out.println(ancho); //error  
    }  
}
```

- Se les llama con el nombre de la clase seguido del punto:


`Mesa.imprimir("hola");`

CONSTRUCTORES

- Bloques de código que se ejecutan al crear un objeto de una clase. Pueden recibir parámetros
- Se definen con el nombre de la clase, sin tipo de devolución
- Utilizados habitualmente para inicializar atributos

```
Mesa m = new Mesa(10, 2, "verde");
```

```
class Mesa{  
    int largo;  
    int ancho;  
    String color;  
    public Mesa(int l, int a, String c){  
        largo=l;  
        ancho=a;  
        color=c;  
    }  
}
```



MODIFICADORES DE ACCESO

➤ Una clase y sus componentes pueden definirse con los siguientes modificadores de acceso:

▪ **public.** El elemento puede utilizarse desde fuera de su paquete. Empleado habitualmente para clases, constructores y métodos

```
package p1;  
public class Mesa{  
    :  
    public int superficie(){  
        :  
    }  
    :  
}
```

```
package p2;  
import p1.Mesa;  
public class Test{  
    public void metodo(){  
        Mesa m=new Mesa();  
        m.superficie();  
    }  
    :  
}
```



MODIFICADORES DE ACCESO II

▪(default). Equivale a no usar ningún modificador e implica que el elemento (clase, método, constructor o atributo) es accesible solo dentro de clases de su mismo paquete:

```
package p1;  
public class Mesa{  
    int superficie(){  
        :  
    }  
}  
class Prueba{  
    public void metodo(){  
        Mesa m=new Mesa();  
        m.superficie();//ok  
    }  
}
```

```
package p2;  
import p1.Mesa;  
public class Test{  
    public void metodo(){  
        Mesa m=new Mesa(); //ok  
        m.superficie(); //error de compilación  
    }  
    :  
}
```



MODIFICADORES DE ACCESO III

- **private.** No está permitido en clases. El miembro solo es accesible desde el interior de la clase. Muy habitual en atributos para encapsulación

```
public class Mesa{
    private int largo;
    private int ancho;
    private String color;
    :
}
class Otra{
    void metodo(){
        Mesa m=new Mesa();
        m.largo=2; //error compilación
    }
}
```

