

- Sintaxis mínima de declaración de una clase

```
[public] class Identificador
{
    declaraciones de variables miembros
    declaraciones de métodos miembros
}
```

- Aunque el orden entre las variables y los métodos es flexible.
- El significado de **public** se estudiará más adelante.

Declaración de variables y funciones miembro



CLASES

- Sintaxis mínima de declaración de variables miembro

TipoVariable Identificador

- Sintaxis mínima de declaración de métodos

```
[public] tipoRetorno Identificador( [parámetros] )  
{  
    declaraciones de variables locales  
    sentencias  
    [return [(] expresión [)] ]  
}
```

- Propiedades
 - Método especial que se llama automáticamente siempre que se crea un objeto de una clase. Su principal función es iniciar los atributos.
 - Su nombre es el mismo que el de la clase a la que pertenece.
 - No tienen valor de retorno.
 - Excepto en casos excepcionales los constructores se declaran públicos.
 - El compilador proporciona a todas las clases declaradas un constructor público por defecto (sin parámetros), y que no hace nada, pero que es necesario porque cada vez que se crea un nuevo objeto se llama al constructor de la clase.

- Proceso de creación de objetos en Java
 - Para crear un objeto de una clase hay que utilizar el operador **new** y un constructor de la clase.

```
nombreClase nombreObjeto = new nombreClase( [parámetros] )
```

- Acceso o variables y funciones miembro de clases desde código Java:
 - La sintaxis para el acceso a estas componentes es la siguiente (hay un punto entre el nombre del objeto y de la componente)
 - Para acceder al valor de una variable miembro.

```
nombreObjeto.nombreVariableMiembro
```

- Para llamar a un método.

```
nombreObjeto.nombreVariableMiembro( [parámetros] )
```

- **Ejemplo:** declaración de la clase *Círculo*.

```
public class Circulo{
```

```
    double radio, cx, cy; —————→ Declaración de las variable miembro
```

```
    public Circulo (double x, double y, double r) { —→ Constructor
```

```
        cx = x;
```

```
        cy = y;
```

```
        radio = r; }
```

```
    double area(){
```

```
        return Math.PI * radio * radio;}
```

```
}
```

Declaración de los métodos o funciones miembro.
En el código se accede a un atributo con .

```
...
```

```
    Circulo c = new Circulo(3.5, 2.6, 1.0);
```

```
    a=c.area();
```

```
...
```

Llamada al método area del círculo c

- El proceso de desarrollo de la aplicación más simple Java se puede resumir de la siguiente manera
 1. Tantos ficheros con extensión **.java** como clases (públicas) se necesiten en los que el nombre de la clase sea el nombre del fichero.
 2. Un fichero con una clase (pública) con el método **main**

public static void main(String [] args)

3. Compilación de cada fichero **.java** (ej. comando DOS)

javac nombreClase.java

4. Se consigue un fichero con extensión **.class** (que puede "ejecutar directamente el intérprete") por cada fichero **.java**.
5. Ejecución de la aplicación (ej. desde DOS) con el comando siguiente (debe haber un fichero de nombre **nombreClaseMain.class** asociado a la clase con el método **main**)

java nombreClaseMain