MI PRIMERA CLASE

INTRODUCCIÓN A LA ORIENTACIÓN A OBJETOS

OpenWebinars

ORIENTACIÓN A OBJETOS EN JAVA

- Java es *totalmente* orientado a objetos.
- En Java, “todo” es un objeto.
  - No como en C, C++, ...
- Hay una única sintaxis, consistente, que usaremos siempre.

MI PRIMERA CLASE

- Mejor comencemos con un ejemplo

Estructura

Comportamiento

```java
package orientacionobjetos;

public class Coche {

    private String marca;
    private String modelo;
    private int anio;

    public Coche(String marca, String modelo, int anio) {
        this.marca = marca;
        this.modelo = modelo;
        this.anio = anio;
    }

    public void arrancar() {
        System.out.println("El coche %s %s %d ha arrancado"
                    .formatted(marca, modelo, anio));
    }


}
```

MI PRIMERA CLASE

- Mejor comencemos con un ejemplo

```
1  package orientacionobjetos;
2
3  public class App {
4
5      public static void main(String[] args) {
6
7          Coche coche = new Coche("Ford", "Focus", 2020);
8          coche.arrancar();
9
10     }
11
12 }
```

Declaración e instanciación

ELEMENTOS A TENER EN CUENTA

- **Definición** e **implementación** de una clase se hace en un mismo archivo (en otros lenguajes no).
- El **archivo** que tiene la clase Java debe tener el **mismo nombre** que la **clase** (public).
- Por ahora, una **única clase** (public) **por fichero**.
- Los nombres de las clases deben ser **sustantivos**
  - Persona, Coche, Factura, Libro, ...
  - Notación *Upper Camel Case*

LAS CLASES SON NUEVOS TIPOS DE DATO

- Java tiene un conjunto *extensible* de tipos de datos.
- Tipos primitivos: int, float, char, boolean, ...
- Tipos no primitivos (o estructurados):
  - Algunos ya proporcionados (String, ...)
- **Nuestras clases son nuevos tipos de datos.**
- Podemos definir *variables* de ese tipo.
  - A la inicialización se le conoce como **instanciación**.

INSTANCIACIÓN DE OBJETOS

- Creación de objetos a partir de la plantilla (clase).
- Uso de la palabra reservada **new**.
- Uso de un constructor.
- A la variable del tipo de dato de la clase se le conoce como referencia.

```
Coche coche = new Coche("Ford", "Focus", 2020);
```

referencia

constructor

REFERENCIA A SÍ MISMO

- Todos los objetos pueden referenciarse a sí mismos.
- Siempre existe una referencia llamada **this**.
- *Apunta* al objeto donde la estamos usando.
- Sirve para diferenciar *variables* que se llaman igual.

```
public Coche(String marca, String modelo, int anio) {
    this.marca = marca;
    this.modelo = modelo;
    this.anio = anio;
}
```

PAQUETES

- Mejor comencemos con un ejemplo

Paquete

```java
package orientacionobjetos;

public class Coche {

    private String marca;
    private String modelo;
    private int anio;

    public Coche(String marca, String modelo, int anio) {
        this.marca = marca;
        this.modelo = modelo;
        this.anio = anio;
    }

    public void arrancar() {
        System.out.println("El coche %s %s %d ha arrancado"
                    .formatted(marca, modelo, anio));
    }


}
```

PAQUETES

- Unidad organizativa.
- Puede agrupar una o más clases.
- Organización jerárquica: paquetes dentro de paquetes
- A nivel práctico, son directorios dentro de nuestro proyecto.
- Eclipse los representa de una forma especial.