

Depuración de objetos

Esteban Álvarez

Depuración

- En la unidad anterior vimos cómo depurar programas sencillos
- Pero ahora es necesario saber cómo **depurar los objetos**

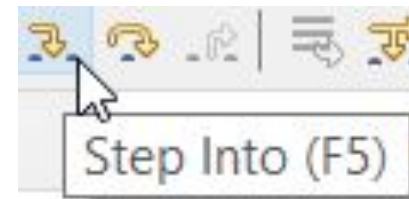
Depuración. Step over



- Hasta ahora siempre depuramos utilizando la opción **step over**
- Esa acción **ejecuta la línea actual** directamente sin entrar en las acciones que realiza ese código

```
➡ boolean resultado = coche.acelerar(50);  
➡ if (resultado) {
```

Depurar un método



STEP INTO

- Pero con step into podemos **depurar los métodos** de esa línea

```
➡ boolean resultado = coche.acelerar(50);  
➡ if (resultado) {
```





```
public boolean acelerar(int aceleracion) {  
➡ if (arrancado) {  
➡ velocidad += aceleracion;  
➡ return true;  
} else {  
    return false;  
}  
}
```

Depurar un método

BREAKPOINT

- Otra forma de depurar un método es poniendo un **breakpoint** dentro del método
- La depuración comienza cuando algún objeto llama a ese método

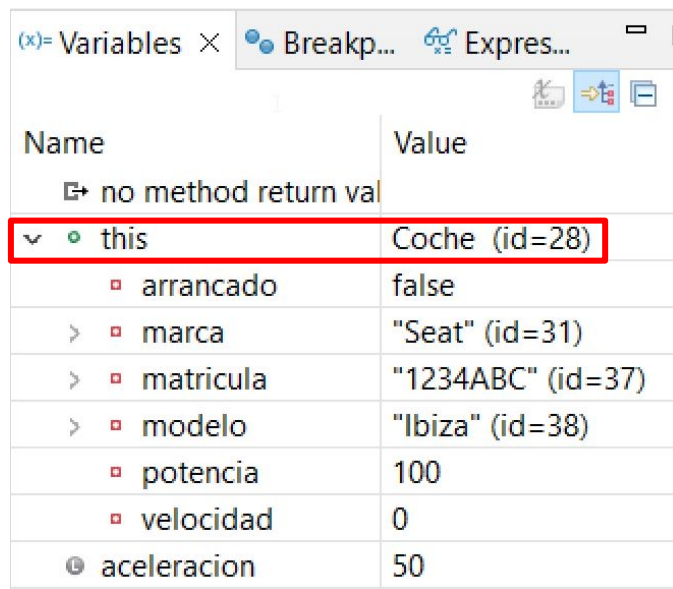
```
public boolean acelerar(int aceleracion) {  
  
      if (arrancado) {  
        velocidad += aceleracion;  
        return true;  
    } else {  
        return false;  
    }  
}
```

Depurar un método

INFORMACIÓN

- Una vez que estamos depurando un método, es posible ver los **atributos** del objeto y variables del método
- **IMPORTANTE:** El objeto actual es **this**

```
public boolean acelerar(int aceleracion) {  
    ➡ if (arrancado) {  
        velocidad += aceleracion;  
        return true;  
    } else {  
        return false;  
    }  
}
```



(x)= Variables × Breakp... Expres...	
Name	Value
↳ no method return val	
▼ this	Coche (id=28)
▪ arrancado	false
> ▪ marca	"Seat" (id=31)
> ▪ matricula	"1234ABC" (id=37)
> ▪ modelo	"Ibiza" (id=38)
▪ potencia	100
▪ velocidad	0
ⓘ aceleracion	50

Finalizar depuración del método



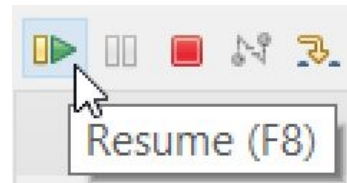
STEP RETURN

- Si cuando estamos depurando un método queremos dejar de hacerlo y volver a **depurar la clase que llamó** a ese método usamos **Step return**

```
boolean resultado = coche.acelerar(50);  
➡ if (resultado) {
```

```
public boolean acelerar(int aceleracion) {  
➡ if (arrancado) {  
    velocidad += aceleracion;  
    return true;  
} else {  
    return false;  
}  
}
```

Finalizar depuración del método



RESUME

- Si cuando estamos depurando un método queremos dejar de hacerlo y volver a depurar en el siguiente breakpoint usamos **Resume**

```
boolean resultado = coche.acelerar(50);  
if (resultado) {  
    System.out.println("Se pudo acelerar");  
} else {  
    System.out.println("No se pudo acelerar");  
}  
System.out.println("Datos del coche: " + coche);
```

```
public boolean acelerar(int aceleracion) {  
    if (arrancado) {  
        velocidad += aceleracion;  
        return true;  
    } else {  
        return false;  
    }  
}
```


FIN!