

1. El método main principalmente saca los datos de los coches, en este caso el c, c2 y el c3. Luego se saca un número random de los 3 coches sus velocidades, velocidad 1, 2 y 3.

```
c2.acelera(velocidad_nueva1);
c3.acelera(velocidad_nueva2);
c.acelera(velocidad_nueva3);
```

En esta imagen muestra el tipo de coche, c, c2, c3.

Con la función acelera lo que haces es sumar las velocidades

```
c2.acelera(velocidad_nueva1);
c3.ace
c.acel
public void acelera(
    int vel
)
System
System
```

3 usages

```
public void acelera(int vel) {
    this.velocidad += vel;
}
```

Con ese this hace el llamamiento de la velocidad almacenada que se va a sumar.

```
c2.acelera(velocidad_nueva1);
c3.acelera(velocidad_nueva2);
c.acelera(velocidad_nueva3);
```

Reassigned local variable

```
int velocidad_nueva1 = aleatorio.nextInt(80)
```

```
c2.acelera(velocidad_nueva1); c2: Coche@735 velocidad_nueva1: 69
c3.acelera(velocidad_nueva2); c3: Coche@736 velocidad_nueva2: 50
c.acelera(velocidad_nueva3); c: Coche@734 velocidad_nueva3: 45
```

```
public void acelera(int vel) {
    this.velocidad += vel;
}
```

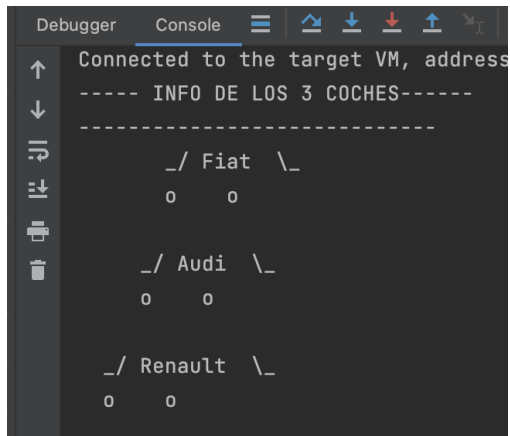
vel: 45
vel: 45 velocidad: 45

```
public void pinta() {
    int i=0; i: 4
    for(i=0; i<this.velocidad/10; i++) i: 4 velocidad: 49
        System.out.print(" ");
    System.out.println(" / "+ this.marca+ "\\"); marca: "Renault"
```

Junto con esta función llamada pinta lo que hace es que con el número random dado en este caso en la captura el 49 del coche Renault que es el c3, empezamos diciendo que "i" es 0, como 0 es menor que "this.velocidad" que es 49 y encima tiene que ser dividido por "10" dará como resultado 4, cosa que se tiene que iterar 4 veces System.out.print(" "); que hará que se desplace 4 posiciones a la derecha y luego se corta y salga del for y imprima System.out.println(" / "+ this.marca+ "\\ ");

y hace lo mismo con el segundo for pero para imprimir las "ruedas".

----Siguiente página----



```

Debugger Console
Connected to the target VM, address
----- INFO DE LOS 3 COCHES-----
-----
  _/ Fiat  \_
    0    0

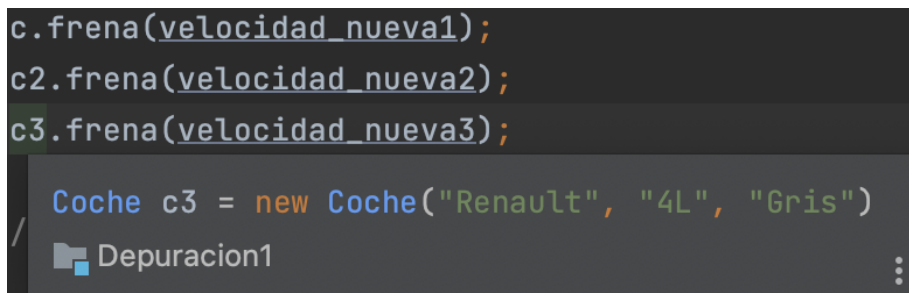
  _/ Audi  \_
    0    0

  _/ Renault \_
    0    0

```

Se imprimirá lo siguiente.

Luego con el frenado viene siendo lo mismo solo que en vez de la función de acelerar es el de frenar.

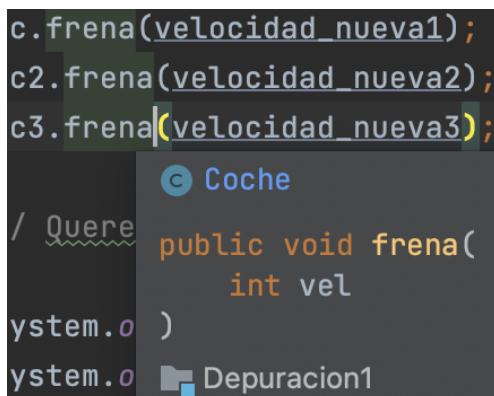


```

c.frena(velocidad_nueva1);
c2.frena(velocidad_nueva2);
c3.frena(velocidad_nueva3);

Coche c3 = new Coche("Renault", "4L", "Gris")
/ Depuracion1

```

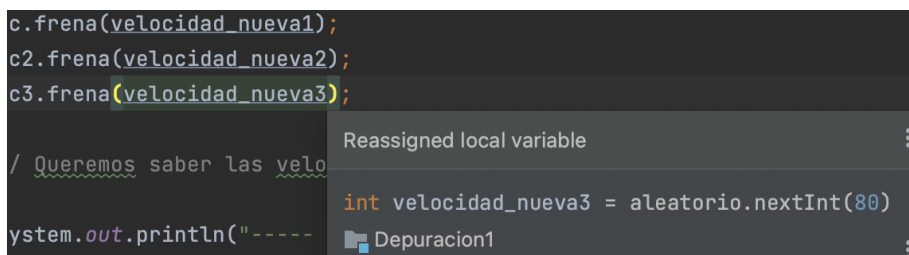


```

c.frena(velocidad_nueva1);
c2.frena(velocidad_nueva2);
c3.frena(velocidad_nueva3);

/ Quere public void frena(
  int vel
ystem.o )
ystem.o Depuracion1

```



```

c.frena(velocidad_nueva1);
c2.frena(velocidad_nueva2);
c3.frena(velocidad_nueva3);

/ Queremos saber las velo
ystem.out.println("-----
Depuracion1

```

con las mismas características que con el de acelerar pero en la función de frenar. En la siguiente página se explica lo de la resta:

```

velocidad_nueva1 = aleatorio.nextInt( bound: 100);
velocidad_nueva2 = aleatorio.nextInt( bound: 100);
velocidad_nueva3 = aleatorio.nextInt( bound: 100);   aleatorio: Random@762

c.frena(velocidad_nueva1);   c: Coche@739   velocidad_nueva1: 63
c2.frena(velocidad_nueva2);  c2: Coche@760   velocidad_nueva2: 51
c3.frena(velocidad_nueva3);  c3: Coche@761   velocidad_nueva3: 80

```

Saca los datos del número aleatorio (y esto no lo tengo claro pero como esta a 100 lo que estás haciendo es aumentar más el número para que a la hora de la resta se mueva a la izquierda, en plan para que el número sea más grande y la resta lo sea igual), una vez almacenado los número se dirige a la función de la resta y hará lo siguiente.

```

for(i=0; i<this.velocidad/10 = false ; i++)   i: 0   velocidad: 0
    System.out.print(" ");
    System.out.println(" _/ " + this.marca+" \\_");

```

Al dar el número como la velocidad está declarada como 0, con esto no entra a los espacios del bucle y directamente se pasa a imprimir el modelo del coche.

```

for(i=0; i<this.velocidad/10 = false ; i++)   i: 0   velocidad: 0
    System.out.print(" ");
    System.out.println(" o   o ");

```

Se pasa al segundo bucle y pasa lo mismo, cosa que no imprime los espacios y pasa directamente a imprimir las “ruedas”. Y así hasta iterar a los 3 “coches”.

2. Las velocidades_nuevasX son estas:

```

Random aleatorio = new Random();   aleatorio: Random@739
int velocidad_nueva1 = aleatorio.nextInt( bound: 80);   velocidad_nueva1: 23
int velocidad_nueva2 = aleatorio.nextInt( bound: 80);   velocidad_nueva2: 13
int velocidad_nueva3 = aleatorio.nextInt( bound: 80);   velocidad_nueva3: 59

```

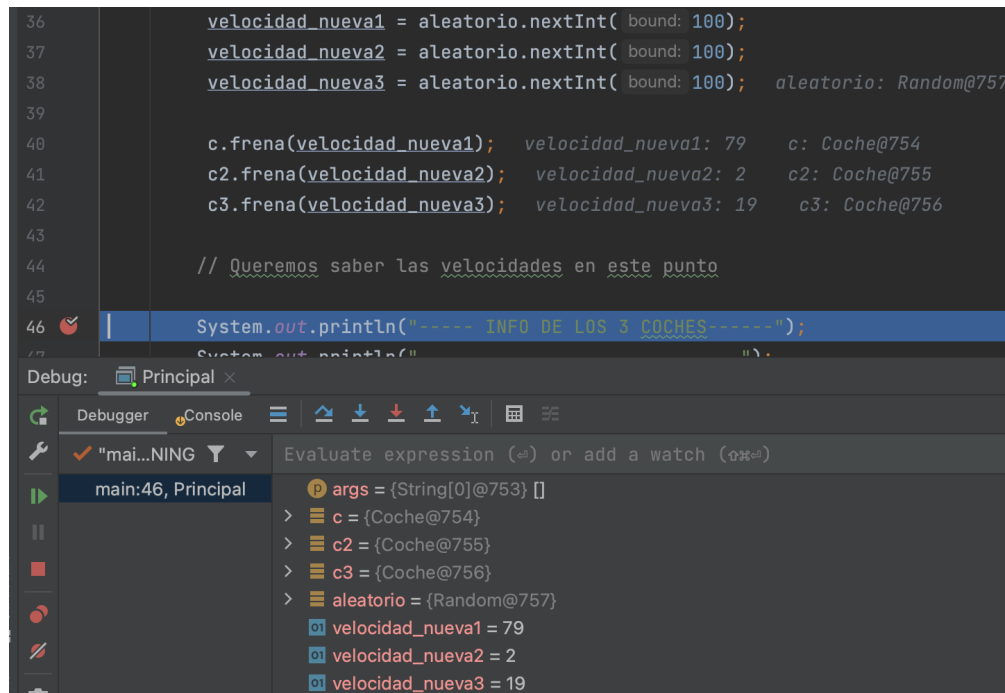
En la siguiente captura están los datos de los coches en la aceleración.

```

c2.acelera(velocidad_nueva1);   velocidad_nueva1: 23   c2: Coche@737
c3.acelera(velocidad_nueva2);   velocidad_nueva2: 13   c3: Coche@738
c.acelera(velocidad_nueva3);   velocidad_nueva3: 59   c: Coche@736

```

3. Esta es la captura en el breakpoint 46, en la que nos indica los datos de las velocidades de los 3 coches y en este caso se mete en la función de frenar.



```
36      velocidad_nueva1 = aleatorio.nextInt( bound: 100);
37      velocidad_nueva2 = aleatorio.nextInt( bound: 100);
38      velocidad_nueva3 = aleatorio.nextInt( bound: 100);    aleatorio: Random@757
39
40      c.frena(velocidad_nueva1);    velocidad_nueva1: 79    c: Coche@754
41      c2.frena(velocidad_nueva2);    velocidad_nueva2: 2    c2: Coche@755
42      c3.frena(velocidad_nueva3);    velocidad_nueva3: 19    c3: Coche@756
43
44      // Queremos saber las velocidades en este punto
45
46      System.out.println("----- INFO DE LOS 3 COCHES-----");
47      System.out.println(" ");
```

Debug: Principal x

Debugger Console

✓ "mai...NING" Evaluate expression (⌘) or add a watch (⌘⌘)

main:46, Principal

- args = {String[0]@753} []
- c = {Coche@754}
- c2 = {Coche@755}
- c3 = {Coche@756}
- aleatorio = {Random@757}
- velocidad_nueva1 = 79
- velocidad_nueva2 = 2
- velocidad_nueva3 = 19