


Open Opened 2 weeks ago by  **Rubén Montero**

Todavía más parámetros

Resumen

- Implementaremos otro método `printProduct` que reciba *distintos* parámetros, de *distintos* tipos
- Comprenderemos el concepto de *sobrecarga* (`overload`)

Descripción

Después de la tarea anterior, tendremos en `SimpleMathDemo` un método parecido a este:

```
public void printProduct(int num1, int num2) {  
    System.out.println(num1 * num2);  
}
```

Probemos a escribir otro método que reciba un parámetro formal tipo `String`, adicionalmente. Así comprobaremos que se pueden recibir parámetros de tipos *heterogéneos*:

```
public void printProduct(int num1, int num2) {  
    System.out.println(num1 * num2);  
}  
  
+ public void printProduct(String var, int num1, int num2) {  
+     // ...  
+ }
```

Podemos usar ese parámetro en el *cuerpo* del método.

Escribamos lo siguiente:

```
public void printProduct(String var, int num1, int num2) {  
    System.out.println("Hola," + " " + var);  
    System.out.println("Este es el producto de tus dos valores:");  
    System.out.println(num1 * num2);  
}
```

Probémoslo desde `Main.java`, enviando *valores cualesquiera* como parámetros actuales.

¿Ves cómo funciona?

La tarea

Ya está hecha.

Sólo **se pide** que tengas añadido el método `printProduct` que recibe **1** `String` y **2** `int`, como arriba.

Un momento

Hay dos métodos `printProduct` diferentes

Efectivamente, en nuestro camino hemos escrito **2** métodos con **el mismo nombre en la misma clase**:

```
public void printProduct(int num1, int num2) { /* ... */ }  
  
public void printProduct(String var, int num1, int num2) { /* ... */ }
```

Esto **está permitido**, y se conoce como **sobrecarga**.

Decimos que el método `printProduct` está **sobrecargado** con *distintos parámetros*.

Ojo

El principio de la *sobrecarga* consiste en indicar *distintos parámetros*.

No se puede sobrecargar un método indican *distintos tipos de dato devuelto*.

```
public int returnProduct() { /* ... */ }

// MAL
// Este método producirá un error de compilación
public float returnProduct() { /* ... */ }
```

Por último

Una vez verifiques que el *test* funciona correctamente, la tarea ha sido completada.

Haz **commit** y **push** para subir los cambios al repositorio.



Rubén Montero @ruben.montero changed milestone to [%Sprint 1](#) 2 weeks ago



Ania Blanco @ania.blanco mentioned in commit [0a79d79e](#) 2 weeks ago