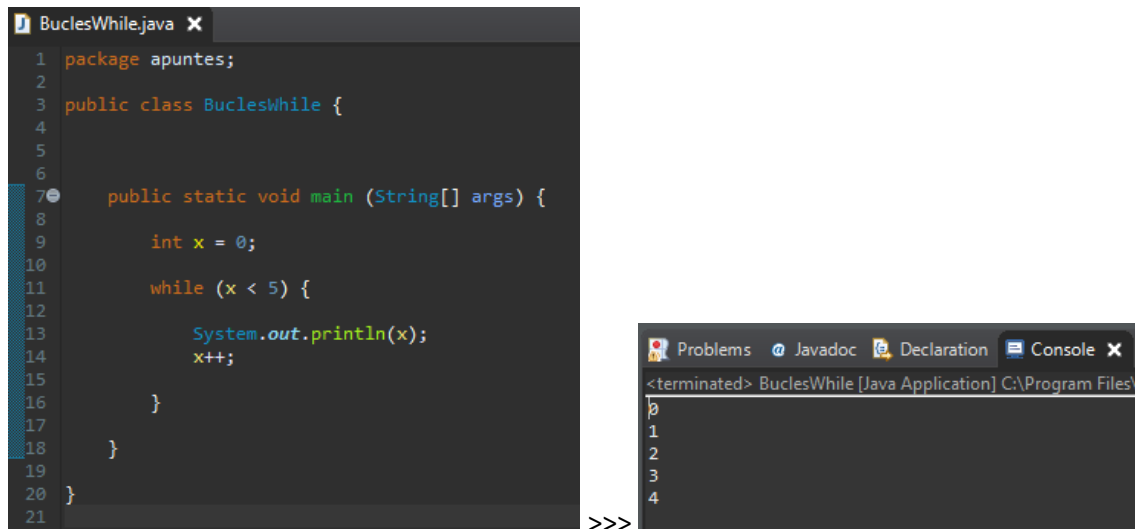


## WHILE

Si queremos realizar un bloque de código varias veces, o mientras se cumpla una condición, para ello utilizaremos bucles, ejemplo:



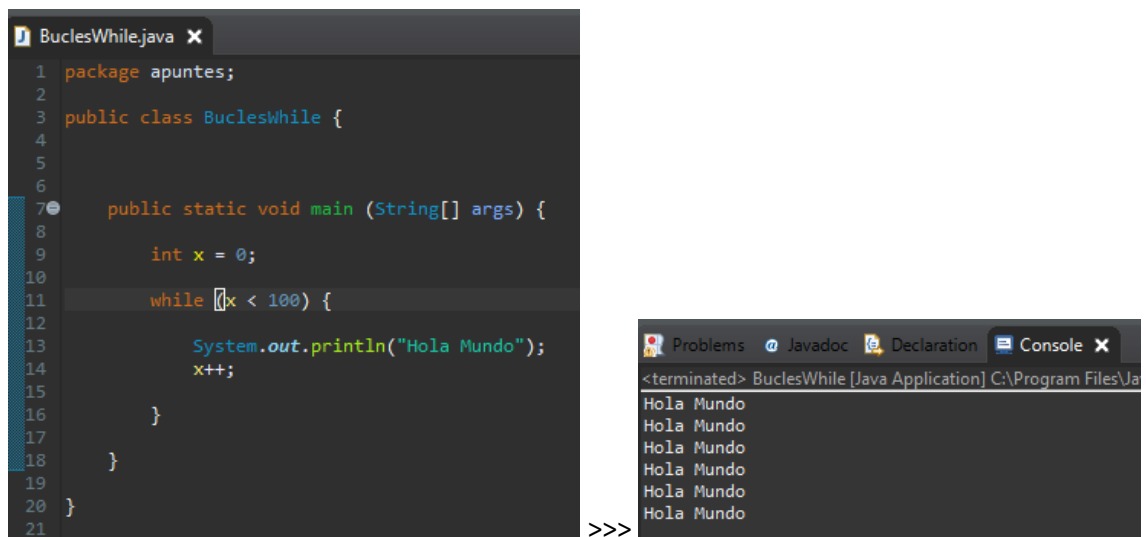
```
1 package apuntes;
2
3 public class BuclesWhile {
4
5
6
7     public static void main (String[] args) {
8         int x = 0;
9
10        while (x < 5) {
11            System.out.println(x);
12            x++;
13        }
14    }
15 }
```

>>>

```
<terminated> BuclesWhile [Java Application] C:\Program Files\
0
1
2
3
4
```

Como se ve, mientras se cumple la condición del `while`, las líneas desde la 8 a las 15 se repiten, se puede decir que: `while (true) { //BLOQUE_DE_CODIGO }`, es decir, mientras se cumpla una condición, se va a repetir el bloque de código.

Si queremos poner “Hola Mundo” 100 veces en pantalla podemos hacer lo siguiente:



```
1 package apuntes;
2
3 public class BuclesWhile {
4
5
6
7     public static void main (String[] args) {
8         int x = 0;
9
10        while (x < 100) {
11            System.out.println("Hola Mundo");
12            x++;
13        }
14    }
15 }
```

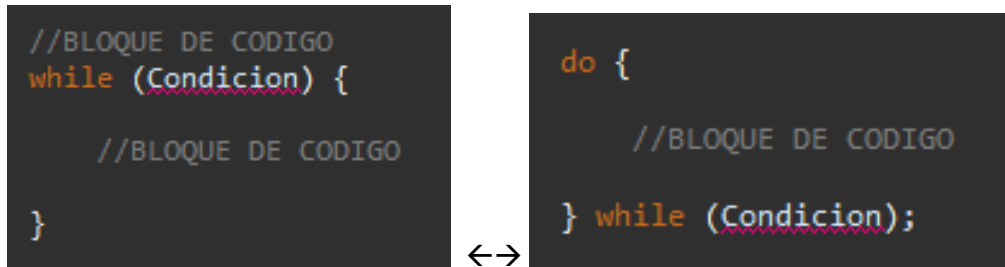
>>>

```
<terminated> BuclesWhile [Java Application] C:\Program Files\Ja
Hola Mundo
Hola Mundo
Hola Mundo
Hola Mundo
Hola Mundo
Hola Mundo
```

Conociendo el funcionamiento de los *ifs* este concepto no es muy complicado, básicamente lo dejaré en que *mientras (seCumpleEstaCondicion) { //HACER ESTO }*

## DO WHILE

Los *do while* hacen lo mismo que el *while*, pero **se realiza el bloque de dentro como mínimo una vez**, éstas 2 estructuras tienen el mismo funcionamiento:



```
//BLOQUE DE CODIGO
while (Condicion) {
    //BLOQUE DE CODIGO
}
```

↔

```
do {
    //BLOQUE DE CODIGO
} while (Condicion);
```

El segundo ejercicio de Bucles1 habrá que utilizar ésta sentencia, explicaré por qué en la descripción de la función.

Solamente recordad que no podéis poner *returns* o *breaks* dentro de los bucles.