



# ORACLE

## Academy



# Java Foundations

6-3

Uso de las sentencias break y continue

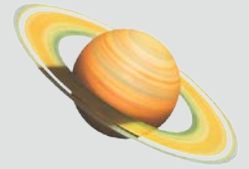
**ORACLE**  
Academy



# Objetivos

- En esta lección se abordan los siguientes objetivos:
  - Utilizar una sentencia break para salir de un bucle
  - Utilizar una sentencia continue para omitir parte de un bucle
  - Explicar la necesidad de comentarios de bucle





# Misión espacial a los anillos de Saturno

- Imaginemos un nuevo escenario para esta misión:
  - Mientras la nave espacial orbita alrededor de Saturno y toma instantáneas, el brazo robótico o la cámara se estropean.
- ¿Cómo resolvería este problema?
  - Si tuviera que escribirlo en un programa Java, ¿qué construcciones utilizaría?
  - Veamos si Java cuenta con una sentencia que permita terminar un bucle inmediatamente.

# ¿Cómo salir de un bucle antes de tiempo?

- Habitualmente, la única forma de salir de un bucle es que la condición de bucle que se evalúe sea falsa.
- Sin embargo, a veces es recomendable terminar un bucle antes de tiempo cuando se cumplen determinadas condiciones.
- En estos casos, que el bucle continúe sería una pérdida de tiempo del procesador.

# ¿Cómo salir de un bucle antes de tiempo?

- Puede utilizar dos sentencias Java para terminar un bucle antes de tiempo:
  - **break**
  - **continue**

# Uso de break en un bucle

- Cuando se ejecuta una sentencia break dentro de un bucle, la sentencia del bucle se termina inmediatamente.
- El programa continúa ejecutándose con la sentencia que le sigue a la sentencia de bucle.
- Sintaxis:

```
break;
```



# Uso de break en un bucle while

```
while(condition){  
    statement1;  
    statement2;  
    break;  
    statement3;  
    statement4  
}  
statement;
```

El control pasa a la sentencia  
fuera del bucle.

[sentencia fuera del bucle while]



# Ejemplo de uso de break en un bucle while

• Resultado: 0 1 2 3

- La ejecución del bucle se termina cuando el contador del bucle es igual que 4.

```
public static void main(String[] args) {  
    int i = 0;  
    while (i < 10) {  
        System.out.println(i + "\t");  
        i++;  
        if (i == 4) {  
            break;  
        }  
    }  
}
```

# Uso de break en un bucle for

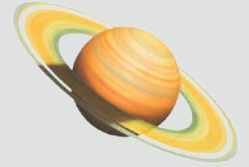
- Escribamos un programa para demostrar una sentencia break en un bucle for.
- El programa debe...
  - Leer 10 números de la consola.
  - Calcular la suma de los números que introduce el usuario.
  - Si el usuario introduce el número 999, terminar el bucle independientemente del valor del contador del bucle y sin agregarlo a la suma.

# Ejemplo de uso de break en un bucle for

```
public static void main(String[] args) {
    Scanner in = new Scanner(System.in);
    int numInputs = 10, input = 0, sum = 0, stopLoop = 999;
    System.out.println("Introduzca 10 números");
    for (int i = 0; i < numInputs; i++) {
        input = in.nextInt();
        if (input == stopLoop){
            break;
        }
        else {
            sum += input;
        } //endif
    } //end for
    System.out.println("La suma de los números:" + sum);
} //end method main
```



# Misión espacial a los anillos de Saturno: implantación de condiciones



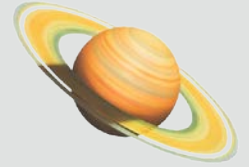
- Vamos a utilizar un bucle while y una sentencia break para implantar las condiciones especificadas al principio de la lección.

```
public static void main(String[] args) {  
    long distTravelled = 0;  
    long maxDistance = 50000000;  
  
    while (distTravelled <= maxDistance) {  
        if (isCameraBroken()) {  
            break;  
        }  
        else {  
            cameraSnap();  
        }  
    }  
    shipRotate();  
}
```

# Ejercicio 1

- Cree un nuevo proyecto y agréguele el archivo `ComputeSum.java`
- Examine `ComputeSum.java`.
- Implante las siguientes condiciones:
  - Aceptar 10 números del usuario
  - Calcular la suma de los números introducidos
  - Si se introduce 0, el programa debe salir y mostrar la suma de los números

# Misión espacial a los anillos de Saturno: un nuevo escenario



- Imaginemos un nuevo escenario para esta misión:
  - Mientras la nave espacial orbita alrededor de Saturno y toma instantáneas de sus anillos...
    - Si la visibilidad es 0, no se deben tomar instantáneas.
    - Si no es así, continuar tomando instantáneas.
- ¿Cómo resolvería este problema?
  - Si tuviera que escribirlo en un programa Java, ¿qué construcciones utilizaría?
  - Veamos si Java cuenta con una sentencia que permita omitir la iteración actual del bucle.

# Uso de continue en un bucle

- Es posible que en algún momento quiera omitir la iteración actual en un bucle sin que este se termine.
- Puede utilizar la sentencia continue para omitir la iteración actual en un bucle:
  - Es decir, el resto del cuerpo del bucle se omitirá hasta el final del mismo.
  - Sin embargo, el bucle no se terminará.
  - Cuando el programa alcance el final del bucle, volverá a probar la condición de continuación del bucle.
- Sintaxis:

```
continue;
```



# Uso de continue en un bucle while

```
while(condition){  
    statement1;  
    statement2;  
    continue;  
    statement3;  
    statement4  
}  
statement; [sentencia fuera del bucle while]
```

El control pasa a la condición del bucle.

Estas sentencias se omitirán en la iteración actual.

# Uso de continue en un bucle for

```
for (i = 0; i < 10; i++) {
```

```
    statement1;
```

```
    statement2;
```

```
    continue;
```

```
    statement3;
```

```
    statement4;
```

```
} //end for
```

El control pasa a la condición del bucle.

Estas sentencias se omitirán en la iteración actual.

# Uso de continue en un bucle for

- Resultado: 

0	1	2	3	5	6	7	8	9
---	---	---	---	---	---	---	---	---

  - El resultado no incluye el 4.
  - Debido a la sentencia continue, la ejecución del bucle se ha omitido cuando el contador del bucle es 4.

```
public static void main(String[] args) {  
    for (int i = 0; i < 10; i++) {  
        if (i == 4) {  
            continue; //control jumps to update i++  
        }//endif  
        System.out.print(i + "\t");  
    }//end for  
}//end method main
```

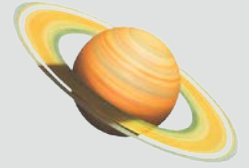
# Conclusiones

- Vamos a escribir un programa utilizando el bucle while y la sentencia continue.
- El programa debe...
  - Calcular la suma de los números entre 1 y 99 mediante el bucle while
  - Si el número es múltiplo de 10, la iteración actual se omitirá y el número no se agregará a la suma
  - Mostrar la suma en la consola

# Cálculo de la suma de los números

```
public static void main(String[] args) {  
    int counter = 0;  
    int sum = 0;  
    while (counter < 100) {  
        counter++;  
        if (counter % 10 == 0) {  
            continue;  
        }  
        else {  
            sum += counter;  
        }  
    }  
    System.out.println("Suma de 1 a 99: " + sum);  
}
```

¿Es múltiplo de 10? Si lo es, se omitirá la iteración actual.



# Misión espacial a los anillos de Saturno: implantación de condiciones

- Vamos a utilizar un bucle `while` y una sentencia `continue` para implantar las condiciones especificadas al principio de este tema.

```
public static void main(String[] args) {  
    long distTravelled = 0;  
    long maxDistance=50000000;  
  
    while (distTravelled <= maxDistance) {  
        if (getVisibility() == 0) {  
            continue;  
        }  
        else {  
            cameraSnap();  
        }  
    }  
    shipRotate();  
}
```

## Ejercicio 2

- Agregue el archivo `CountChar.java` al proyecto creado para el ejercicio 1
- Examine `CountChar.java`.
  - El programa se utiliza para contar el número de ocurrencias del carácter 'w' en la cadena.
  - Modifique el programa para...
    - Resolver el error de sintaxis.
    - Imprimir el recuento del carácter 'w'.
  - Resultado esperado:
    - Número de 'w': 3



## Ejercicio 3

- Agregue el archivo `BreakContinue.java` al proyecto creado para el ejercicio 1
- Examine `BreakContinue.java`.
- Modifique el programa mediante las sentencias `break` y `continue`.
  - Si el número es par, no se imprimirá.
  - La ejecución del bucle se debe parar cuando el valor del contador del bucle sea 7.

# Escritura de comentarios de bucle

- Agregar comentarios adecuados a los bucles es una buena práctica.
- De lo contrario...
  - El código puede ser confuso.
  - No se podrá entender la lógica fácilmente.
- Ayuda a entender...
  - Las variables del bucle utilizadas y su finalidad
  - La lógica del bucle
  - El número de iteraciones
  - Las ejecuciones de las sentencias del bucle en función de las condiciones, los criterios o ambos

# Ejemplo de cómo escribir comentarios de bucle

```
public static void main(String[] args) {  
  
    Scanner in = new Scanner(System.in);  
    int numInputs = 10, input = 0;  
  
    //This loop is executed 10 times  
    for (int i = 0; i < numInputs; i++) {  
        input = in.nextInt(); //user inputs a number  
  
        if (input % 2 == 0) { //if the number is even skip the  
            continue;        //remaining code and restart the loop  
        }//endif  
  
        System.out.println("Ese número es impar");  
    }//end for  
}//end method main
```

## Ejercicio 4

- Agregue el archivo `Divisors.java` al proyecto creado para el ejercicio 1
- Examine `Divisors.java`.
- El programa busca todos los divisores de un número.

## Ejercicio 4

- Modifique el programa para incluir comentarios para el bucle sobre...
  - Las variables del bucle utilizadas
  - La lógica del bucle
  - El número de iteraciones
  - La condición utilizada
  - El flujo de control en el bucle

# Resumen

- En esta lección, debe haber aprendido lo siguiente:
  - Utilizar una sentencia break para salir de un bucle
  - Utilizar una sentencia continue para omitir parte de un bucle
  - Explicar la necesidad de comentarios de bucle





# ORACLE

## Academy

