

The logo for Oracle Academy is centered on a light gray background. It features the word "ORACLE" in a bold, orange, sans-serif font. Below it, the word "Academy" is written in a smaller, dark gray, sans-serif font. The entire logo is framed by two horizontal dark gray bars, one at the top and one at the bottom.

ORACLE

Academy

Java Foundations

6-3

Uso de las sentencias break y continue

ORACLE
Academy



Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

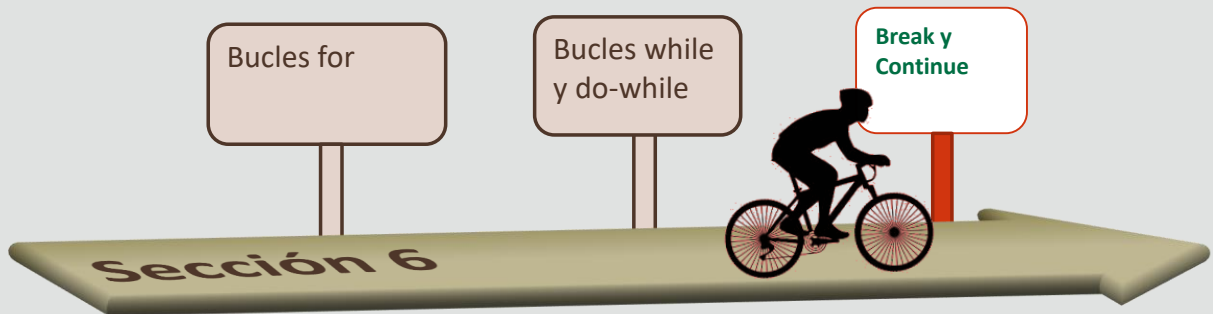
Objetivos

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



Temas

- **Uso de una sentencia break en un bucle**
- Uso de una sentencia continue en un bucle
- Escritura de comentarios en bucle

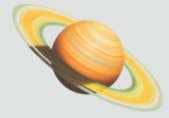


ORACLE
Academy

JFo 6-3
Uso de las sentencias break y continue

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

4



Misión a los anillos de Saturno

- Vamos a considerar otro escenario para esta misión
 - Cuando la nave está girando alrededor de Saturno y tomando instantáneas, el brazo robótico o la cámara se rompen
- ¿Cómo solucionaría este problema?
 - Si tuviera que escribir un programa Java, ¿qué construcción debe utilizar?
 - Vamos a ver si Java tiene una sentencia que le permita terminar un bucle inmediatamente

¿Cómo saldría de un bucle anticipadamente?

- Normalmente, la única manera de salir de un bucle es que la condición que se va a evaluar sea false
- Sin embargo, a menudo es conveniente terminar un bucle con antelación cuando se cumplen determinadas condiciones
- En estos casos, continuar con el bucle sería una pérdida de tiempo del procesador

¿Cómo saldría de un bucle anticipadamente?

- Puede utilizar dos sentencias de Java para terminar un bucle con antelación:
 - break
 - continue

Uso de break en un bucle

- Al ejecutar una sentencia break en un bucle, la sentencia de bucle termina de forma inmediata
- El programa seguirá ejecutándose con la sentencia posterior 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]

Uso de break en un bucle while: Ejemplo

• Salida: 0 1 2 3

- La ejecución del bucle termina cuando el contador de bucles es igual a 4

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

En el ejemplo del código, aunque el bucle se ha declarado que se ejecute 10 veces, la sentencia `break` sale del bucle después de únicamente 4 iteraciones. El último valor de `i` se muestra en la consola después de que el bucle termine.

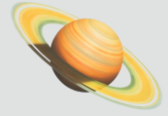
Uso de break en un bucle for

- Vamos a escribir 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 999, terminar el bucle independientemente del valor del contador de bucles y sin agregar a la suma

Uso de break en un bucle for: Ejemplo

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

Misión a los anillos de Saturno: Implantación de las condiciones



- Vamos a usar 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 minDistance = 50000000;  
    while (distTravelled >= minDistance) {  
        snap++; //click snap  
        if (camera == broken) {  
            break;  
        }  
        else {  
            rotate();  
        }  
    }  
}
```

En este ejemplo, la cámara está rota, las sentencias en el bucle `while` se terminan y el control pasa a la sentencia fuera del bucle `while`. Es decir, la nave continúa girando alrededor de Saturno.

Ejercicio 1



- Importe y abra el proyecto BreakContinueEx
- Examine ComputeSum.java
- Implante lo siguiente:
 - Aceptar 10 números del usuario
 - Calcular la suma de los números introducidos
 - Al introducir 0, el programa se debe cerrar y mostrar la suma de los números

Salida esperada:

Introduzca 10 números.

Introduzca 0 para salir.

1

2

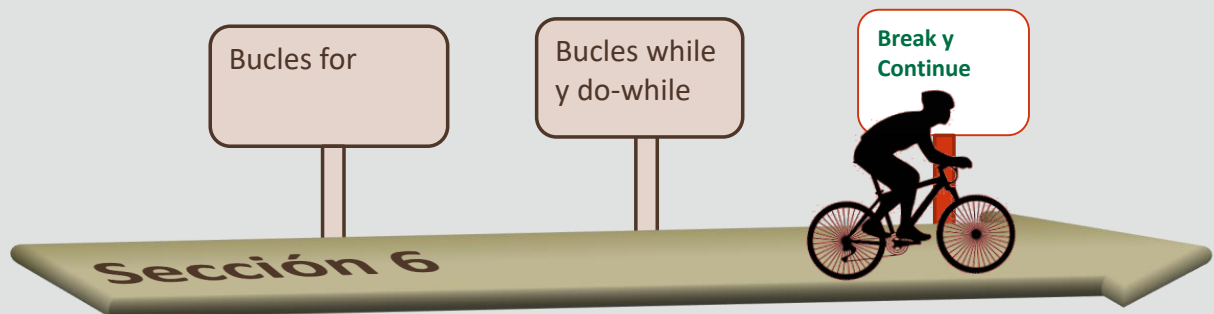
3

0

La suma de los números introducidos es 6.

Temas

- Uso de una sentencia break en un bucle
- **Uso de una sentencia continue en un bucle**
- Escritura de comentarios en bucle



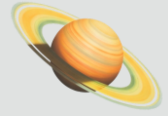
ORACLE
Academy

JFo 6-3
Uso de las sentencias break y continue

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

15

Misión a los anillos de Saturno: Otro escenario



- Vamos a considerar otro escenario para esta misión
 - Cuando la nave está girando alrededor de Saturno y tomando instantáneas de los anillos de Saturno...
 - Si la visibilidad es cero, no realice instantáneas
 - De lo contrario, continúe con las instantáneas
- ¿Cómo solucionaría este problema?
 - Si tuviera que escribir un programa Java, ¿qué construcción debe utilizar?
 - Vamos a ver si Java tiene una sentencia que le permita omitir la iteración actual del bucle

Uso de continue en un bucle

- A veces, puede que desee omitir la iteración actual en un bucle y no terminar el bucle en sí
- Puede utilizar una sentencia continue para omitir la iteración actual en un bucle:
 - Es decir, el resto del cuerpo del bucle se omite al final del bucle. Sin embargo, no finaliza el bucle
 - Cuando el programa alcanza el final del bucle, el programa volverá a comprobar 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 se transfiere a la condición de bucle

These Estas sentencias se omitirán en la iteración actual are skipped in the current iteration

Uso de continue en un bucle for

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

```
    statement1;
```

```
    statement2;
```

```
    continue;
```

```
    statement3;
```

```
    statement4;
```

```
} //end for
```

El control se transfiere a la condición de bucle

Estas sentencias se omitirán en la iteración actual

Uso de continue en un bucle for

- Salida: `0 1 2 3 5 6 7 8 9`

- La salida no incluye el 4 Debido a la sentencia continue, la ejecución del bucle se omite cuando el contador de bucles 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 con el bucle while y la sentencia continue
- El programa debe...
 - Calcular la suma de números entre 1 y 99 mediante el bucle while
 - Si el número es un múltiplo de 10, se debe omitir la iteración actual y el número no se debe agregar a la suma
 - Mostrar la suma en la consola

Cálculo de la suma de números

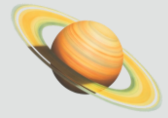
```
public static void main(String[] args) {  
    int counter = 0;  
    int sum = 0;  
    while (counter < 100) {  
        counter++;  
        if (counter % 10 == 0) {  
            continue;  
        }  
        else {  
            sum += counter;  
        }  
    }  
    System.out.println("Sum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

¿Es un múltiplo de 10? En caso afirmativo, omitir la iteración actual

La salida de este ejemplo es:

Suma de 1 - 99:4500

Misión a los anillos de Saturno: Implantación de las condiciones



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

```
public static void main(String[] args) {  
    long distTravelled = 0;  
    long minDistance=500000000;  
    while (distTravelled >= minDistance) {  
        if (visibility == 0) {  
            continue;  
        }  
        else {  
            snap++;  
        }  
    }  
}
```

En este ejemplo, la visibilidad es cero, las instantáneas de los anillos de Saturno no se toman, el control se transfiere a la condición y el programa continúa ejecutando la siguiente iteración. Estas acciones se deben a que ha utilizado la sentencia `continue` statement.



Ejercicio 2

- Importe y abra el proyecto `BreakContinueEx`
- Examine `CountChar.java`
 - El programa se utiliza para contar el número de incidencias del carácter "w" en la cadena
 - Modifique el programa para...
 - Resolver el error de sintaxis
 - Imprimir el recuento del carácter "w"
 - Salida esperada:
 - Número de w: 3

Ejercicio 3



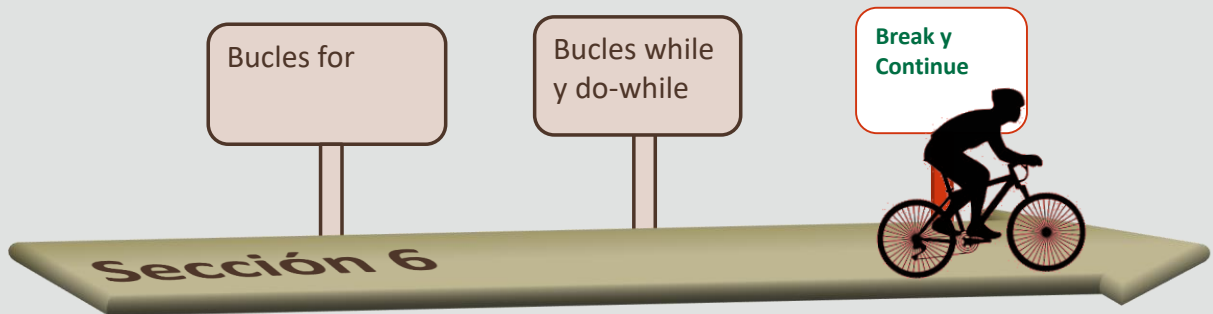
- Importe y abra el proyecto `BreakContinueEx`
- Examine `BreakContinue.java`
- Modifique el programa mediante sentencias `break` y `continue`...
 - Si el número es par, el número no se debe imprimir
 - La ejecución del bucle se debe parar cuando el valor del contador de bucles es 7

Salida esperada:

```
El número es 1
El número es 3
El número es 5
El número es 7
```

Temas

- Uso de una sentencia break en un bucle
- Uso de una sentencia continue en un bucle
- **Escritura de comentarios en bucle**



ORACLE
Academy

JFo 6-3
Uso de las sentencias break y continue

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

26

Escritura de comentarios en bucle

- Es una práctica recomendada para agregar comentarios adecuados a los bucles
- De lo contrario...
 - El código tiende a resultar confuso
 - No podrá entender la lógica muy fácilmente
- Ayuda a entender...
 - Las variables de bucle utilizadas y su finalidad
 - La lógica del bucle
 - El número de iteraciones
 - La ejecución de las sentencias en el bucle en función de la condición o criterio o ambos

Escritura de comentarios en bucle: Ejemplo

```
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("That number was odd");  
    } //end for  
} //end method main
```

Ejercicio 4



- Importe y abra el proyecto `BreakContinueEx`
- Examine `Divisors.java`
- El programa busca todos los divisores de un número

Ejercicio 4



- Modifique el programa para que incluya comentarios del bucle acerca de...
 - Las variables de 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:
 - Usar una sentencia break para salir de un bucle
 - Usar una sentencia continue para omitir una parte de un bucle
 - Explicar la necesidad de comentarios en bucle



