



Java 7 SE Fundamentals





Sección 7

Uso de ciclos



Crear ciclos while y ciclos while anidados.

Alcance de las variables.

Crear ciclos do-while

Crear ciclos for

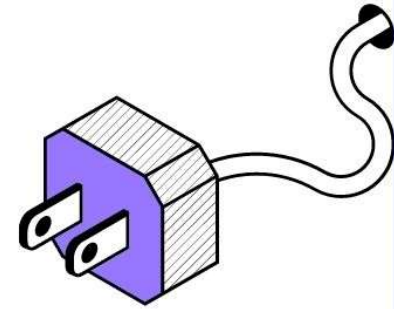
Utilizar un ArrayList con el ciclo for.

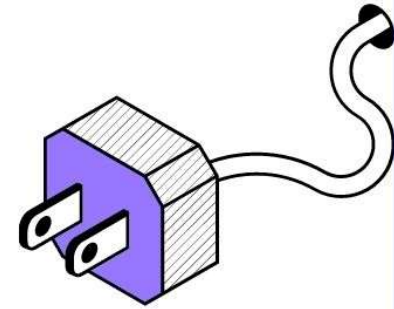


Ciclo while

Características

- Un bucle o ciclo while ejecuta declaraciones repetidamente mientras una condición es verdadera.
- La sintaxis del ciclo while es:
while (condición){
 //cuerpo del ciclo
 Instrucción (es);
}





Ciclos while anidados

Características

- Es posible definir un ciclo **while** dentro de otro.
- La sintaxis de ciclos while anidados es:

```
while ( condición1 ){  
    while ( condición2 ){  
        //cuerpo del ciclo  
        Instrucción (es);  
    }  
}
```





Alcance de las variables.

Como funciona

- El alcance de una variable es la parte del programa donde se puede hacer referencia a la variable.
- Una variable definida dentro de un método se denomina variable local.
- El alcance de una variable local comienza desde su declaración y continúa hasta el final del bloque que contiene la variable.
- Una variable local debe declararse y asignarle un valor antes de que pueda ser utilizada.





Alcance de las variables.

```
public static void method1() {  
    .  
    .  
    for (int i = 1; i < 10; i++) {  
        .  
        .  
        int j;  
        .  
        .  
        .  
    }  
}
```

Alcance de *i* →

Alcance de *j* →



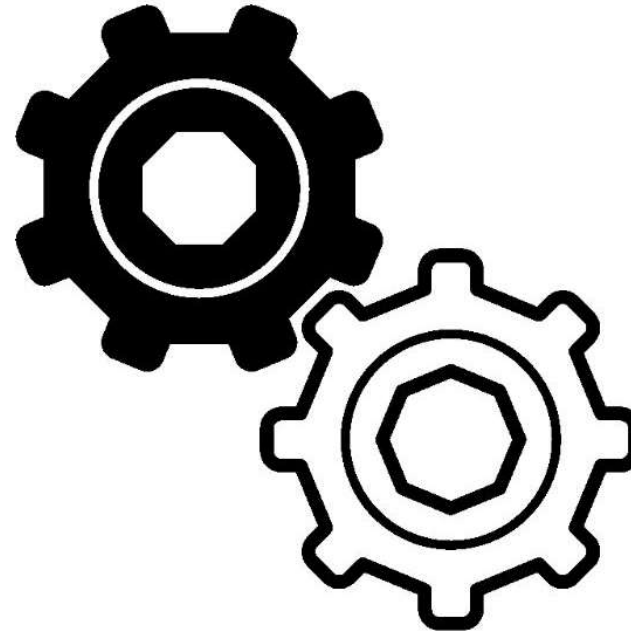
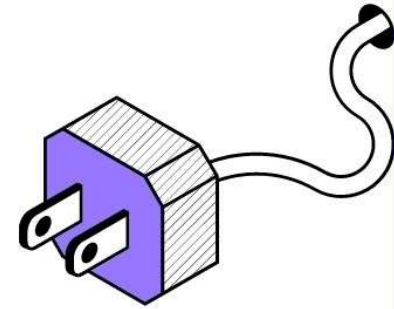
Ciclo do while

Características

- Un ciclo do-while es lo mismo que un ciclo while excepto que primero ejecuta el cuerpo del ciclo y luego verifica la condición de continuación del ciclo.

- La sintaxis del ciclo do while es:

```
do{  
    //cuerpo del ciclo  
    Instrucción (es);  
}while ( condición );
```



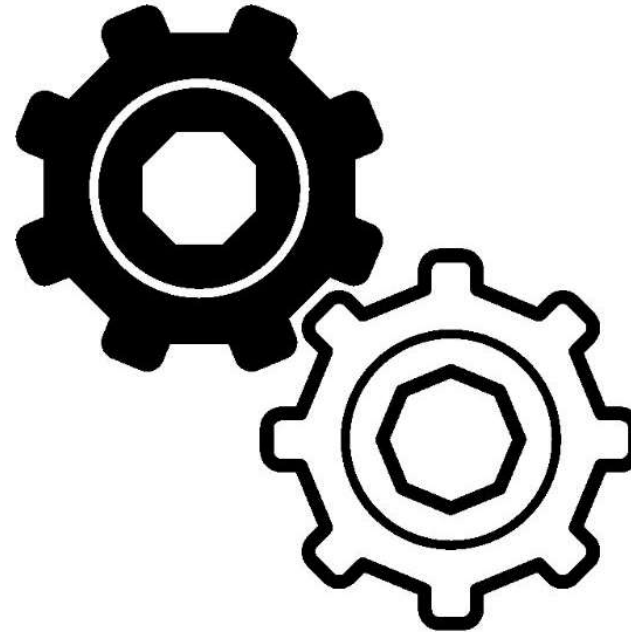
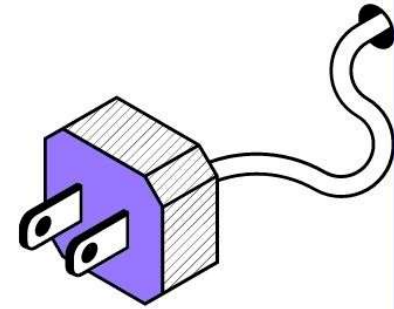


Ciclo for

Características

- Un ciclo for también tiene una condición, pero esta condición, se sabe desde un principio cuando dejara de ser verdadera.
- La sintaxis del ciclo for es:

```
for (int i = valorInicial; i < valorFinal; i++) {  
    // Loop body ...  
}
```





Ejemplo de uso del ciclo for con un ArrayList.

Suponiendo que tenemos definido un ArrayList que guardara elementos de tipo entero y que agregamos tres elementos, es posible procesar o manipular un objeto ArrayList con ciclos, tal y como se hizo con un arreglo convencional.

```
ArrayList<Integer> arrlist = new ArrayList<Integer>();
```

```
arrlist.add(14);  
arrlist.add(7);  
arrlist.add(39);
```

```
for (int i= 0; i< arrlist.size(); i++)  
{  
    System.out.println(arrlist.get(i));  
}
```

```
int j = 0;  
while (arrlist.size() > j) {  
    System.out.println(arrlist.get(j));  
    j++;  
}
```