4.C. Estructuras de repetición.

✔ Hecho

1. Estructuras de repetición.

1.2. Estructura for/in.

Junto a la estructura `for, for/in` también se considera un bucle controlado por contador. Este bucle es una mejora incorporada en la versión 5.0. de Java.

```
1
2  public class repetitiva_for_in {
3      public static void main(String[] args) {
4          // Declaración e inicialización de variables
5          String[] semana = {"Lunes","Martes","Miércoles","Jueves","Viernes","Sábado","Domingo"};
6
7          //Salida de información
8
9          //Utilizamos ahora el bucle for/in
10         for (String dia: semana){
11         /* La cabecera del bucle incorpora la declaración de la variable dia
12          * a modo de contenedor temporal de cada uno de los elementos que forman
13          * el array semana.
14          * En cada una de las iteraciones del bucle, se irá cargando en la variable
```



```
20         }
21     }
22
23 }
24
```

Este tipo de bucles permite realizar recorridos sobre `arrays` y colecciones de objetos. Los `arrays` son colecciones de variables que tienen el mismo tipo y se referencian por un nombre común. Así mismo, las colecciones de objetos son objetos que se dice son iterables, o que se puede iterar sobre ellos.

Este bucle es nombrado también como bucle `for` mejorado, o bucle `foreach`. En otros lenguajes de programación existen bucles muy parecidos a este.

La sintaxis es la siguiente:

```
for (declaración: expresión) {
sentencia1;
…
sentenciaN;
}
```

- Donde `expresión` es un array o una colección de objetos.
- Donde `declaración` es la declaración de una variable cuyo tipo sea compatible con expresión. Normalmente, será el tipo y el nombre de la variable a declarar.

El funcionamiento consiste en que para cada elemento de la expresión, guarda el elemento en la variable declarada y realiza las instrucciones contenidas en el bucle. Después, en cada una de las iteraciones del bucle tendremos en la variable declarada el elemento actual de la expresión. Por tanto, para el caso de los `arrays` y de las colecciones de objetos, se recorrerá desde el primer elemento que los forma hasta elúltimo.

Observa el contenido del código representado en la siguiente imagen, puedes apreciar cómo se construye un bucle de este tipo y su utilización sobre un `array.`

Los bucles `for/in` permitirán al programador despreocuparse del número de veces que se ha de iterar, pero no sabremos en qué iteración nos encontramos salvo que se añada artificialmente alguna variable contadora que nos pueda ofrecer esta información.

◄ 4.B. Estructuras de selección.

Ir a...

4.D. Estructuras de salto. ►