**Práctica 1 Método científico**

**Ejercicio 1. Dado el siguiente Código de ordenación por el método de la burbuja:**

**public class BubbleSort {**

**/\*\***

**\* This method the BubbleSort method. Returns**

**\* the array in increasing order.**

**\***

**\* @param array The array to be sorted**

**\* @param size The count of total number of elements in array**

**\*\*/**

**public static <T extends Comparable<T>> void BS(T array[], int size) {**

**boolean swap;**

**int last = size - 1;**

**do**

**{**

**swap = false;**

**for (int count = 0; count < last-1; count++)**

**{**

**int comp = array[count].compareTo(array[count + 1]);**

**if (comp > 0)**

**{**

**T temp = array[count];**

**array[count] = array[count + 1];**

**array[count + 1] = temp;**

**swap = true;**

**}**

**}**

**} while (swap);**

**}**

**}**

**Al ejecutar el código con los valores de entrada obtenemos estos resultados:**

**[1 0 0 0 2 0 ] --> [0 0 0 1 2 0 ]**

**[8 7 6 5 4 9 ] --> [4 5 6 7 8 9 ]**

**Rellena la siguiente plantilla aplicando el método científico para depuración:**

**Solución**

|  |  |
| --- | --- |
| **Hipótesis** | Todo funciona correctamente |
| **Predicción** | Sale el resultado correcto |
| **Experimento** | Ejecución del código con valores de prueba [1 0 0 0 2 0 ] y [8 7 6 5 4 9 ] |
| **Observación** | El código devuelve [0 0 0 1 2 0 ] y [4 5 6 7 8 9 ], el código da resultados incorrectos |
| **Conclusión** | Hipótesis incorrecta |

|  |  |
| --- | --- |
| **Hipótesis** | Variable Last de la función BS es incorrecta |
| **Predicción** | Al modificar el valor de la variable eliminando el -1, el resultado será correcto |
| **Experimento** | Ejecutar el código con los cambios realizados |
| **Observación** | El resultado aún no es correcto, aunque ahora debugueando sabemos que el valor de Last es correcto |
| **Conclusión** | Hipótesis correcta |

|  |  |
| --- | --- |
| **Hipótesis** | El problema está en el while |
| **Predicción** | Si cambiamos el while por un for, quitando la variable swap funcionará |
| **Experimento** | Ejecutamos el código con las pruebas del enunciado. |
| **Observación** | El resultado sigue dando fallo, pero observamos que el problema está en el <T extends Comparable<T>> |
| **Conclusión** | Hipótesis correcta. |

|  |  |
| --- | --- |
| **Hipótesis** | Si no hacemos uso de <T extends Comparable<T>> el resultado será correcto. |
| **Predicción** | Al sustituir el párametro de T array por int array y cambiar las condiciones para ajustarlas al nuevo tipo de parámetro dará un resultado correcto. |
| **Experimento** | Ejecutamos el código con los cambios realizados. |
| **Observación** | El código funciona con las pruebas [1 0 0 0 2 0 ] --> [0 0 0 0 1 2] y [8 7 6 5 4 9 ] --> [4 5 6 7 8 9 ]. |
| **Conclusión** | Hipótesis correcta y ejercicio resuelto. |