

CONSIDERACIONES GENERALES PARA ABORDAR LOS EJERCICIOS DE PROGRAMACIÓN

Vamos a intentar “clarificar” algunos aspectos a la hora de afrontar la realización de los ejercicios propuestos en este módulo.

Lo primero que tenemos que tener claro son las FASES que deberíamos seguir, que son de aplicación tanto a estos ejercicios como a un “programa” que nos encargaran (si ya fuéramos programadores profesionales):

1. Entender qué es lo que nos están pidiendo.
2. Pensar en la estrategia de solución que vamos a seguir: lo que sería elaborar la receta de cocina:
 - a) Analizar qué datos va a necesitar el programa, y qué datos va a producir. Tendremos que sopesar si el programa tiene que “acordarse” de los datos de entrada como de los datos producidos, o si una vez producidos directamente se vuelcan a la consola, un fichero, etc. Esto nos servirá para determinar qué variables y de qué tipo de datos nos harán falta.

Por decirlo con otras palabras, serían los ingredientes y el plato resultante

- b) Analizar qué pasos va a seguir el programa. Esto incluirá el uso de las estructuras de control condicional y bucles que consideremos necesarias. También las variables auxiliares, necesarias para hacer cálculos intermedios o para controlar la ejecución de algunos pasos, así como su repetición. Para pensar los pasos que hay que seguir, habrá que inspirarse en la experiencia previa de programas similares que hayamos realizado, o habrá que ponerse “a dibujar” la situación, es decir, hacerse un esquema (quizá en papel) en el que veamos los pasos que hay que ir dando, y cómo a cada paso “evoluciona” el contenido de las variables.

Estos pasos pueden incluir “referencias” a los pasos dados en otros programas, que pudieran ser reutilizados en el que ahora nos ocupa, como parte de la solución.

Nos estaríamos refiriendo a las “instrucciones” que se han de seguir al ejecutar la receta.

3. Hasta este momento, no empezamos a “codificar”, es decir, escribir en un lenguaje de programación determinado la reserva de variables y los pasos que debe seguir nuestro programa. Si los “pasos” que debe seguir el programa no están claros, será imposible realizar la codificación. Si los pasos están claros, es posible que aparezcan problemas para codificar el programa en el lenguaje de programación que estemos usando, y esos problemas se resuelven asegurándose de que estamos usando bien la sintaxis de dicho lenguaje.

Es decir, es como si la receta la tuviéramos que escribir en ruso, y ahora el problema está en la sintaxis en ruso. A estas alturas, no deberíamos tener problema en escribir la receta dibujando los pasos, o explicándolos en otro idioma, porque sabemos lo que queremos hacer; solo faltaría escribirlo bien en la lengua que fuese.

Por tanto, el “núcleo” de la cuestión a la hora de diseñar cómo vamos a hacer un programa (que no tiene nada que ver con el lenguaje de programación utilizado) está en “descomponer” la solución en los sucesivos pasos que debemos ir dando. Esos “pasos” pueden ser operaciones cortas que cualquier lenguaje de programación soporta; nosotros debemos establecer el orden y el “sentido” de ese orden en esos pasos. Ejemplos de este tipo de operaciones serían: operaciones aritméticas, concatenación de strings, comparaciones de valores, asignación de valores obtenidos en cálculos, etc.

¿Cómo dan los pasos los ordenadores? Ejecutan un paso y, a continuación, el siguiente. Salvo que utilicemos estructuras de control que permitan ejecutar o no determinados pasos en función de que se cumpla una condición (estructuras de control condicional; permiten que nos “saltemos” pasos), o estructuras de control que permitan “repetir” instrucciones, un número de veces a priori conocido o no (estructuras de control repetitivas).

Habrá que imaginar cómo queremos que se ejecuten los pasos, “llevando de la mano” al ordenador en todos los pasos que de. Si no somos capaces de imaginarlo al principio, habrá que “ponerse en su pellejo”, y mover por él los datos en la memoria, como si las variables fueran cubos que contienen en lugar de líquido el valor que corresponda. Quizá un dibujo pueda servir. Quizá un diagramador como PSEINT sea útil y ayude a visualizar la solución, en lo que respecta a los pasos a seguir.

Habrá situaciones en las que será necesario “saber” sobre cómo se hacen determinados cálculos. Si no se dispone de ese conocimiento, habrá que buscarlo y habrá que adquirirlo. Y una vez supiéramos cómo hacer lo que se nos pide “a mano”, podremos pensar en cómo automatizarlo para que lo haga un ordenador.

Otras veces habrá que “descubrir” el camino a seguir, pero siempre deberíamos plantearlo como pasos que hay que dar, sin pensar en la sintaxis de ningún lenguaje de programación.

Por ejemplo: hacer un programa que almacene 5 datos en un array de enteros y muestre por pantalla la lista de los datos en orden inverso y la suma de sus valores.

1. Vamos a necesitar un array de 5 posiciones enteras y una variable entera en la que guardemos la suma de sus valores.
2. Los pasos a seguir podrían ser:
 - a) Rellenar los valores en el array
 - b) Recorrer el array para obtener la suma de sus valores
 - c) Sacar por pantalla los valores en orden inverso
 - d) Sacar por pantalla la suma de sus valores

Podemos y debemos “afinar” los pasos que hemos descrito:

- a) Rellenar los valores en el array:

Bucle que de tantas vueltas como posiciones tiene el array

En cada pasada, acceder a una posición distinta y almacenar ahí el valor capturado desde el teclado

b) Recorrer el array para obtener la suma de sus valores

Inicializar la suma a cero

Bucle que de tantas vueltas como posiciones tiene el array

En cada pasada, acceder a una posición distinta y sumar a lo que ya hay en suma el valor de esta posición

c) Sacar por pantalla los valores en orden inverso

Bucle que de tantas vueltas como posiciones tiene el array

En cada pasada, acceder a una posición distinta, yendo en el orden inverso al que se han rellenado, y sacar el valor correspondiente a la posición por pantalla.

d) Sacar por pantalla la suma de sus valores

Con este nivel de detalle, ya no debería costarnos mucho llegar a escribir en un lenguaje de programación estos “pasos”:

```
int [] valores = new int[5];
int suma;

for (int i=0; i<5; i++) {
    System.out.print("Indique valor: ");
    int auxiliar = Integer.parseInt(System.console().readLine());
    valores[i] = auxiliar;
}

suma = 0;
for (int i=0; i<5; i++) {
    suma = suma + valores[i];
}

for (int i=0; i<5; i++) {
    System.out.print(valores[i]+" ", "");
}
System.out.println();
System.out.println("La suma de los valores es: "+suma);
```