Software

Desarrollo software, depuración de código, diagrama de clases

Jose Manuel Soldado

Introducción

Se ha contratado a un informático caótico para gestionar un código java para gestionar una biblioteca. De momento solo tiene un trozo de código en el que le han pedido que haga un programa que le pida al usuario 10 números sin decimales por teclado y los muestre de menor a mayor por pantalla. También de esos números tiene que sacar cuales son pares y cuales son impares. Así que vamos a realizar una revisión a su código y vamos a ver porque no funciona. Identificaremos cuales son los errores, porque no se puede ejecutar el código.

Así como método de introducción se ha detectado que utiliza demasiados scanner que al fin y al cabo, con que se use uno para la entrada de datos por teclado sería mas que suficiente. También el intenta ordenar los números MANUALMENTE lo cual es ineficiente e incluso podría a llegar a provocar errores masivos en el código, y llegaría a un punto de a la vista del que revise el código borrarlo entero y realizarlo desde cero. También se usa una operación % 2 para comprobar los números pares, hasta ahí bien, pero luego lo almacena en una variable float que no se exactamente el porque. Por ultimo la suma de números también lo hace de manera manual, lo cual en escribir eso habrá tardado una eternidad y además de eso genera un código espagueti muy difícil de leer.

A continuación vamos a realizar un estudio mas detallado de los errores y las posibles soluciones que se le ha dado a su código.

1. Arreglo de errores

Múltiples objetos Scanner innecesarios:

- Estás creando DEMASIADOS Scanner (por ejemplo, Scanner sn1 = new Scanner(System.in);, Scanner sn2 = new Scanner(System.in);, etc.). Solo necesitas un solo Scanner para leer los datos desde la entrada estándar. Es decir, un solo objeto Scanner.

Ordenación de los números:

- Está intentando ordenar los números utilizando un conjunto de if y comparaciones manuales. Esto es innecesariamente complicado.
- He usado un método Arrays.sort() de la clase Arrays es una forma mucho más sencilla y eficiente de ordenar los números.

Control de pares e impares:

- Estás comprobando si cada número es par usando float z = a % 2;, lo cual no tiene sentido, porque a % 2 te da un resultado entero (0 o 1) y no necesitas convertirlo a float.
- Además, estás imprimiendo "PAR" para cada número par, pero no llevas un conteo de los pares e impares como se requería.

Suma de números:

- Está imprimiendo la suma de los números en la línea System.out.println(a+b+c+d+e+f+g+h+i+j);. Aunque esto funciona, no es una forma elegante ni escalable. Si tuviera más o menos números, tendrías que modificar esta línea manualmente. Mejor usar un bucle y una variable que almacene todos los números y los clasifique en pares o impares (pares++ ó impares++) para sumar los números de manera más eficiente.

Mal manejo de la variable x:

- Usa la variable x para controlar un bucle while, pero no está bien implementado. El código de ordenación debería estar dentro de un solo bloque de código, no como múltiples if secuenciales fuera de un bucle.

Impresión de los resultados de manera incorrecta:

- El programa solo imprime "PAR" cuando encuentra un número par, pero no lo hace de manera eficiente ni clara. También falta la lógica para mostrar los números impares o el número total de pares e impares.

2. Solución

He implementado una solución basándome en POO, ya que a mi parecer es la opción más cómoda para realizar cualquier tarea. Las ventajas que te da eso es un orden en tu código que cada vez que abras el proyecto o haya que hacer algún mantenimiento, al leerlo sabes como ejecuta el programa sin tener que ejecutarlo manualmente.

2.1. CLASES

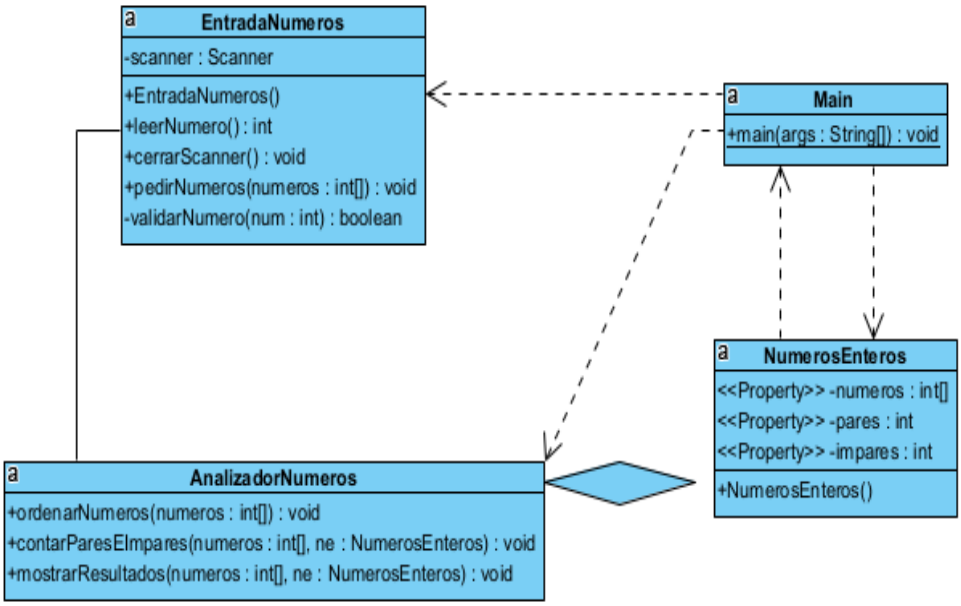
Class AnalizadorNumeros -> Clase que ordena los números del array y posteriormente recorre ese array e divide los números entre 2 y los almacena en variables cuyo nombre son impares e pares

Class EntradaNumeros -> Creación del objeto Scanner para la entrada mediante teclado de números e implementamos un bucle do while para la entrada de numeros, mientras que los números no superen el tamaño del array el bucle no parará

Class NumerosEnteros -> Se hace la plantilla de números enteros donde se almacena un array de 10 elementos, una variable pares, una variable impares y un objeto scanner

Class Main -> La ejecución del programa

3. DIAGRAMA DE CLASES



4. BIBLIOGRAFIA

ARRAY.SHOT-> https://www.datacamp.com/es/doc/java/sort

Scanner.close error "Resource leak: input never close" -> https://es.stackoverflow.com/questions/70219/cuando-se-utiliza-in-close-de-la-clase-scanner

SCANNER.hasNext.int https://es.stackoverflow.com/questions/465283/java-clase-scanner

@PARAM

https://stackoverflow.com/questions/16138217/how-exactly-does-param-work-java

visual paradimg

https://www.visual-paradigm.com/