**Actividad**

Elementos de un programa

**Objetivos**

* Utilizar estructuras de control de flujo y de errores
* Crear algoritmos simples
* Recorrer estructuras array y acceder a sus valores.
* Depurar y comentar los programas.

|  |
| --- |
| **¿Cómo lo hago?** |
| * Rellena los datos que se piden en la tabla “Antes de empezar”. * Haz uso de fuentes comunes como Arial, Calibri, Times New Roman etc. * Utiliza el color negro para desarrollar tus respuestas y usa otros colores para destacar contenidos o palabras que creas necesario resaltar. * Entrega un zip que contenga todos los archivos. java que has creado. Para poder aprobar un ejercicio, éste debe poder ejecutarse sin errores. * Recuerda nombrar el archivo zip siguiendo estas indicaciones: * Ciclo\_Módulo o crédito\_Tema\_ACT\_número actividad\_Nombre y apellido * Ejemplo: AF\_M01\_T01\_ACT\_01\_Maria Garcia |

|  |  |
| --- | --- |
| **Antes de empezar…** | |
| Nombre | Jon |
| Apellidos | Sanchez Eguia |
| Módulo/Crédito | DAW\_M03A\_Programación |
| UF (solo ciclos LOE) | 1 |
| Título de la actividad | Actividad 2 |

***Se debe entregar un zip que contenga todos los archivos. java que has creado. Para poder aprobar un ejercicio, éste debe poder ejecutarse sin errores. Crea los archivos .java dentro de una carpeta de nombre actividad02***

* **Ejercicio01.java: Crea un programa que cree un array de 5 posiciones para controlar la cantidad de litros de refresco en 5 recipientes inicialmente a 0.**
* Inicalmente el programa debe mostrar el siguiente mensaje al usuario:
* Introduzca la operación a realizar del siguiente menú de opciones:
* 1- Añadir refresco a un recipiente.   
  2-Quitar refresco de un recipiente.   
  3- Calcular el total de refresco.
* 0-Finalizar
* Cada vez que se realice la operación 1, 2, 3 se ha de mostrar por pantalla el estado de los recipientes.
* Controla mediante un DO WHILE que una vez realizada la operación seleccionada se vuelva a mostrar el menú de operaciones excepto si el usuario ha introducido un 0. Si se introduce un valor incorrecto (que no sea un número) o menor a 0, indica que se ha introducido un valor incorrecto y vuelve a mostrar el menú.
* Si el usuario introduce un 1, el programa ha de pedir el número de recipiente y la cantidad de refresco con decimales y añadir esa cantidad al recipiente.
* Si la cantidad de refresco es inferior a 0 o el número de recipiente es incorrecto hay que volver a pedir el valor.
* Si el usuario introduce un 2, el programa ha de pedir el número de recipiente y la cantidad de refresco con decimales y quitar esa cantidad del recipiente.
* Si la cantidad de refresco restante o indicado es inferior a 0 o el número de recipiente es incorrecto hay que volver a pedir el valor.
* Si el usuario introduce un 3, el programa ha de mostrar la cantidad total de litros que hay en todos los recipientes
* Si el usuario introduce un 0, se ha de mostrar un mensaje de despedida y finalizar el programa.

Ejemplo de ejecución del programa:









**>Soluciones:**

**Menu**



**Opcion 1**



**Opcion 2 (con funcionalidad de no volumen negativo)**



**Opcion 3**



**Opcion 0**



* **Ejercicio02.java:** escribe un programa que almacene en un array bidimensional (tipo String) el nombre de tres asignaturas y sus cuatro puntuaciones (puedes ver un esquema después del enunciado).
* Inicialmente se mostrará por consola todos los valores almacenados en el array y luego se pedirá al usuario si quiere:
* 1-Mostrar el nombre de todas las asignaturas.
* 2-Modificar el nombre de una asignatura.
* 3-Modificar la puntuación de una asignatura.
* 4-Mostrar la puntuación menor de cada una de las asignaturas.
* 0-Salir
* El programa no ha de finalizar hasta que el usuario seleccione la opción 0.
* Programa cada una de las opciones del menú.
* Si el usuario indica una posición de array incorrecta, o introduce un valor numérico incorrecto vuelve a pedir el valor.

ESTRUCTURA Y VALORES DEL ARRAY BIDIMENSIONAL DE STRINGS

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | “M01” | | “8” | | “5” | | “10” | | |  | | --- | | “M02” | | “3” | | “8” | | “9” | | |  | | --- | | “M03” | | “9” | | “7” | | “6” | | |  | | --- | | “M04” | | “10” | | “4” | | “9” | |

Ejemplo de ejecución del programa:









**CAPTURAS:**











**En la última captua no sale actualizado el valor 2 de M02 debido a que he lanzado el programa otra vez porque olvidé mostrar esa funcionalidad, en caso de haberlo hecho bien se vería actualizado**

**Las capturas son en Eclipse debido a que estaba testeando el IDE por la asignatura "Entornos de desarrollo"**

**En el propio codigo dejo comentada fuera la funcionalidad 5, que es la que busca el menor valor de cada asignatura, en caso de volverla a poner y amentar el valor**

**En los códigos he dejado mis errores fallidos para enseñar mi proceso de aprendizaje, espero que no sea mucha molestia.**