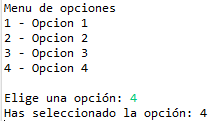
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Curso / *Kurtsoa*** | **Fecha / *Data*** | **Eval. /*Ebal.*** | **Calificación**  ***Kalifikazioa*** |
| 2024/25 | 15-11-2023 | 1º EVAL |  |
| **Módulo / Modulua** | | | | **Cód. exam.**  ***Azter. kod.*** |
| Programación (1DAM) | | | | PROG |

Nombre: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

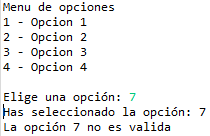
**EJERCICIOS [10pts]**

Lee **atentamente** el enunciado del examen. El examen consta de **10 preguntas**, cada una de las cuales está valorada en **1 punto**. La dificultad de las preguntas se va incrementando con cada sucesiva pregunta. **No es posible** contestar (ni se van a corregir) respuestas salteadas: se debe de empezar por la primera, pasar por la segunda, etc. hasta llegar a la última. Si el programa hace de más o de menos para una pregunta, esta se puntuará como **errónea**.

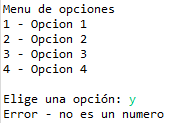
Para responder a este examen, abre tu Eclipse y crea un paquete llamado **examenEvalUno**. Incluye dentro de este paquete cada una de las clases que te solicitemos en las preguntas. Para entregar el examen, comprime la carpeta **examenEvalUno** con el código fuente en un fichero .zip. Dicho fichero debe de llamarse de la siguiente forma: **tuNombreYApellidosEvalUno**.zip. Cualquier clase entregada que NO compile o esté vacía cuenta como errónea. Lo mismo sucede si el fichero comprimido no contiene el código fuente (ficheros .java) solicitados en el examen.

**1.-** Crea una clase llamada **Ejercicio1**. Esta clase debe de escribir el siguiente menú por pantalla, preguntar al usuario por la opción numérica pertinente, y después escribir la opción seleccionada por pantalla.

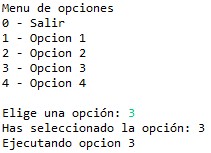
Resultado de la ejecución:

**2.-** Crea una clase llamada **Ejercicio2**. Copia el contenido de **Ejercicio1** en **Ejercicio2.** A demás de lo anterior, esta clase debe de controlar que el usuario solo pueda introducir valores del 1 al 4. Si no los recibe, debe de volver a solicitarlos hasta que meta un valor del 1 al 4.

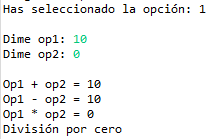
Resultado de la ejecución:

**3.-** Crea una clase llamada **Ejercicio3**. Copia el contenido de **Ejercicio2** en **Ejercicio3.** A demás de lo anterior, esta clase debe de controlar mediante una Excepción que el usuario no introduzca un valor que no sea un número entero.

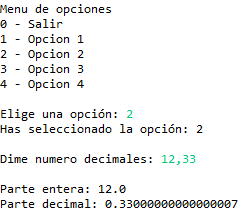
Resultado de la ejecución:

**4.-** Crea una clase llamada **Ejercicio4**. Copia el contenido de **Ejercicio3** en **Ejercicio4.** A demás de lo anterior, esta clase debe de llamar a 4 métodos llamados metodo1 (), método2 (), metodo3 () y metodo4 () según la opción seleccionada. Cada método escribirá un mensaje “Ejecutando opción” con su número de opción cada vez que sea invocado. Modifica el menú para añadir la opción 0 – Salir. El programa no debe terminar mientras no se escoja la opción 0.

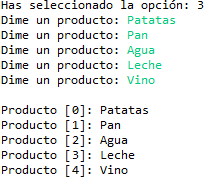
Resultado de la ejecución:

**5.-** Crea una clase llamada **Ejercicio5**. Copia el contenido de **Ejercicio4** en **Ejercicio5.** A demás de lo anterior, esta clase debe, en el metodo1 (), solicitar dos números enteros op1 y op2 al usuario, y calcular la suma, resta, multiplicación y división de ambos números. Muestra los resultados por pantalla. Si la división fuese a ser por cero, en lugar de eso, debe de mostrarse un mensaje “División por cero”.

Resultado de la ejecución:

**6.-** Crea una clase llamada **Ejercicio6**. Copia el contenido de **Ejercicio5** en **Ejercicio6.** A demás de lo anterior, esta clase debe, en el metodo2 (), solicitar un número con decimales y mostrar por separado la parte entera y la parte decimal (resto).

Resultado de la ejecución:

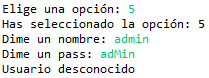
**7.-** Crea una clase llamada **Ejercicio7**. Copia el contenido de **Ejercicio6** en **Ejercicio7.** A demás de lo anterior, esta clase debe, en el metodo3 (), solicitar 5 productos (Strings) y almacenarlos en una estructura de datos (Array). Después, recorrer la estructura de datos imprimiendo por pantalla todos sus elementos guardados.

Resultado de la ejecución:

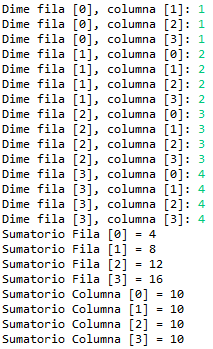
**8.-** Crea una clase llamada **Ejercicio8**. Copia el contenido de **Ejercicio7** en **Ejercicio8.** A demás de lo anterior, en esta clase modifica el metodo4 () para que reciba por parámetros un precio (double) y un IVA (int). En base a este precio, calcula el precio final con el IVA, y devuélvelo al main. Muestra el resultado en el main ().

**Nota**: no hace falta coger los parámetros por teclado.

Resultado de la ejecución:

**9.-** Crea una clase llamada **Ejercicio9**. Copia el contenido de **Ejercicio8** en **Ejercicio9.** A demás de lo anterior, modifica esta clase para añadir una opción de menú 5. Esta opción debe invocar al metodo5 (). El programa debe de funcionar correctamente. El método debe de pedir un String al usuario, que será su nombre. A continuación, deberá pedirle otro String que será su password. Si el primer String es igual al String ‘admin’ ignorando mayúsculas o minúsculas, y el segundo es exactamente igual a ‘admin’; entonces se mostrará el mensaje “Bienvenido!”. En caso contrario, se mostrará el mensaje “Usuario desconocido”

Resultado de la ejecución:

**10.-** Crea una clase llamada **Ejercicio10**. Copia el contenido de **Ejercicio9** en **Ejercicio10.** A demás de lo anterior, modifica esta clase para añadir una opción de menú 6. Esta opción debe invocar al metodo6 (). El programa debe de funcionar correctamente. El método debe de contener un matriz 4 x 4 de números enteros y solicita al usuario que la rellene. A continuación, muestra la suma de todas las columnas y cada una de las filas.

Resultado de la ejecución:

**Corrección**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
|  |  |  |  |  |  |  |  |  |  |

**Por cada una correcta +1 PTO. Si no funciona, +0 PTS**