
EJERCICIOS PROGRAMACION III

Práctica SEMANA 4

CONTENIDOS: Más POO, JAR con argumentos, Tipos primitivos y de referencia, Arrays en Java y uso de static

Ejercicios obligatorios:

1. Debes crear un proyecto simple en Java que reciba como argumentos 2 argumentos que deben de ser numéricos. El mostrará por pantalla la suma de ambos. En caso de no introducir el número correcto de parámetros o de ser incorrectos se indicará al usuario con un mensaje de error informativo. Debes realizar esto empleando el IDE y por otro lado, de forma manual con el binario java en la CLI.
2. Programa que recibe por línea de órdenes como parámetros el nombre, el peso y la altura de del usuario. El programa deberá mostrar en formato de tabla los datos proporcionados en columnas y adicionalmente el IMC. Se debe crear un clase Usuario y crear un objeto de dicha clase a partir de un array de Strings. Se deberá crear en la clase un método estático que recibe un Array de Strings y devuelve un objeto de la clase Usuario. Revisar con el profesor el patrón Factory Method simple.
3. Una frutería precisa de un programa que calcule la factura de la compra de sus clientes. La frutería sólo trabaja con dos productos: peras y manzanas cuyo valor varía cada día. El programa pedirá por consola al tendero el precio/kg sin IVA de las peras y las manzanas cada mañana al comienzo de su jornada laboral. El IVA ya está prefijado en el programa. Posteriormente el programa permite introducir la cantidad de cada tipo de fruta que desea comprar cada cliente y calculará el precio total de la factura para dicho cliente, mostrando en forma de tabla el tipo de fruta, cantidad en Kg, precio con IVA por Kg, precio total por fruta. Haz que el programa se ejecute continuamente hasta que el tendero decida salir, incrementando el número de cliente mostrado en el ticket mostrado por pantalla.
Tip: Realizar este ejercicio empleando Arrays de objetos y empleando una clase que abstraiga todo lo común que tienen los ítems que vende el tendero: e.g. Producto. **TIP2:** Usa String.format y especificadores de formato y especifica el ancho del campo %10s, %10.2f etc. para la tabla.

```
|-----|
| Cliente                               | 1 | | |
|---|---|---|---|
| Manzanas | 2 kg | precio Kg con IVA 1,99 | 3,98 € |
| Peras    | 3 kg | precio Kg con IVA 2,25 | 6,75 € |
|-----|
| Total con IVA 10,73 €                 |
|-----|
```

4. Se debe realizar un programa que solicite por consola el número de alumnos que tiene un curso. Posteriormente se solicitarán las notas de 3 exámenes (2 parciales y uno final) por cada alumno. El programa deberá mostrar las calificaciones de cada alumno en formato de tabla y la nota final (suponiendo un valor de 10%, 10% y 80% respectivamente). Por último se deberá obtener la media por cada examen y de la nota final de la clase. Utiliza Arrays de objetos para su resolución.

Ejercicios de extensión:

- Tomando como base lo visto hasta el momento, escribir un programa que muestre en pantalla los partidos del pleno al 15, pida los resultados y muestre finalmente la quiniela*. Desarrolla el ejercicio empleando clases y objetos que modelen el dominio del problema.

QUINIELA		
Levante	Las Palmas	X
Real Madrid	Barcelona	1
Betis	Sevilla	2

*La quiniela son un total de 15 partidos donde X representa empate, 1 ganador el equipo local (que aparece en primera columna) y 2 ganador el equipo visitante (que aparece en la segunda columna). Esto último se determina a partir de los goles del local y visitante introducidos por el usuario. Pensad una solución empleando clases y objetos con POO.

Versión 1: establece un ancho de columna fijo lo suficientemente grande para que los nombres de los equipos quepan, en caso contrario se truncarán. Todo con operadores de formato. Utiliza `String.format` y métodos de la clase `String` que te puedan ser útiles (`repeat`, etc.).

Versión2: intenta ingeníartelas para que el ancho venga definido por el nombre de equipo más largo introducido por el usuario. Revisa los atributos y métodos que tiene la clase `String` en Java.

- Contesta a las siguientes preguntas sin IDE. Realiza un pequeño ejemplo para comprobar su solución empleando cuando sea necesario el depurador y puntos de interrupción.
 - `Item[] misObjetos = new Item[7];`
¿Qué valores tendrá la variable `misObjetos`? ¿Qué valor tiene la propiedad `length` de `misObjetos`?
 - `final int AMOUNT_OF_PEOPLE = 4;`
¿Es posible modificar la variable `AMOUNT_OF_PEOPLE` después de su inicialización? ¿Por qué?
 - `System.out.println("Hello World");`
¿Por qué es posible acceder al atributo `out` de la clase `System` sin crear un objeto de la clase `System` con `new`? Usa F12 sobre el atributo `out` para inspeccionar el IDE y ver qué ocurre.
 - ¿Cuál es la diferencia entre emplear `import` o `import static`? Pon un ejemplo de cuando utilizar cada uno con `Biblioteca.jar`
 - ¿Qué significa el error *"Cannot use this in a static context"* que se produce en el siguiente código?
¿Por qué se produce? ¿Cómo arreglarías el error?

```
public class Person {
    String name;
    int ageYears;
    float heightCm;
    float weightKg;
    public Person() {
    }
    public static float calculateIMC() {
        float weightSquaredinMeters = Math.pow(this.heightCm/100.0f, 2);
        return (this.weightKg/weightSquaredinMeters);
    }
}
```