

UNIDAD TEMÁTICA 1: Introducción a JAVA , PARTE I

TRABAJO DE APLICACIÓN 2

Ejercicios #1 y #2

Para realizar estos ejercicios, cada Equipo se dividirá en dos subequipos, “A” y “B”. El subequipo A realizará el **Ejercicio 1** y el subequipo B el **Ejercicio 2**.

Pasos a desarrollar:

1. Crear un paquete, dentro de **UT1**, “**TA2**” Y verificar que se refleja en todas las computadoras del Equipo.
2. **Subequipo A** desarrolla el **Ejercicio #1** y lo prueba; **Subequipo B** desarrolla el **Ejercicio #2** y lo prueba.
3. Los subequipos sincronizan el código en el repositorio. (**hasta hora 18:50**) (o intercambian el código por otros medios)
4. El **subequipo B** analiza y prueba el código generado para el Ejercicio #1, con los comentarios del **subequipo A**. Sincroniza eventuales cambios.
5. Se responden las preguntas planteadas en pantalla. (**hora 18:55**)
6. El **subequipo A** analiza y prueba el código generado para el Ejercicio #2, con los comentarios del **subequipo B**. Sincroniza eventuales cambios.
7. Se responden las preguntas planteadas en pantalla (**hora 19:05**)

Ejercicio #1 – subequipo “A”

Muchos cálculos pueden ser expresados de manera concisa usando la operación “**multsuma**”, que toma tres operandos y computa $a*b + c$. Algunos procesadores incluso proveen una implementación de hardware para esta operación para números de punto flotante.

Crear un nuevo programa llamado **Multsuma.java**. Escribir un método llamado **multsuma** que toma tres *doubles* como parámetros y que imprime el resultado de “**multisumarlo**”.

Escribir un método **main** que testee **multsuma** invocándolo con unos pocos parámetros simples, como por ejemplo 1.0, 2.0, 3.0, y después imprima por consola el resultado, que en ese caso debería ser 5.0.

- a) Utilizando la función desarrollada “**multsuma**” computar:

$$\sin \frac{\pi}{4} + \frac{\cos \frac{\pi}{4}}{2}$$

- b) Aplicando el mismo método, indicar el resultado de calcular

$$\log 148 + \log 296$$

Ejercicio #2 – subequipo “B”

Hacer un programa en Java que indique si un número dado es primo. Si lo es, el programa deberá calcular la suma de los números pares desde 0 hasta el número dado, y en caso contrario deberá calcular la suma de los números impares desde 0 hasta el número dado. Utilizar solamente sentencias **while o do-while**.

Posible código para calcular si un número es primo:

```
public static boolean isPrime(long n) {
    boolean prime = true;
    for (long i = 3; i <= Math.sqrt(n); i += 2)
        if (n % i == 0) {
            prime = false;
            break;
        }
    if ((n%2 != 0 && prime && n > 2) || n == 2) {
        return true;
    } else {
        return false;
    }
}
```

Ejercicio #3 (una sola computadora)

Referencia: ORACLE JAVA TUTORIAL.

Un marcapasos debe funcionar consumiendo la menor cantidad de energía posible y ser lo más barato posible, por lo que incluye una mínima cantidad de memoria.

Con sus sensores obtiene continuamente las siguientes medidas, con los rangos que se indican:

- Presión Sanguínea (0-250)
- Frecuencia cardíaca (0-226)
- Nivel de azúcar en sangre. (0 - 1000).

Además para autodiagnóstico mantiene un registro de:

- Máxima fuerza a la que fue expuesto (0-3.000.000.000).
- Mínimo tiempo entre latidos (0-100 con decimales).
- Batería restante. (% de batería restante con decimales con la mayor precisión posible).
- Código del Fabricante (Números y letras, máximo 8 caracteres).

Trabajo a realizar:

- a) Crear una clase "**Marcapasos**" que contenga variables para almacenar esta información. Utilice los **tipos de datos primitivos** de java teniendo en cuenta que se quiere consumir la menor cantidad de memoria posible.
- b) **Calcular cuánta memoria** consume un objeto de su clase (considerando solamente el espacio ocupado por las variables indicadas).
- c) Responder las preguntas presentadas en pantalla