BÚSQUEDA DE INFORMACIÓN

**1. ¿Qué es un operador ternario en Java? ¿Cómo se utiliza y para qué? Pon un ejemplo.**

Es una simplificación de IF – ELSE

La estructura del operador ternario ?: es la siguiente:

resultado = (condicion)?valor1:valor2;

Donde a la variable resultado recibirá el valor1 en el caso de que la condición sea true o bien el valor2 en el caso de que la condición sea false.

Ejemplo :

System.out.println(numero > 10 **?** "El número es mayor a 10."**:**"El número es menor o igual a 10.");

**2. ¿A qué se refiere el concepto de encapsulamiento en POO y particularmente en Java?**

La encapsulación es un principio fundamental de la programación orientada a objetos y consiste en ocultar el estado interno del objeto y obligar a que toda interacción se realice a través de los métodos del objeto. Para ello, el acceso a los atributos se establece como privado y se crean 2 métodos por cada atributo, un **getter** y un **setter**.

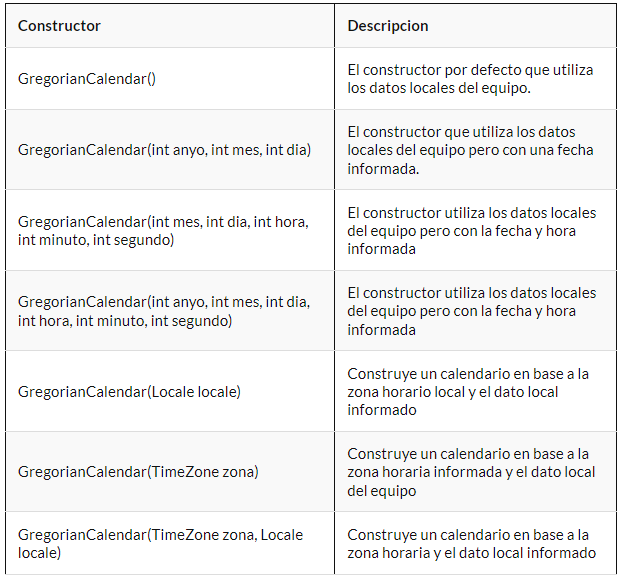
**3. Estudia la clase GregorianCalendar. Estudia sus métodos y explica el uso de los que te parezcan más útiles. Pon un ejemplo de cada método que expliques.**

**GregorianCalendar** es una subclase concreta de Calendar y proporciona el

sistema de calendario estándar utilizado por la mayor parte del mundo.

Tiene muchos constructores con diferentes entrada de parámetros

Por ejemplo,



La clase **Calendario** es una [clase abstracta](https://baike.baidu.com/item/%E6%8A%BD%E8%B1%A1%E7%B1%BB?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/GregorianCalendar/_blank) que proporciona métodos para convertir entre un instante específico y un conjunto de campos de calendario (como AÑO, MES, DÍA\_DE\_MES, HORA, etc.) .

Esta clase también proporciona campos y métodos adicionales para implementar un sistema de calendario concreto fuera del alcance del paquete. Estos campos y métodos se definen como protegidos.

El método getInstance() de Calendar devuelve un objeto Calendar cuyos campos de calendario se han inicializado con la fecha y hora actuales:

Por ejemplo,

Calendar rightNow = Calendar.getInstance();

Calendar tiene dos modos para interpretar los campos del calendario, indulgente y no indulgente(lenient y non-lenient). Cuando un calendario está en modo indulgente, acepta valores en un rango más amplio que los campos de calendario que genera. Todos los campos de calendario se normalizan cuando Calendar vuelve a calcular los valores de los campos de calendario para que get() los devuelva.

Por ejemplo,

GregorianCalendar en modo indulgente interpreta el 1 de febrero como

MES == ENERO, DÍA\_DE\_MES == 32.

Cuando un calendario está en modo no indulgente, lanza una excepción si hay inconsistencias en sus campos de calendario.

**Los metodos**.

|  |  |
| --- | --- |
| isLeapYear(int anyo) | Determina si el año informado es bisiesto |
| roll(int campo, boolean up) | Adiciona o substrae una unidad simple de tiempo sobre el campo de tiempo informado sin reportar cambios en los campos mas grandes |
| set(int campo, int value) | Setea el campo del tiempo con el valor informado |

|  |  |
| --- | --- |
| add(int campo, int cantidad) | Agrega la cantidad (con signo) de tiempo informada en el campo de tiempo dado segun las reglas del calendario |
| equals(Objeto obj) | Compara el calendario Gregoriano con el objeto de referencia |

El método getTime() nos permite obtener un objeto Date a partir del objeto Calendar y al objeto Date le podemos pedir una representación tipo String con el método toLocaleString().

Por ejemplo,

Calendar c1 = GregorianCalendar.getInstance();

        System.out.println("Fecha actual: "+c1.getTime().toLocaleString());

El método set() nos permite alterar la fecha

Por ejemplo,

 c1.set(2000, Calendar.AUGUST, 31);