

Nombre de la actividad: Primeras prácticas Tema 3

Contenido relacionado: Tema 3 Clases

En esta práctica vamos a crear una clase `Fecha` que nos permita trabajar con fechas. La clase debe almacenar una serie de atributos y debe definir una serie de métodos para realizar distintas operaciones básicas.

Antes de escribir código, piensa cuidadosamente el diseño de la clase `Fecha`.

1. ¿Qué información debe almacenar cada objeto? Si un mismo dato puede almacenarse en distintos formatos, ¿usarías varios atributos para representar la misma información? ¿Por qué? ¿Qué formato crees que es más conveniente?
2. ¿Qué constructores proporcionarías? Observa que los constructores tendrán que inicializar los datos internos del objeto.
3. Piensa qué métodos crees que sería útil incluir.
4. ¿Se te ocurre alguna información que sería conveniente incluir en la clase, común para todos los objetos `Fecha`?

Atributos de la clase Define los atributos de la clase `Fecha`.

Constructores de la clase

La clase `Fecha` debe proporcionar constructores que inicialicen objetos de tipo `Fecha`:

- Un constructor sin parámetros, se usará para instanciar la `Fecha` correspondiente al día actual.
- Un constructor con un parámetro `day` para instanciar la `Fecha` del día indicado para el mes y año en curso.
- Un constructor con dos parámetros `day` y `month` para instanciar la `Fecha` del día y mes indicado para el año en curso.

Programar dichos constructores, de acuerdo a las siguientes declaraciones:

```
public Fecha() { //TODO ...}  
public Fecha(String day) { //TODO ...}  
public Fecha(Integer day) { //TODO...}  
public Fecha(Integer day, Integer month) { //TODO ...}  
public Fecha(Integer day, Integer month, Integer year) { //TODO...}
```

Para conseguir valores por defecto se puede utilizar la clase `Calendar`, ejemplo:

```
java.util.Calendar.getInstance().get(Calendar.DAY_OF_MONTH)  
java.util.Calendar.getInstance().get(Calendar.MONTH)  
java.util.Calendar.getInstance().get(Calendar.YEAR)
```

Métodos de acceso de los atributos de la clase

Los atributos del objeto conviene que se declaren de acceso privado para evitar errores en su modificación. Programa los métodos `get` y `set` necesarios para leer y modificar la información encapsulada en los objetos `Fecha`.

Método `toString()` de representación del objeto

El método `toString()` tiene un significado especial en los objetos Java. Es el método que se utiliza para obtener una representación como cadena de texto de dicho objeto mostrando los valores de los atributos en un instante determinado.

1. Programa el método `toString()` que tiene la siguiente declaración:

```
public String toString() {  
    /* completar */  
}
```

Este método tiene que devolver una cadena de texto con la representación de una fecha, según el siguiente formato:

día(Numero) **de** mes(Texto) **de** año(Numero)

EJEMPLO : 6 **de** Agosto **de** 2007

Método `compareTo()` para comparar fechas

El método `compareTo()` tiene un significado especial en los objetos Java. Es el método que se utiliza para comparar dos objetos.

LECTURA RECOMENDADA: Consulta la API de la interfaz [Comparable](#) para comprobar qué valor debe devolver en los distintos casos: según si el objeto con el que comparamos el objeto actual es mayor, menor o igual que éste.

1. Programa el método `compareTo()` que tiene la siguiente declaración:

```
public int compareTo(Fecha otraFecha) {  
    /* ... */  
}
```

El método compara la fecha almacenada en el objeto con la que se le pasa como parámetro, devolviendo como resultado:

- -1 si la fecha contenida en el objeto actual es anterior a la que se pasa por parámetro.
- 0 si son iguales.
- 1 si la fecha contenida en el objeto actual es posterior a la que se pasa por parámetro.

Método auxiliar sobrecargado

Recuerda que en una clase pueden aparecer varios métodos con el mismo nombre, siempre que tengan distintos parámetros de entrada. A esto se le denomina sobrecarga de métodos (en inglés, “overload”).

1. Programa un método auxiliar que devuelve el número de días que tiene el mes que se le pasa como parámetro. Sobrecarga el método de modo que pueda utilizarse el número o el nombre del mes. Los métodos tienen las siguientes declaraciones:

```
public Integer diasMes() {  
    //devuelve los días del mes actual  
}  
public Integer diasMes(Integer year, Integer month) {  
    //devuelve los días del mes indicado  
}
```

Método main

```
public static void main(String[] args) {  
    Fecha today = new Fecha();  
    System.out.println(today+ " cuyo mes tiene "+  
                        today.diasMes()+ " días");  
  
    Fecha octubre7 = new Fecha(7);  
    System.out.println(octubre7);  
  
    Fecha octubre10 = new Fecha(10,10);  
    System.out.println(octubre10);  
  
    Fecha octubre10Del2003 = new Fecha(10,10,2003);  
    System.out.println(octubre10Del2003);  
  
    if (octubre7.esAnterior(octubre10)) {  
        System.out.println("la fecha " +  
                            octubre7+ " es anterior a "+octubre10);  
    }  
}
```

Resultado esperado:

día 8 de Octubre de 2022 cuyo mes tiene 31 días

día 7 de Octubre de 2022

día 10 de Octubre de 2022

día 10 de Octubre de 2003

la fecha día 7 de Octubre de 2022 es anterior a día 10 de Octubre de 2022

Fecha de límite de entrega: Dos semanas desde el día de trabajo en clase

Formato y espacio de entrega:

- La entrega se realizará a través de Blackboard.
- Se entregará en formato digital en un archivo comprimido en el enlace Practica1 del módulo de aprendizaje del Tema 3. **Se espera un documento pdf o en su defecto doc con el código fuente y el resultado de la consola.**
- Si es demasiado pesado para subir directamente a Blackboard, subirlo a vuestra nube de OneDrive de u-tad y compartirlo a través de Blackboard con la opción de subir archivo de almacenamiento en la nube.
- El alumno recibe un mail de confirmación cada vez que hace una entrega. Es su responsabilidad revisar que la actividad se ve correctamente (en caso de no visualizarla, contactar con el profesor).
- Sólo se permite una entrega.
- No se permitirán entregas fuera de plazo.