

ANEXO DE AYUDA:**1) Implementación completa de la clase Fecha**

A continuación se facilita al alumno la implementación completa de la clase Fecha y se comenta algunos de sus métodos (la solución mostrada no es la única posible: existen otras formas distintas de implementar algunos de los métodos que también son perfectamente válidas).

```
package libClases;

import java.util.Scanner;

public final class Fecha implements Cloneable, Proceso {
    private int dia;
    private int mes;
    private int anio;

    public void setFecha(int d, int m, int a) {
        int dmax, diaMes[] = {31,28,31,30,31,30,31,31,30,31,30,31};
        anio=a; //VIP debo asignar año para al llamar a bisiestos() tengo el año bien
        if (m<1) //si el mes es incorrecto
            m=1;
        else if (m>12) //si el mes es incorrecto
            m=12;

        dmax=diaMes[m-1];
        if (m==2 && bisiestos())
            dmax++;
        if (d>dmax)
            d=dmax;
        else if (d<1)
            d=1;

        dia=d;
        mes=m;
    }

    public Fecha(int dia, int mes, int anio) {
        setFecha(dia, mes, anio);
    }

    public Fecha(Fecha f) {
        dia=f.dia;
        mes=f.mes;
        anio=f.anio;
    }

    public int getDia() { return dia; }
    public int getMes() { return mes; }
    public int getAnio() { return anio; }
```

Para implementar el método toString podemos formar el String con una serie de if o bien utilizar el método format de la clase String que permite formatear el String que se quiere formar (en el código se muestra comentado como última línea del método)

```
public String toString() {
    String s="";
    if (dia<10) s=s+0;
    s=s+dia+"/";
    if (mes<10) s=s+0;
    s=s+mes+"/"+anio;
    return s;
    //LO ANTERIOR SE PUEDE SUSTITUIR POR LO SIGUIENTE
    //return String.format("%02d/%02d/%02d", dia, mes, anio);
}
```

```

public boolean bisiesto() {
    boolean b=false;
    if (anio % 4 == 0) {
        b=true;
        if (anio%100 == 0 && anio%400 !=0 )
            b=false;
    }
    return b;
}

public void ver() {
    System.out.println(this/*.toString()*/);
}

```

El método pedirFecha() es estático ya que en el main() de prueba se invoca así:

```
f3=Fecha.pedirFecha(); // pide una fecha por teclado
```

En dicho método vamos a controlar todos los errores que se puedan cometer y que son:

- Que no se introduzcan las 3 partes de la fecha o se introduzcan más de la cuenta
- Que el día, mes y/o año introducido no sea un valor numérico
- Que la fecha introducida no sea una fecha válida

En esos 3 casos vamos a lanzar una excepción que capturaremos en el propio método

```

public static Fecha pedirFecha() {
    Fecha fecha = null;
    boolean valida = false;
    Scanner sc = new Scanner(System.in);
    int dia, mes, anio;
    do {
        System.out.println("Introduce la Fecha (dd/mm/aaaa): ");
        String cadena = sc.next();
        String[] tokens = cadena.split("/");
        try {
            if (tokens.length != 3)
                throw new NumberFormatException();

            dia = Integer.parseInt(tokens[0]); //parseInt lanza la excepcion
            mes = Integer.parseInt(tokens[1]); //NumberFormatException si no
            anio = Integer.parseInt(tokens[2]); //puede convertir el String a int

            fecha = new Fecha(dia, mes, anio);

            if (fecha.getDia() != dia || fecha.getMes() != mes)
                throw new NumberFormatException();

            valida=true;
        } catch (NumberFormatException e) {
            System.out.println("Fecha no valida");
        }
    } while(!valida);
    sc.close();
    return fecha;
}

```

El método mayor() también es estático ya que en el main() de prueba se invoca así:

```
Fecha.mayor(f2,f1); // pide una fecha por teclado
```

En vez de tener una serie de if else anidados para discernir cuando una fecha es mayor que otra podemos convertir ambas fechas en un número usando la siguiente fórmula:

$$\text{año} \times 1000 + \text{mes} \times 100 + \text{día}$$

de forma que la comparación se pueda hacer con un simple if:

```

static public boolean mayor(Fecha f1, Fecha f2) {
    /*
    boolean esmayor=false;
    if (f1.anio>f2.anio)
        esmayor= true;
    else if (f1.anio<f2.anio)
        esmayor= false;
    else {
        if (f1.mes>f2.mes)
            esmayor= true;
        else if (f1.mes<f2.mes)
            esmayor= false;
        else {
            if (f1.dia>f2.dia)
                esmayor= true;
            else
                esmayor= false;
        }
    }
    return esmayor;
    */
    if (f1.anio*10000+f1.mes*100+f1.dia>f2.anio*10000+f2.mes*100+f2.dia)
        return true;
    else
        return false;
}
/*
2011/11/10    2011*10000+11*100+10=20111110
2011/11/09    2011*10000+11*100+9 =20111109
*/

```

En cuanto al método clone() como la clase Fecha es una clase final la implementación del método clone() se puede hacer simplemente invocando “el constructor de copia” o cualquier otro método parecido (tal como se indica en la primera línea comentada):

```

public Object clone() {
    //return new Fecha(this);
    Object obj=null;
    try {
        obj=super.clone(); //se llama al clone() de la clase base (Object)
                           //que hace copia binaria de los atributos
    } catch(CloneNotSupportedException ex) {
        System.out.println(" no se puede duplicar");
    }
    return obj;
}

```

En cuanto al método equals() el esquema debe ser como el indicado aquí (los 3 primeros if mostrados deben aparecer siempre):

```

public boolean equals(Object obj) { //true sin son iguales
    if (this == obj) return true; //si apuntan al mismo sitio son iguales
    if (obj == null) return false;
    if (getClass() != obj.getClass())//if (!(obj instanceof Cliente))
        return false; // si los 2 no son de la misma clase no son iguales
    Fecha c = (Fecha) obj;
    return (dia==c.dia && mes==c.mes && anio==c.anio);
}

public static void main(String[] args) {
    Fecha f1 = new Fecha(29,2,2001), f2 = new Fecha(f1), f3 = new Fecha(29,2,2004);
    final Fecha f4=new Fecha(05,12,2003); //es constante la referencia f4
    System.out.println("Fechas: " + f1.toString() + ", " + f2 + ", " + f3 + ", " + f4);
    f1=new Fecha(31,12,2016); //31/12/2016
    f4.setFecha(28, 2, 2008); //pero no es constante el objeto al que apunta
    System.out.println(f1 + " " + f2.toString() + " " + f3 + " " + f4 + " " + f1);
    f1=new Fecha(f4.getDia()-10, f4.getMes(), f4.getAnio()-7); //f1=18/02/2001
    f3=Fecha.pedirFecha(); // pide una fecha por teclado
    if (f3.bisiesto() && Fecha.mayor(f2,f1))
        System.out.println("El " + f3.getAnio() + " fue bisiesto, " + f1 + ", " + f3);
    System.out.print("Fin\n");
}
}

```