Parte 4 - POO | Ignacio CAVALLO

Clase 21 | 27/05

Conceptos Aprendidos

En Clase:

- Expansión de habilidades al manejar una clase.

Fuera de clase:

- Uso de HasMap
- Buenas practicas en la implemetnación de Constructores.
- Recursión
- Mejora de Código.
- No desanimarse.
- Volver sobre el mismo Código.

Unico Ejecicio

Continuamos con **Programación Orientada a Objetos**.

En clase se planteó un problema que al principio parecia simple, pero ***terminó siendo mucho más COMPLEJO***.
Se pidió la implementación de una clase FECHA, con los siguientes atributos y métodos :

```
int dia;
int mes;
int año
public boolean fechaCorrecta(){}
public diaSiguiente(){}
```

El método fechaCorrecta debe confirmar que sea una fecha válida, y el díaSiguiente debe seguir el orden cronologico de las fechas.
Ahora, analizandolo en mayor profundidad me percaté que tenía varios niveles de complejidad:

> 1. Los días y meses no pueden ser menores a 1.
> 2. Los meses no pueden ser mayor a 12.
> 3. Los días no pueden ser mayor a 31.
> 4. Dependiendo del mes, el último día será distinto.
> 5. En función a al año, el último día de febrero será 28 o 29.

Después de analizarlo en clase, y fuera de ella, decidí separar el problema de la siguiente manera:

- Definir el constructor con una restricción:
  Si los días son validos (1-31), serán cargados.
    - Si los meses son validos (1-31), serán cargados.
    - El año al aceptar valores siempre negativos no tiene restricción.
    - Si alguna de las condiciones no son válidas, la fecha por defecto será:
    - 4 / 2 /42
    - Esto simplifica el uso del mapeo y de el día siguiente.
- Implementar un Check de año bisiesto:
- Para evitar el uso excesivo de condicionales, consideré pertinente utilizar un
    - HashMap, que vincule el mes con su respectivo último día.
    - A su vez, utilicé un operador Tenary en el mes de Febrero, este usa método para comprobar
    - si es Bisiesto, en caso verdadero el último días es 29, sino, 28.
- Por requerimiento del ejercició implemente la función fechaCorrecta.
    - Esta llama al mapeoFecha y devuelve TRUE si el día en cuestión es menor al que está asignado en el HASH.
- Día Siguiente fué el método más complejo, por que tenemos varias restricciones:
    - ```
      1. Si el día actual es mayor al día máximo del mes. En este caso debemos pasar al próximo mes.
      ```
    - ```
      2. Si el mes actual es mayor al último mes del año, debemos pasar a un nuevo año.
      ```
    - ```
      3. No debemos preocuparnos si el día o el mes es incorrecto, ya que por definición de la clase, esto no es permitido.
      ```

Mi implementación FINAL fué la siguiente:

```
package com.nacho;

import java.util.*;

/*          Clase Fehca:
 Atributos:
 dia
 mes
 año
 Métodos Getters y Setters.
```
```

```
 fechaCorrecta()
    Comprueba si fecha es correcta.
diasSiguientes()
    Imprime los siguientes días

 */
public class Fecha {
    private int dia;
    private int mes;
    private int año;
    private Map mapa = new HashMap(); //HasMap para mapear los días 28,29,30 y 31

    /* El constructor es definido por una restricción:
     * Si los días son validos (1-31), serán cargados.
     * Si los meses son validos (1-31), serán cargados.
     * El año al aceptar valores siempre negativos no tiene restricción.
     * Si alguna de las condiciones no son válidas, la fecha por defecto será:
     * 4 / 2 /42
     * Esto simplifica el uso del mapeo y de el día siguiente.
     * */

    public Fecha(int dia, int mes, int año) {

        if ((dia >= 1 && dia <= 31) && (mes >= 1 && mes <= 12)) {
            this.dia = dia;
            this.mes = mes;
            this.año = año;
        } else {
            this.dia = 4;
            this.mes = 2;
            this.año = 42;
            System.out.println("Día o Mes incorrecto");
            System.out.println("Fecha por defecto: 4/2/42");
        }


    }
    /* Getters y Setters.*/

    public int getDia() {
        return dia;
    }

    public void setDia(int dia) {
        this.dia = dia;
    }

    public int getMes() {
        return mes;
    }

    public void setMes(int mes) {
        this.mes = mes;
    }
```

```
    public int getAño() {
        return año;
    }

    public void setAño(int año) {
        this.año = año;
    }

    /* Método que devuelve verdadero en caso de ser Bisiesto */
    private boolean esBisiesto() {
        return (this.año % 4 == 0) && ((this.año % 100 != 0) || (this.año % 400 ==
0));
    }

    /* Para evitar el uso excesivo de condicionales, consideré pertinente utilizar
un
    * HashMap, que vincule el mes con su respectivo último día.
    * A su vez, utilicé un operador Tenary en el mes de Febrero, este usa método
para comprobar
    * si es Bisiesto, en caso verdadero el último días es 29, sino, 28.
    * */

    public void mapeoFecha() {
        mapa.put(1, 31);
        mapa.put(2, (esBisiesto() ? 29 : 28)); // Tenary para check Bisiesto
        mapa.put(3, 31);
        mapa.put(5, 31);
        mapa.put(6, 30);
        mapa.put(7, 31);
        mapa.put(8, 31);
        mapa.put(9, 30);
        mapa.put(10, 31);
        mapa.put(11, 30);
        mapa.put(12, 31);
    }

    /* Por requerimiento  del ejercició implemente la función fechaCorrecta.
    * Esta llama al mapeoFecha y devuelve TRUE si el día en cuestión es menor al
que
    * está asignado en el HASH.*/


    public boolean fechaCorrecta() {
        mapeoFecha();
        return this.dia < (int) mapa.get(this.mes);
    }
    /* Representación en String de la Clase*/
    public String toString() {
        return this.dia + "/" + this.mes + "/" + this.año;
    }

    /* Está fué el  método más complejo, por que tenemos varias restricciones:
    *       1. Si el día actual es mayor al día máximo del mes. En este caso
```

```
     *           debemos pasar al próximo mes.
     *       2. Si el mes actual es mayor al último mes del año, debemos pasar
     *          a un nuevo año.
     *       3. No debemos preocuparnos si el día o el mes es incorrecto, ya que
     *          por definición de la clase, esto no es permitido.
     *
     * */

    public void diaSiguiente() {
        if (!fechaCorrecta()) { //Si la fecha es mayor al día máx.
            if (getMes() < 12) { // si no es el último mes del año
                setDia(1); // El día es 31 pasa a 1.
                this.mes += 1; // el mes 1 pasa a mes 2.
                System.out.println("********");
                System.out.println("Pasamos a otro mes: " + getMes());
                System.out.println("********");
            } else if (getMes() == 12) { // si es el ultimo més del año.
                                         // Ej: 31/12/1999
                setDia(1); // el día pasa a 1.
                setMes(1); // el mes 12 pasa a mes 1.
                this.año += 1; //  el año se incrementa en 1 ==>2000
                System.out.println("********");
                System.out.println("FELIZ AÑO NUEVO");
                System.out.println("Te deseamos un muy buen " + getAño());
                System.out.println("********");

            }
        } else { //Si la fecha es menor al último día
            this.dia += 1; // Se aumenta en uno.
        }
        System.out.println(this.dia + "/" + this.mes + "/" + this.año);
    }

    /* Implemnentación Recursiva para imprimir los n días que
    * requiera el usuario.*/

    public void printDiaSiguiente(int stop) {
        if (stop == 0) {
            System.out.println(this.dia + "/" + this.mes + "/" + this.año);
        } else {
            diaSiguiente();
            printDiaSiguiente(stop - 1);
        }
    }

}
}

/* ---------RESULTADO------------*/

 ---- FECHAS INCORRECTAS ----

Día o Mes incorrecto // -15/3/1927
Fecha por defecto: 4/2/42
```

```
Fecha Correcta?: true // ==> Es true por que se aplica la
                     // fecha por defecto
Día o Mes incorrecto
Fecha por defecto: 4/2/42 // 21/42/1933
Fecha Correcta?: true

 ---- AÑO BISIESTO ----

Fecha Ingresada:27/2/2020

Fecha Correcta?: true

Los próximos 5 dias son:
28/2/2020
29/2/2020
********
Pasamos a otro mes: 3
********
1/3/2020
2/3/2020
3/3/2020
3/3/2020

 ---- AÑO NO BISIESTO ----

Fecha Ingresada:27/2/2018

Fecha Correcta?: true

Los próximos 5 dias son:
28/2/2018
********
Pasamos a otro mes: 3
********
1/3/2018
2/3/2018
3/3/2018
4/3/2018
4/3/2018

 ---- AÑO Nuevo ----

Fecha Ingresada:30/12/2020

Fecha Correcta?: true

Los próximos 5 dias son:
31/12/2020
********
FELIZ AÑO NUEVO
Te deseamos un muy buen 2021
********
1/1/2021
2/1/2021
```

```
3/1/2021
4/1/2021
4/1/2021
```