Tarea 4 – Objetos con atributos tipo array

Enunciado

Hasta ahora hemos creado objetos, con campos o atributos de tipos de dato primitivo.

Ejemplo: class Persona {

private **int** edad;

Dentro de una clase también podemos tener variables de instancia o campos que sean de tipo array.

Vamos a escribir un programa para realizar un seguimiento del clima. Se desea registrar la temperatura máxima de cada día durante un mes.

1. Creamos la clase **MesCollector.java** con las siguientes variables de instancia: **mes**, **anyo**, **diasMes** de tipo entero, un array de tipo entero llamado **temperatura**. En dicho array iremos almacenando las temperaturas máximas de ese mes.
   La temperatura será añadida para cada día después de construir el objeto MesCollector.

2. Por tanto, para construir el MesCollector solo necesitaremos mes y anyo, el array temperatura se creará e inicializará con valores 0 en cada celda. Para el tamaño del array deberemos pensar en crear un array que nos permita almacenar todas las temperaturas en caso de que el objeto del mes creado sea un més con el máximo de días posible.

3. El campo mes, que es un entero y será 1 para el mes de enero, 2, febrero, y asi sucesivamente. Nos permitirá inicializar la variable diasMes, es decir, si el mes es 1 (enero) diasMes será igual 31. En caso de que el mes sea un entero menor a 1 o mayor a 12 diasMes será igual a 0. Obviaremos los años bisiestos.

4. Crea un método getTemperatura, donde se le pasa un entero con un número de día y debe devolver cuál es la temperatura máxima para ese día. Si el entero que se le pasa está fuera de rango devolverá -1. Ejemplo: getTemperatura(10) → devuelve la temperatura máxima del día 10 del mes. Recuerda las temperaturas máximas están almacenadas en el array.

5. Crea un método setTemperatura, le pasamos número de día y una temperatura y la asigna en el array para ese día. Si el día no existe no asigna nada.

6. En Java, cada vez que se crea un objeto, por defecto tenemos un método llamado toString() que imprime el objeto. Si no añadimos el método en nuestra clase, java utiliza el que se crea automáticamente e imprime nombre de la clase más un código hash. Por tanto, lo definimos en nuestra clase e imprimimos el objeto completo con todos sus campos y la temperatura para cada día. Si no se ha asignado temperatura para algún día imprimimos “no hay datos”.
   ```
   @Override
   public String toString() {}
   ```
   En un futuro veremos que significa la anotación @Override.

7. Para probar el código creamos una clase llamada MesTester.java con el siguiente código.
   ```
   public class MesTester {

       public static void main( String[] args) {
           Scanner sc = new Scanner( System.in);
           MesCollector enero = new MesCollector( 1, 2021);
           String respuesta = "Y";
   ```

```
            while (respuesta.toUpperCase().charAt(0) == 'Y') {
                System.out.print("Inserta día  ");
                int dia = sc.nextInt();

                System.out.print("Inserta temperatura ");
                int temperatura = sc.nextInt();

                if (enero.getTemperatura(dia) == -1){
                    System.out.println(" error de entrada ");
                } else {
                    enero.setTemperatura(dia, temperatura );
                }

                System.out.print("Continuar (Y/N)? ");
                respuesta = sc.next();
            }
            System.out.println(enero);
        }
    }
```

Entrega

- Copia y pega el código en PDF.