

Examen de Prácticas de Programación I (Ingeniería Informática)

Septiembre 2009

Se dispone de una clase llamada `Fecha` que permite almacenar una fecha compuesta por el día, mes y año. Dispone de una operación, `numDias`, para calcular el número de días entre dos fechas. La interfaz de esta clase es:

```
public class Fecha {
    /**
     * Constructor al que se le pasa el día, mes y año
     */
    public Fecha(int dia, int mes, int año) {...}

    /**
     * Retorna el número de días entre la fecha del objeto actual y la fecha
     * del objeto f (positivo si f es posterior a actual)
     */
    public int numDias(Fecha f) {...}
}
```

Se dispone también de otra clase llamada `HistoriaConsumos` para almacenar el historial de consumos eléctricos de un cliente. Este historial consiste en guardar lecturas del contador eléctrico, cada una junto a la fecha en que se realizó. Para ello la clase guarda en un array, `kwh`, las lecturas del contador en kilovatios hora, y en otro array, `fecha`, la fecha de cada lectura. Estos arrays tienen una estructura “paralela”, de modo que la lectura número i se guarda en `kwh[i]` (la lectura del contador) y en `fecha[i]` (la fecha de esa lectura). Se suponen las fechas ordenadas de modo que la fecha i es anterior a la $i+1$.

Ambas tablas son de tamaño variable y se van rellenando desde las casillas de índices más bajos. El atributo entero `num` indica cuántas lecturas válidas hay, de modo que los arrays tienen lecturas válidas entre las casillas 0 a `num-1`. El resto de las casillas no tienen valores válidos.

La clase dispone de un constructor y de un método para añadir una lectura del contador con su fecha. Los atributos y la interfaz de esta clase son:

```
public class HistoriaConsumos {

    // Dos arrays paralelos: kwh[i] se leyó en la fecha fecha[i]
    private double[] kwh; // lectura del contador en kilovatios-hora
    private Fecha[] fecha; // fecha de lectura del contador
    private int num; // numero de lecturas del contador válidas

    /**
     * Constructor que crea el historial de consumos vacío,
     * pero con espacio para guardar un número máximo de lecturas
     * del contador igual a max
     */
    public HistoriaConsumos(int max) {...}

    /**
```

```
* Añade una lectura del contador si se puede
*/
public void añadeLectura(double lectura, int dia, int mes, int año) {...}
}
```

Se pide implementar en Java los siguientes métodos adicionales para esta clase, así como un programa principal de prueba:

- 1) `dibuja()`: Dibuja en una ventana de la clase `Grafica` un gráfico del historial de consumos. En el eje X se pondrá el número de días transcurridos desde la primera lectura del contador. En el eje Y figurará el valor de lectura del contador.
- 2) `lee()`: Lee de un objeto de la clase `CajaTexto` varias lecturas del contador con sus fechas, y los añade al historial usando `añadeLectura()`.
- 3) `esCorrecto()`: Usa el esquema de búsqueda en tablas para buscar una casilla de la tabla con una lectura errónea. Retorna `true` si el resultado de la búsqueda es infructuoso (es decir, si todas las casillas son correctas), y `false` si se encuentra una casilla errónea. Una lectura errónea se caracteriza porque el valor de la lectura del contador es menor al de la casilla anterior, o porque la fecha de la lectura es anterior a la de la casilla anterior.
- 4) Programa principal: Hace lo siguiente:
 - Crea una historia de consumos
 - Llama a `lee()` para leer de una caja de texto varias lecturas correctas del contador con sus fechas
 - Invoca a `esCorrecto()` para ver si el historial es correcto, y muestra el resultado en la pantalla
 - Dibuja en pantalla los consumos llamando a `dibuja()`.
 - Añade al historial un par de lecturas erróneas (de fechas desordenadas, o consumos negativos) con `añadeLectura()`
 - Invoca a `esCorrecto()` para ver si el historial es correcto, y muestra el resultado en la pantalla

Nota: para saber si una fecha es anterior o posterior a otra se puede usar el método `numDias()` y ver si el número de días transcurridos entre una fecha y otra es positivo o negativo.