Ingeniería Informática - Depto. de Informática e Ingeniería de Sistemas
Examen de Programación I - 2 de Febrero de 2011

- Disponer sobre la mesa en lugar visible un ***documento de identificación*** provisto de fotografía. Escribir ***nombre y dos apellidos*** en cada una de las hojas de papel que haya sobre la mesa.
- Comenzar a resolver cada una de las partes del examen ***en una hoja diferente*** para facilitar su corrección por profesores diferentes.
- El tiempo total previsto para realizar el examen es de ***tres horas***. No está permitido consultar libros ni apuntes, excepto los dos documentos facilitados por los profesores de la asignatura: *Un breve resumen del lenguaje Java* y *Documentación de las clases Java utilizadas en la asignatura*.

Problema 1º (3.0 puntos)

Las clases *Fecha* y *Estudiante*, definidas en el ***package*** *examen*, permiten gestionar la información de una fecha del calendario y de un estudiante universitario, respectivamente. Alguno de los métodos públicos que ofrecen ambas clases se muestran a continuación. Estos métodos y sólo éstos, podrán ser utilizados en el diseño que se pide más adelante.

```
package examen;

/**
  * Cada uno de los objetos de esta clase gestiona la información asociada a una fecha
  * del calendario.
  */
public class Fecha {

    ...

    /**
      * Post: Crea un objeto Fecha fijando los valores del día (parámetro [dia]), mes
      *       (parámetro [mes]) y año (parámetro [año]) que definen una fecha y devuelve la
      *       referencia al objeto Fecha creado.
      */
    public Fecha (int dia, int mes, int año){ ... }

    /**
      * Post: Devuelve un entero que, escrito en base 10 tiene ocho dígitos [aaaammdd]. Los cuatro
      *       más significativos (aaaa) equivalen al año gestionado por el objeto Fecha, los dos que
      *       le siguen (mm) equivalen al mes y los dos menos significativos (dd) al día del mes.
      */
    public int laFecha () { ... }

    ...
}
```

```
package examen;

/**
  * Cada uno de los objetos de esta clase gestiona la información asociada a un estudiante
  * universitario .
  */
public class Estudiante {

    ...

    /**
      * Post: Crea un objeto Estudiante fijando los valores de su nombre completo (parámetro
      *     [nombre]), su NIP (Número de Identificación Personal) (parámetro [nip]) y su fecha
      *     de nacimiento (parámetro [nacimiento]) y devuelve la referencia al objeto Estudiante
      *     creado.
      */
    public Estudiante (String nombre, int nip, Fecha nacimiento) { ... }

    /**
      * Post: Devuelve la referencia al String que gestiona el nombre completo del estudiante .
      */
    public String nombre () { ... }

    /**
      * Post: Devuelve la referencia al objeto Fecha que gestiona la fecha de nacimiento del
      *         estudiante .
      */
    public Fecha nacido () { ... }

    ...
}
```

Se pide diseñar el método *diaMasCelebrado* que se especifica a continuación y, en su caso, los métodos auxiliares sobre los que se apoye su diseño. Este método pertenece a una clase ubicada en el ***package*** *examen*. El código de las clases *Fecha* y *Estudiante* no puede ser modificado.

```
/**
  * Pre: tabla.length>=1 y ninguno de los elemento de [tabla] tiene valor [null]
  * Post: Devuelve una referencia a un objeto [Fecha] que corresponde al día de nacimiento de
  *     uno de los estudiantes referenciados desde [tabla]. El día y mes de esa fecha corresponde
  *     al día del año en el que celebran su cumpleaños un mayor número de los estudiantes de
  *     [tabla]. En el caso de que el día y mes de varias fechas proporcione el máximo de
  *     cumpleaños, cualquiera de esas fechas es aceptable como resultado del método.
  */
private static Fecha diaMasCelebrado (Estudiante [] tabla)
```

Problema 2º (3.0 puntos)

Se está trabajando en un proyecto software para el desarrollo de un simulador de sistemas dinámicos. En este simulador hay que gestionar la medida del tiempo de simulación. Para ello se ha decidido desarrollar en el ***package*** *simulador* la clase pública *Cronometro*.

Un objeto de la clase *Cronometro* gestiona un tiempo simulado que se expresa en segundos. Para trabajar con objetos de esta clase se deben desarrollar los siguientes métodos públicos con un comportamiento exactamente igual al que se explica a continuación.

- Un método constructor que permita crear un objeto de la clase y ponga a cero el valor del tiempo a cronometrar.
- Un método de nombre *reset* que ponga a cero el valor del tiempo cronometrado por el objeto.
- Un método de nombre *tick* que incremente en un segundo el valor del tiempo cronometrado por el objeto.
- Un método de nombre *tiempo* que devuelva un entero de tipo *int* que, al ser escrito en base 10, presente el valor del tiempo cronometrado por el objeto de la forma [*ddhhmmss*], es decir, sus dos cifras menos significativas denotan los segundos (*ss*), la tercera y cuarta cifra denotan los minutos (*mm*), la quinta y sexta cifra denotan las horas (*hh*) y la séptima y octava cifra denotan los días (dd).

  Así, por ejemplo, si el tiempo cronometrado por el objeto fuera de 12 días 7 horas 5 minutos y 17 segundos este método devolvería el valor entero 12070517. Si el tiempo cronometrado fuera de 13 horas y 30 minutos el método devolvería el valor entero 133000. Y si el tiempo cronometrado fuera de 6 minutos y 9 segundos el método devolvería el valor entero 609.
- Un método de nombre *toString* que devuelva una referencia a un *String* con 11 caracteres que representen el valor del tiempo cronometrado por el objeto de la forma [dd:hh:mm:ss], es decir, los dos caracteres [*ss*] denotan los segundos, los dos caracteres [*mm*] denotan los minutos, los dos caracteres [*hh*] denotan los horas y los dos caracteres [*dd*] denotan los días cronometrados.

  Así, por ejemplo, si el tiempo cronometrado por el objeto fuera de 12 días 7 horas 5 minutos y 17 segundos este método devolvería el *String* ”12:07:05:17”. Si el tiempo cronometrado fuera de 13 horas y 30 minutos el método devolvería el *String* ”00:13:30:00”. Y si el tiempo cronometrado fuera de 6 minutos y 9 segundos el método devolvería el *String* ”00:00:06:09”.

Se pide escribir el texto del fichero *Cronometro.java* con el código completo de la clase *Cronometro*.

Problema 3º (4.0 puntos)

La información de las asignaturas matriculadas por los alumnos de una universidad se almacena en ficheros binarios cuya estructura se describe a continuación.

```
< fichero_de_matriculas > ::= { <matricula> }
<matricula> ::= <NIP> <código_asignatura>
<NIP> ::= <int>
<código_asignatura> ::= <int>
```

Un fichero almacena información de una secuencia de matrículas. Cada matrícula se representa mediante una dupla de datos (NIP, código_asignatura). Tanto el NIP (Número de Identificación Personal) de un alumno como el código de cada de las asignaturas se representa mediante un dato entero de tipo *int*.

La información de las matrículas de los alumnos de una titulación se encuentran almacenadas en varios ficheros binarios con la estructura anterior. Se desea unificar la información de estos ficheros en un único fichero.

Para ello se dispone un fichero de texto de nombre ***fNombreFMat.txt*** que se ubica en el directorio ***/home/admon*** de una máquina **Unix**. La primera línea de este fichero almacena el nombre completo que ha de tener el fichero resultante de unificar los ficheros de matrículas (por ejemplo: ***/home/admon/matriculas/matriculas2100.dat***). Las sucesivas líneas del fichero almacenan el nombre de cada uno de los ficheros binarios de matrículas que se pretenden unificar (por ejemplo: ***/home/admon/matriculas/sep/f1_mat.dat***, ***/home/admon/matriculas/sep/f29_mat.dat***, . . . , ***/home/admon/matriculas/nov/f63_mat.dat***.

```
/home/admon/matriculas/matriculas2100.dat
/home/admon/matriculas/sep/f1_mat.dat
/home/admon/matriculas/sep/f29_mat.dat
/home/admon/matriculas/oct/f3_mat.dat
    .     .     .
/home/admon/matriculas/oct/f12_mat.dat
/home/admon/matriculas/oct/f92_mat.dat
/home/admon/matriculas/nov/f63_mat.dat
```

Se pide escribir el código de un programa Java que, al ser ejecutado, cree un fichero binario unificado de matrículas cuyo nombre figura en la primera línea del fichero de texto ***fNombreFMat.txt*** que almacene la información completa con todas las matrículas registradas en los ficheros binarios de matrículas que figuran a partir de la segunda línea, hasta la última, del mismo fichero ***fNombreFMat.txt***. La clase o clases Java que se escriban se ubicarán en el ***package*** *examen*.

El programa informará además al operador del proceso de unificación de matrículas mostrando por pantalla los siguientes mesajes informativos:

```
... leídas 53 matrículas del fichero /home/admon/matriculas/sep/f1_mat.dat
... leídas 110 matrículas del fichero /home/admon/matriculas/sep/f29_mat.dat
                .     .     .
... leídas 282 matrículas del fichero /home/admon/matriculas/nov/f63_mat.dat
Creado el fichero /home/admon/matriculas/matriculas2100.dat
```

El programa se ejecutará desde una cuenta de usuario con los privilegios necesarios de lectura y escritura en los directorios donde se ubican los diferentes ficheros.