Examen de Prácticas de Programación 1 - 31/enero/2014

- Tiempo máximo para realizar el trabajo de programación propuesto: 90 minutos
- Entrega del trabajo a través de la plataforma *Moodle2*.

Especificación del trabajo a desarrollar en el turno 1º (15:00 horas)

Las clases ***problemas.cap6.Ciudadano*** y ***problemas.cap6.FicherosDeCiudadanos*** fueron propuestas en el capítulo 6 de la colección de problemas la asignatura, han sido desarrolladas en la práctica 5 y han sido utilizadas en el trabajo obligatorio de la asignatura. En esta nueva prueba práctica se va a trabajar de nuevo con ficheros de ciudadanos (para mayor detalle consultar el capítulo 6 de la colección de problemas) y va a ser necesario o, en su caso, conveniente, hacer uso de los recursos definidos en ellas.

En esta prueba cada uno debe desarrollar una única clase cuyo nombre coincida con sus apellidos (ej: **class** LopezGutierrez). Esta clase incluirá un método ***main(...)*** que, al ser ejecutado, presente el siguiente comportamiento:

```
Nombre de un fichero de ciudadanos: datos/pruebas/personal.txt

[A] Dña. MARIA ISABEL AZANZA SIPAN nacida el 20-febrero-1995
[B] Dña. ANA MARIA BADENES MORGADO nacida el 19-agosto-1996
[C] Dña. DANIELA CARDONA PUEBLA nacida el 17-septiembre-1998
[D] Dña. ELENA MARIA DOMINGO LOPE nacida el 02-agosto-1947
          .   .   .
[T] Dña. ESTHER TOMAS OTERO nacida el 07-diciembre-2012
[U] D. FRANCISCO JAVIER UTIEL AZANZA nacido el 17-mayo-1948
[Z] Dña. ANA ISABEL ZARAGOZA ZABALZA nacida el 22-agosto-1974

El mes con el máximo número de nacidos es abril con 33 hombres y 35 mujeres
```

El operador proporciona el nombre de un fichero de ciudadanos, al ser preguntado por el programa.

A continuación, el programa presenta por pantalla un listado de ciudadanos almacenados en el fichero cuyo primer apellido comience por las diferentes letras del alfabeto. El listado se presenta ordenado alfabéticamente por la inicial del primer apellido (A, B, ..., Y, Z) [Nota: sólo se considerarán como iniciales las letras del alfabeto inglés]. Se presenta un solo ciudadano por cada letra. Si, para una letra, no hay ningún ciudadano cuyo primer apellido comience por ella, esa letra es omitida en el listado.

Finalmente, el programa informa por pantalla del mes del año en el que han nacido un mayor número total de ciudadanos de los almacenados en el fichero. En el ejemplo mostrado, el mes de abril, con 68 ciudadanos nacidos en ese mes (de distintos años), es el que arroja un mayor número de ciudadanos que coinciden en haber nacido en un mes determinado.

Se dispone del fichero de ciudadanos **personal.txt** en la sección de *Materiales docentes comunes* de la web de la asignatura para realizar pruebas. Se debe seleccionar el enlace situado en el apartado **Código Java descargable**, accediendo posteriormente a la carpeta ***datos/pruebas***.

Para calificar esta prueba se tendrán en cuenta, en este orden de importancia, los siguientes aspectos:

1. Que el programa presente un comportamiento idéntico al descrito anteriormente, sin erorres de compilación ni de ejecución. Es preferible que el programa realice correctamente solo una parte de las tareas exigidas, a que intente realizarlas todas pero presente errores al ser ejecutado.
2. El diseño del código de acuerdo con los principios explicados en la asignatura.
3. Una adecuada especificación de cada uno de los métodos resultantes del diseño del programa.

Examen de Prácticas de Programación 1 - 31/ENE/2014

- Tiempo máximo para realizar el trabajo de programación propuesto: 90 minutos
- Entrega del trabajo a través de la plataforma *Moodle2*.

Especificación del trabajo a desarrollar en el turno 2º (17:00 horas)

Las clases ***problemas.cap6.Ciudadano*** y ***problemas.cap6.FicherosDeCiudadanos*** fueron propuestas en el capítulo 6 de la colección de problemas la asignatura, han sido desarrolladas en la práctica 5 y han sido utilizadas en el trabajo obligatorio de la asignatura. En esta nueva prueba práctica se va a trabajar de nuevo con ficheros de ciudadanos (para mayor detalle consultar el capítulo 6 de la colección de problemas) y va a ser necesario o, en su caso, conveniente, hacer uso de los recursos definidos en ellas.

En esta prueba cada uno debe desarrollar una única clase cuyo nombre coincida con sus apellidos (ej: **class** LopezGutierrez). Esta clase incluirá un método ***main(...)*** que, al ser ejecutado, presente el siguiente comportamiento:

```
Nombre de un fichero de ciudadanos: datos/pruebas/personal.txt

[1930] D. FEDERICO FERNANDEZ GARCIA nacido el 18-mayo-1930
[1933] D. TIBURCIO BORAO GIMENO nacido el 20-abril-1933
[1934] D. LUIS MIGUEL BUENO ALMUNIA nacido el 20-diciembre-1934
[1935] Dña. MARIA PILAR RODRIGUEZ BORAO nacida el 12-octubre-1935
          .    .    .
[2007] D. LUIS MIGUEL PI BALMES nacido el 14-noviembre-2007
[2008] D. LUIS IGNACIO FONSECA TOLEDO nacido el 12-diciembre-2008
[2009] Dña. ROSA SANTIAGO CARDONA nacida el 07-noviembre-2009
[2012] Dña. EDURNE ARENAS JUANES nacida el 05-octubre-2012

Máximo número de ciudadanos en 1958 con 9 hombres y 8 mujeres
```

El operador proporciona el nombre de un fichero de ciudadanos, al ser preguntado por el programa.

A continuación, el programa presenta por pantalla un listado de algunos de los ciudadanos almacenados en el fichero, todos ellos nacidos a partir de 1900 y hasta el presente año. El listado se presenta ordenado por años. Se presenta un solo ciudadano por cada año. Si, para un año, no hay ningún ciudadano nacido en él, ese año es omitido en el listado.

Finalmente, el programa informa por pantalla del año en el que han nacido un mayor número total de ciudadanos de los almacenados en el fichero. En el ejemplo mostrado, el año de 1958, con 17 ciudadanos, es el que arrojar un mayor número de ciudadanos nacidos en un mismo año.

Se dispone del fichero de ciudadanos **personal.txt** en la sección de *Materiales docentes comunes* de la web de la asignatura para realizar pruebas. Se debe seleccionar el enlace situado en el apartado **Código Java descargable**, accediendo posteriormente a la carpeta ***datos/pruebas***.

Para calificar esta prueba se tendrán en cuenta, en este orden de importancia, los siguientes aspectos:

1. Que el programa presente un comportamiento idéntico al descrito anteriormente, sin erorres de compilación ni de ejecución. Es preferible que el programa realice correctamente solo una parte de las tareas exigidas, a que intente realizarlas todas pero presente errores al ser ejecutado.
2. El diseño del código de acuerdo con los principios explicados en la asignatura.
3. Una adecuada especificación de cada uno de los métodos resultantes del diseño del programa.

Una solución del problema propuesto en el turno 1º

```
import java.util.Scanner;
import java.io.File;
import problemas.cap6.Ciudadano;
import problemas.cap6.FicherosDeCiudadanos;

/**
 * Presenta el programa que resuelve el problema planteado en el turno 1º de la prueba
 * práctica correspondiente a la convocatoria de febrero de 2014 de Programación 1
 */
public class Apellido1Apellido2 {

    private static final String[] MESES = {"enero", "febrero", "marzo", "abril", "mayo", "junio",
            "julio", "agosto", "septiembre", "octubre", "noviembre", "diciembre"};

    /**
     * Pre: T!=null y no almacena [T] ninguna referencia null
     * Post: Informa por pantalla del mes del año con más ciudadanos nacidos en él, entre
     *       los referenciados desde la tabla [T]. Lo hace del siguente modo:
     *
     *         El mes con el máximo número de nacidos es xxxxx con xx hombres y xx mujeres
     */
    private static void maximo(Ciudadano[] T) {
        int[] numHombres = new int[12];
        int[] numMujeres = new int[12];
        for (int i=0; i<T.length; ++i) {
            int mes = T[i].fechaNacimiento()/100 % 100;
            if (T[i].esHombre()) {
                ++numHombres[mes-1];
            }
            else {
                ++numMujeres[mes-1];
            }
        }
        int indMax = 0;
        int maximo = numHombres[0] + numMujeres[0];
        for (int i=1; i<numHombres.length; ++i) {
            int suma = numHombres[i] + numMujeres[i];
            if (suma>maximo) {
                indMax = i;   maximo = suma;
            }
        }

        System.out.printf("%nEl mes con el máximo número de nacidos es %s con %d"
                          + " hombres y %d mujeres%n",
                          MESES[indMax], numHombres[indMax], numMujeres[indMax]);
    }

    /**
     * Pre: T!=null, no almacena [T] ninguna referencia null y [letra] es una letra mayúscula
     * Post: Si el apellido de alguno de los ciudadanos referenciados desde T comienza por
     *       la letra [letra] entonces devuelve la referencia a un objeto Ciudadano cuyo apellido
     *       comienza por dicha letra. En caso contrario devuelve null.
     */
    private static Ciudadano buscar(Ciudadano[] T, char letra) {
        Ciudadano c = null;
        int indice = 0;
        while (c==null && indice<T.length) {
            if (T[indice].apellidos().toUpperCase().charAt(0)== letra) {
                c = T[indice];
```

```
            }
            else {
                ++indice;
            }
        }
        return c;
    }

    /**
     * Pre: --
     * Post: Pregunta al operador por el nombre de un fichero de ciudadanos:
     *           Nombre de un fichero de ciudadanos: [respuesta del operador]
     *
     *        Por cada letra del alfabeto, desde la A hasta la Z, presenta por pantalla en una
     *        línea el nombre completo y el DNI de un ciudadano almacenado en el fichero cuyo
     *        apellido comienza por la letra considerada. Si para una letra no hay ciudadanos
     *        cuyo apellido no comienza por esa letra, entonces omite esa letra. Ejemplo:
     *          [A] Dña. MARIA ISABEL AZANZA SIPAN nacida el 20-febrero-1995
     *          [B] Dña. ANA MARIA BADENES MORGADO nacida el 19-agosto-1996
     *          [C] Dña. DANIELA CARDONA PUEBLA nacida el 17-septiembre-1998
     *                  . . .
     *          [U] D. FRANCISCO JAVIER UTIEL AZANZA nacido el 17-mayo-1948
     *          [Z] Dña. ANA ISABEL ZARAGOZA ZABALZA nacida el 22-agosto-1974
     *
     *        A continuación, informa por pantalla del mes del año con más ciudadanos
     *        nacidos, entre los almacenados en el fichero. Lo hace del siguente modo:
     *           El mes con el máximo número de nacidos es abril con 33 hombres y 35 mujeres
     */
    public static void main(String[] args) {
        /*
         * Pregunta al operador por el nombre del fichero de ciudadanos
         */
        System.out.print("Nombre de un fichero de ciudadanos: ");
        Scanner teclado = new Scanner(System.in);
        String nombreFichero = teclado.nextLine();
        File f = new File(nombreFichero);
        if (f.isFile()) {
            /*
             * Define una tabla T de referencias a objetos Ciudadano. Cada uno de ellos
             * gestiona la información de uno de los ciudadanos cuyos datos constan en
             * el fichero
             */
            Ciudadano[] T = FicherosDeCiudadanos.leerFichero(f);
            /*
             * Presenta un listado de parcial de ciudadanos del fichero . El apellido de
             * cada uno de ellos comienza por una letra distinta del alfabeto.
             */
            System.out.println();
            for (char letra = 'A'; letra <='Z'; ++letra) {
                Ciudadano c = buscar(T, letra);
                if (c!=null) {
                    if (c.esHombre()) {
                        System.out.printf("[%s] D. %s %s nacido ", letra, c.nombre(),
                                c.apellidos());
                    }
                    else {
                        System.out.printf("[%s] Dña. %s %s nacida ", letra, c.nombre(),
                                c.apellidos());
                    }
                    System.out.printf("el %02d-%s-%d %n", c.fechaNacimiento()%100,
                            MESES[c.fechaNacimiento()/100%100-1], c.fechaNacimiento()/10000);
```

```
                }

            }
            /*
             * Informa por pantalla del mes del año con más ciudadanos nacidos, entre los
             * almacenados en el fichero.
             */
            maximo(T);
        }
        else {
            /*
             * Informa que el nombre del fichero de ciudadanos aportado es erróneo
             */
            System.out.printf("El nombre del fichero <%s> es desconocido %n", nombreFichero);
        }
    }

}
```

Una solución del problema propuesto en el turno 2º

```
import java.util.Scanner;
import java.io.File;
import problemas.cap6.Ciudadano;
import problemas.cap6.FicherosDeCiudadanos;


/**
 * Presenta el programa que resuelve el problema planteado en el turno 2º de la prueba
 * práctica correspondiente a la convocatoria de febrero de 2014 de Programación 1
 */
public class FebreroTurno2 {

    private static final String[] MESES = {"enero", "febrero", "marzo", "abril", "mayo", "junio",
            "julio", "agosto", "septiembre", "octubre", "noviembre", "diciembre"};

    /**
     * Pre: desde<=hasta, T!=null y no almacena [T] ninguna referencia null
     * Post: Informa por pantalla del año comprendido en el periodo [desde,hasta]
     *       en el que más ciudadanos hayan nacido entre los referenciados desde
     *       T. Informa del siguiente modo:
     *
     *          Máximo número de ciudadanos en 19xx con xx hombres y xx mujeres
     */
    private static void maximo(Ciudadano[] T, int desde, int hasta) {
        /*
         * Cuenta el número de hombre y mujeres referenciados desde [T] nacidos
         * en cada uno de los años del periodo [desde,hasta
         */
        int[] numHombres = new int[hasta-desde+1];
        int[] numMujeres = new int[hasta-desde+1];
        for (int i=0; i<T.length; ++i) {
            int año = T[i].fechaNacimiento()/10000;
            if (T[i].esHombre()) {
                ++numHombres[año-desde];
            }
            else {
                ++numMujeres[año-desde];
            }
        }
        /*
         * Busca el año en el que la suma de nacimientos sea máxima
         */
        int indMax = 0;
        int maximo = numHombres[0] + numMujeres[0];
        for (int i=1; i<numHombres.length; ++i) {
            int suma = numHombres[i] + numMujeres[i];
            if (suma>maximo) {
                indMax = i;   maximo = suma;
            }
        }
        /*
         * Presenta los resultados pr pantalla
         */
        System.out.printf("%nMáximo número de ciudadanos en %d con %d hombres y %d mujeres%n",
                          desde+indMax, numHombres[indMax], numMujeres[indMax]);
    }

    /**
     * Pre: T!=null y no almacena [T] ninguna referencia null
```

```
     * Post: Si alguno de los ciudadanos referenciados desde T ha nacido en el año
     *       [año] entonces devuelve la referencia a un objeto Ciudadano nacido en
     *       dicho año. En caso contrario devuelve null.
     */
    private static Ciudadano buscar (Ciudadano[] T, int año) {
        Ciudadano c = null;
        int indice = 0;
        while (c==null && indice<T.length) {
            if (T[indice].fechaNacimiento()/10000==año) {
                c = T[indice];
            }
            else {
                ++indice;
            }
        }
        return c;
    }

    /**
     * Pre: --
     * Post: Pregunta al operador por el nombre de un fichero de ciudadanos:
     *          Nombre de un fichero de ciudadanos: [respuesta del operador]
     *
     *       Por cada año, desde 1990 hasta el presente, presenta por pantalla en una línea
     *       con el nombre completo y la fecha de nacimiento de un ciudadano almacenado en el fichero
     *
cuyo año de nacimiento coincida con el año considerado. Si para un año no hay ciudadanos
     *       nacidos en él, entonces omite ese año. Ejemplo:
     *         [1930] D. FEDERICO FERNANDEZ GARCIA nacido el 18-mayo-1930
     *         [1933] D. TIBURCIO BORAO GIMENO nacido el 20-abril-1933
     *         [1934] D. LUIS MIGUEL BUENO ALMUNIA nacido el 20-diciembre-1934
     *                . . .
     *         [2008] D. LUIS IGNACIO FONSECA TOLEDO nacido el 12-diciembre-2008
     *         [2009] Dña. ROSA SANTIAGO CARDONA nacida el 07-noviembre-2009
     *         [2012] Dña. EDURNE ARENAS JUANES nacida el 05-octubre-2012
     *
     *       A continuación, informa por pantalla del año con más ciudadanos nacidos en él.
     *       Lo hace del siguente modo:
     *          Máximo número de ciudadanos en 1958 con 9 hombres y 8 mujeres
     */
    public static void main(String[] args) {
        /*
         * Comienzo y final del periodo a considerar
         */
        final int INICIO = 1901;
        final int FIN = 2014;
        /*
         * Pregunta al operador por el nombre del fichero de ciudadanos
         */
        System.out.print("Nombre de un fichero de ciudadanos: ");
        Scanner teclado = new Scanner(System.in);
        String nombreFichero = teclado.nextLine();
        File f = new File(nombreFichero);
        if (f.isFile()) {
            /*
             * Define una tabla T de referencias a objetos Ciudadano. Cada uno de ellos
             * gestiona la información de uno de los ciudadanos cuyos datos constan en
             * el fichero
             */
            Ciudadano[] T = FicherosDeCiudadanos.leerFichero(f);
            /*
```

```
            * Presenta un listado de parcial de ciudadanos del fichero. Por cada año
            * comprendido en el periodo [INICIO,FIN] presenta, si existe, la información
            * de un ciudadano nacido en dicho año y almacenado en el fichero
            */
           System.out.println();
           for (int año = INICIO; año<=FIN; ++año) {
               Ciudadano c = buscar(T,año);
               if (c!=null) {
                   if (c.esHombre()) { System.out.printf("[%d] D. ", año); }
                   else { System.out.printf("[%d] Dña. ", año); }
                   System.out.printf("%s %s", c.nombre(), c.apellidos());
                   if (c.esHombre()) { System.out.print(" nacido "); }
                   else { System.out.print(" nacida "); }
                   System.out.printf("el %02d-%s-%d%n", c.fechaNacimiento()%100,
                           MESES[c.fechaNacimiento()/100%100-1], c.fechaNacimiento()/10000);
               }
           }
           /*
            * Informa por pantalla del año con más ciudadanos nacidos, entre los almacenados
            * en el fichero.
            */
           maximo(T,INICIO,FIN);
       }
       else {
           System.out.printf("El nombre del fichero <%s> es desconocido%n", nombreFichero);
       }
   }

}
```