

Examen de prácticas de Programación 1

Escuela de Ingeniería y Arquitectura
Departamento de Informática e Ingeniería de Sistemas

12 de septiembre de 2013

- Tiempo para realizar el trabajo de programación propuesto: **75 minutos**.
- Entrega del trabajo: envío del fichero correspondiente al código fuente Java de la clase que se pide desarrollar (fichero de extensión «.java», no «.class») por correo electrónico a la dirección que se indique al comienzo del examen. No debe levantarse de su puesto de trabajo ni abandonar la sala hasta que el profesor le confirme la recepción del trabajo.
- En esta prueba se valorarán los siguientes aspectos y con este orden de importancia:
 1. Comportamiento del programa según las especificaciones de este enunciado.
 2. Legibilidad del código.
 3. Diseño algorítmico del código.

Un programa que tenga errores de compilación o que, al ser ejecutado, no proporcione ningún resultado, será calificado con un cero.

Introducción al problema a resolver

En este examen se va a trabajar con las clases Ciudadano, FicheroCiudadanos, Socio y Asociacion del paquete trabajoProg1 correspondientes al trabajo de la asignatura. Se van a aumentar las capacidades del programa realizado en el trabajo de la asignatura al permitir realizar un control del pago de las cuotas de los socios dados de alta en una determinada asociación.

Para ello, se dispone de **ficheros binarios** que recogen información sobre el valor de las cuotas y el pago de las mismas por parte de los socios de una determinada asociación en un determinado año. En concreto, el formato de dichos ficheros es el siguiente: el primer dato

de cada fichero es un dato de tipo **double** que representa el importe de la cuota anual de un determinado año. Le siguen un número desconocido de ternas de datos que representan los distintos pagos realizados por los socios para satisfacer la cuota del año de que se trate. Cada pago está representado por tres datos: un dato de tipo **int** que representa el número de DNI del socio que ha hecho el pago, un dato de tipo **int** que representa la fecha en la que se realizó el pago (codificada como un entero en el que sus cuatro dígitos más significativos en base 10 representan el año en el que se efectuó el pago, los dos dígitos siguientes, el mes y, por último, los dos dígitos menos significativos representan el día del mes) y, por último, un dato de tipo **double** que representa el importe satisfecho en ese pago. Por ejemplo, la terna de datos <<12345678>, <20130912>, <24.0>> representa el hecho de que el socio con DNI n.º 12345678 realizó el día 12 de septiembre de 2013 un pago de 24 euros.

El formato del fichero expresado en notación de Backus-Naur es el siguiente:

```
<ficheroPagos> ::= <cuotaAnual> {<pagoSocio>}
<cuotaAnual> ::= double
<pagoSocio> ::= <dniSocio> <fechaPago> <importePago>
<dniSocio> ::= int
<fechaPago> ::= int
<importePago> ::= double
```

A modo de ejemplo, se muestra el contenido del fichero de pagos denominado «mus2012.dat» correspondiente a la cuota y pagos de una asociación de jugadores de mus durante el año 2012:

```
<15.0> <12345678> <20120101> <10.0> <23456789> <20120102> <10.0>
<34567890> <20120102> <5.0> <12345678> <20120601> <5.0>
```

Los datos del fichero de ejemplo indican que durante el año en cuestión, el valor de la cuota anual fue de 15 euros. El socio con DNI n.º 12345678 realizó un pago de 10 euros el 1 de enero del 2012. El día 2 de enero de 2012 los socios con DNI n.º 23456789 y 34567890 realizaron pagos de 10 y 5 euros cada uno, respectivamente. El día 1 de junio de 2012, el socio con DNI n.º 12345678 volvió a realizar un pago, esta vez por un importe de 5 euros. A la vista de los datos, puede afirmarse que el socio con DNI n.º 12345678 ha satisfecho el pago de la cuota de ese año.

Especificación del programa a desarrollar

Dentro de una clase ubicada en el paquete trabajoProg1, que tenga por nombre su primer apellido, debe escribir el método cuya especificación se muestra a continuación:

```

/**
 * Pre: «asoc» != null; «ficheroCuotas» != null; «ficheroCuotas» representa
 *       el nombre de un fichero binario existente con el formato descrito en
 *       el enunciado y que puede ser leído.
 * Post: Ha devuelto el número de socios de la asociación «asoc» que no están
 *       al corriente de los pagos de su cuota, según la información del
 *       fichero binario «cuotas».
 */
public static int numSociosConImpagos(Asociacion asoc, String ficheroCuotas) {
    ...
}

```

El método solicitado podrá apoyarse en la utilización de los métodos públicos de las clases Ciudadano, Socio, Asociacion y FicheroCiudadanos del paquete trabajoProg1 que se estimen convenientes, y en aquellos otros métodos adicionales que se estimen necesarios. Estos últimos deberán declararse e implementarse, con carácter privado, en el seno de la clase solicitada.

Con el objeto de facilitar la prueba del código solicitado, se proporcionan los siguientes materiales, disponibles en la sección «Materiales docentes comunes. Código Java descargable» de página web de la asignatura¹:

- Un fichero de texto denominado «sociosMus.txt» y ubicado en el directorio «datos/trabajoProg1/sept», cuyo contenido está formado por los datos de cuatro ciudadanos que son socios de una determinada asociación.
- Un ficheros binario denominado «mus2012.dat» y ubicado también en el directorio «datos/trabajoProg1/sept», cuyo contenido corresponde con los pagos de cuota de socios de la asociación. Su contenido coincide con el mostrado en el ejemplo de este enunciado.
- Un esqueleto de la clase cuyo desarrollo se solicita, denominada provisionalmente «Apellido», ubicada en el directorio «trabajoProg1» y que incluye un método main que, al ser ejecutado, crea una asociación a partir de los ciudadanos listados en el fichero de ejemplo de ciudadanos, analiza el número de impagados a través de los datos del fichero de cuotas de ejemplo e informa escribiendo en la pantalla el número de impagados existentes en la asociación. En el caso de que el método solicitado sea correcto, debe escribir el siguiente mensaje:

Número de socios con impagos de la asociación «Amigos del mus»: 3

Se facilita el dato adicional de que los DNI de los socios con impagos en el ejemplo anterior son los siguientes: 23456789, 34567890 y 10123123.

¹http://webdiis.unizar.es/asignaturas/PROG1/doc/codigo_java/

Solución

```
package trabajoProg1;

import java.io.DataInputStream;
import java.io.EOFException;
import java.io.FileInputStream;
import java.io.IOException;

public class Septiembre2013 {

    /**
     * Pre: «asoc» != null; «ficheroCuotas» != null; «ficheroCuotas»
     * representa el nombre de un fichero binario existente con el formato
     * descrito en el enunciado y que puede ser leído.
     *
     * Post: Ha devuelto el número de socios de la asociación «asoc» que no
     * están al corriente de los pagos de su cuota, según la información del
     * fichero binario «cuotas».
     */
    /*
     * La cabecera de este método irá en el fichero <<Apellido>>
     */
    public static int numSociosConImpagos(Asociacion asoc,
        String ficheroCuotas) {
        double totalPagosPorSocio[] = new double[asoc.numSocios()];
        double cuota = leerPagos(asoc, ficheroCuotas, totalPagosPorSocio);
        return contarSociosImpago(cuota, totalPagosPorSocio);
    }

    /**
     * Pre: «asoc» != null; «ficheroCuotas» != null; «ficheroCuotas»
     * representa el nombre de un fichero binario existente con el formato
     * descrito en el enunciado y que puede ser leído; «totalPagosPorSocio»
     * no es nulo, tiene un número de componentes igual al número de socios
     * de «asoc» y el valor de cada componenete es 0.0
     *
     * Post: Ha devuelto el valor de la cuota anual según el contenido del
     * fichero «ficheroCuotas». Además, el valor la i-ésima componente de
     * «totalPagosPorSocio» es igual al importe total satisfecho por el
     * socio de la asociación «asoc» cuyo índice de antigüedad es i+1, de
     * acuerdo con el contenido del fichero «ficheroCuotas».
     */
}
```

```

*/
private static double leerPagos(Asociacion asoc, String ficheroCuotas,
    double[] totalPagosPorSocio) {
    try {
        DataInputStream f =
            new DataInputStream(new FileInputStream(ficheroCuotas));
        // Lectura del valor de la cuota anual
        double cuotaAnual = f.readDouble();

        // Lectura de los pagos hechos por los socios
        try {
            while (true) {
                // Lectura de un pago: socio, fecha e importe
                int dniSocio = f.readInt();
                f.readInt(); // fecha, que se ignora
                double pago = f.readDouble();

                // Actualización del importe del socio
                int indiceSocio = buscar(asoc, dniSocio);
                totalPagosPorSocio[indiceSocio - 1] += pago;
            }
        }
        catch (EOFException ex) {
        }
        f.close();
        return cuotaAnual;
    }
    catch (IOException ex) {
        System.out.println("Error al procesar el fichero:_"
            + ex.getMessage());
        return -1.0;
    }
}

/**
 * Pre: «asoc» != null y existe entre los socios de «asoc» uno cuyo DNI
 * coincide con «dniSocio»;
 *
 * Post: Ha devuelto el índice del socio cuyo DNI es «dniSocio» según su
 * antigüedad en la asociación «asoc» (el socio más antiguo es el de
 * índice 1)
 */
private static int buscar(Asociacion asoc, int dniSocio) {

```

```

    int indice = 1;
    while (asoc.socio(indice).datosPersonales().dni() != dniSocio) {
        indice++;
    }
    return indice;
}

/**
 * Pre: «totalPagosPorSocio» != null
 *
 * Post: Ha devuelto el número de componentes de «totalPagosPorSocio»
 * cuyo valor es inferior al valor de «cuota»
 */
private static int contarSociosImpago(double cuota,
    double[] totalPagosPorSocio) {
    int cuenta = 0;
    for (int i = 0; i < totalPagosPorSocio.length; i++) {
        if (totalPagosPorSocio[i] < cuota) {
            cuenta++;
        }
    }
    return cuenta;
}

/**
 * Nombre y ruta del fichero binario de ejemplo con el pago de cuotas
 */
/*
 * Esta constante irá en el fichero <<Apellido>>
 */
protected static final String FICHERO_CUOTAS =
    "datos/trabajoProg1/sept/mus2012.dat";

/**
 * Nombre y ruta del fichero binario de ejemplo con socios
 */
/*
 * Esta constante irá en el fichero <<Apellido>>
 */
protected static final String FICHERO_CIUDADANOS =
    "datos/trabajoProg1/sept/sociosMus.txt";

```

```

/**
 * Post: Ha creado una asociación a partir de los ciudadanos listados en
 * FICHERO_CIUDADANOS, ha analizado el número de impagados a través de
 * los datos del fichero FICHERO_CUOTAS y ha informado escribiendo en la
 * pantalla el número de impagados existentes en la asociación.
 *
 * Debería haber escribir el siguiente mensaje:
 *
 * Número de socios con impagos de la asociación «Amigos del mus»: 3
 */
/*
 * Este método irá en el fichero <<Apellido>>
 */
public static void main(String[] args) {
    Asociacion asoc = new Asociacion("Amigos_del_mus", 100, 1);
    asoc.alta(FICHERO_CIUDADANOS);
    int numeroDeSociosConImpagos =
        numSociosConImpagos(asoc, FICHERO_CUOTAS);
    System.out.printf(
        "Número_de_socios_con_impagos_de_la_asociación_«%s»:_%d%n",
        asoc.nombre(), numeroDeSociosConImpagos);
}
}

```