

Programación 2

Examen de teoría (julio 2014)

8 de julio de 2014



Instrucciones

- **Duración: 3 horas**
- El fichero del primer ejercicio debe llamarse **entidades.cc**. Para el segundo problema es necesario entregar cuatro ficheros, llamados **Collection.cc**, **Collection.h**, **Album.cc**, **Album.h**. Pon tu DNI y tu nombre en un comentario al principio de todos los ficheros fuente.
- La entrega se realizará como en las prácticas, a través del servidor del DLSI (<http://pracdlsi.dlsi.ua.es>), en el enlace **Programación 2**. Puedes realizar varias entregas, aunque sólo se corregirá la última.

Problemas

1. (5.5 puntos)

Se pide realizar un programa que extraiga entidades (personas, lugares, fechas y compañías) a partir de un conjunto de ficheros de texto. A continuación se muestran dos ejemplos de fichero¹:

noticia1.txt

La madrugada del @8 de octubre@ murio \$Manuel Onrrubia\$ un abonado de la compania #Movistar# en el *hospital Gregorio Maranon de Madrid* aquejado de una infeccion pulmonar. \$Manuel\$ lucho hasta el ultimo momento para mantenerse con vida y poder cambiar de proveedor pues nunca habia conocido otra compania que no fuera la suya. La gente le hablaba mal de #vodafone# #orange# #ono# y otras empresas de la competencia pero el deseaba experimentar nuevas sensaciones y conocer otra gente.

noticia2.txt

El presidente de los *Estados Unidos* \$Barack Obama\$ aseguro el pasado @3 de mayo@ que ya ha tomado una decision respecto a la crisis en *Ucrania* y tiene pensada una respuesta al apoyo de *Rusia* a las acciones de los separatistas armados que estan tratando de amenazar y desestabilizar al gobierno de la region. Sin embargo, antes de poner en marcha su estrategia, el mandatario quiere escuchar y valorar todas las opiniones de su entorno incluyendo tu cuenta de #twitter#.

Los ficheros tienen identificadas una serie de entidades con el siguiente formato: @fecha@, \$persona\$, #empresa# y *lugar*. Por simplificar, se asumirá que una entidad no se va a encontrar a mitad de dos líneas del fichero, y que dentro de una entidad no habrán otras entidades. Ten en cuenta que una entidad puede estar compuesta por varias palabras (por ejemplo, “Manuel Onrrubia”).

Deberás realizar un programa que reciba como entrada una lista de nombres de ficheros² para mostrar la información de todas las entidades detectadas en todos ellos de manera agrupada. Un ejemplo de llamada y su correspondiente salida sería:

```
./entidades noticia1.txt noticia2.txt
```

Fechas: 8 de octubre | 3 de mayo

Personas: Manuel Onrrubia | Manuel | Barack Obama

Empresas: Movistar | vodafone | orange | ono | twitter

Lugares: hospital Gregorio Maranon de Madrid | Estados Unidos | Ucrania | Rusia

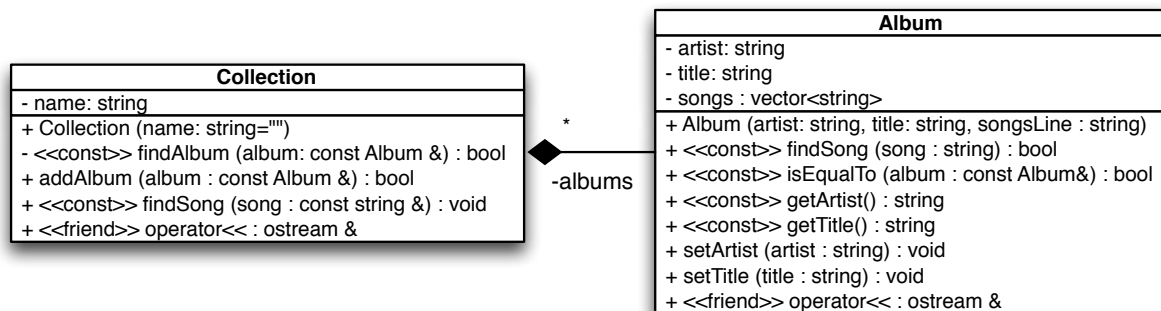
Nota: No hace falta comprobar si existen entidades repetidas o no; se mostrarán todas.

¹Pueden descargarse desde www.dlsi.ua.es/~pertusa/examen/materiales.zip

²Si el programa no recibe ningún parámetro, debe mostrar un error de sintaxis y finalizar.

2. (4.5 puntos)

Queremos hacer un programa que nos permita gestionar una colección de álbumes con canciones. A continuación se muestra el diagrama de clases:



Un album tiene un autor, un título, y una lista de canciones. El constructor recibe como tercer parámetro un `string` con las canciones separadas por comas (tal como puede verse en el ejemplo del `main`), que deben añadirse al vector `songs`.

En la clase `Album`, el método `findSong` devuelve `true` si la canción pasada por parámetro está en el vector, y el método `isEqualTo` devuelve `true` si dos álbumes tienen el mismo título y autor. El operador salida imprime el título, el autor entre paréntesis, y a continuación todas las canciones del album separadas por guiones como puede verse al final del ejemplo de salida.

En la clase `Collection`, el método `findAlbum` devuelve `true` si se ha encontrado el album. El método `addAlbum` añade un album a la colección si este no estaba ya, y en caso contrario muestra un error. El método `findSong` recibe el nombre de una canción³, y debe buscar los álbumes donde está dicha canción, imprimiendo por pantalla su nombre y autor entre paréntesis, como puede verse en el ejemplo del `main`. El operador salida imprime el nombre de la colección y a continuación todos sus álbumes.

Dado el siguiente fichero `main.cc`:

```

#include "Collection.h"

int main()
{
    Album album("Weezer","Hurley","Memories,Ruling Me,Trainwrecks,Unspoken");
    Album album2("Camela","Lo mejor de Camela","Cuando zarpa el amor,Nunca debí enamorarme,Nadie como tu");

    Collection c("My collection");
    c.addAlbum(album);
    c.addAlbum(album2);
    c.addAlbum(album2);           // Ya estaba, no se añade y se muestra un error por pantalla

    c.findSong("Trainwrecks"); // Busca esta canción en todos los álbumes e imprime sus datos

    cout << c << endl;         // Mostramos la colección completa
}
  
```

...el programa debería imprimir lo siguiente:

```

Ya existe un album igual
La canción Trainwrecks está en el album Hurley (Weezer)
-----My collection-----
Hurley (Weezer) : Memories-Ruling Me-Trainwrecks-Unspoken
Lo mejor de Camela (Camela) : Cuando zarpa el amor-Nunca debí enamorarme-Nadie como tu
  
```

Ayuda: Puedes descargar un `makefile` para compilar en <http://www.dlsi.ua.es/~pertusa/examen/materiales.zip>

³Se supondrá que en la colección no habrá dos canciones con el mismo nombre.